ETSI GS NFV-SOL 003 V4.3.1 (2022-07)

Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; RESTful protocols specification for the Or-Vnfm Reference Point

Disclaimer

The present document has been produced and approved by the Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
RGS/NFV-SOL003ed431

Keywords
API, NFV, protocol

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Network Functions Virtualisation (NFV).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document specifies a set of RESTful protocols and data models fulfilling the requirements specified in ETSI GS NFV-IFA 007 [1] for the interfaces used over the Or-Vnfm reference point, except for the "Virtualised Resources Management interfaces in indirect mode" as defined in clause 6.4 of ETSI GS NFV-IFA 007 [1].

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI GS NFV-IFA 007: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Or-Vnfm reference point - Interface and Information Model Specification".

[2] ETSI GS NFV-SOL 004: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; VNF Package and PNFD Archive specification".

[3] IETF RFC 5646: "Tags for Identifying Languages".

NOTE: Available at https://tools.ietf.org/html/rfc5646.

[4] IETF RFC 7233: "Hypertext Transfer Protocol (HTTP/1.1): Range Requests".

NOTE: Available at https://tools.ietf.org/html/rfc7233.

[5] IETF RFC 7396: "JSON Merge Patch".

NOTE: Available at https://tools.ietf.org/html/rfc7396.

[6] ETSI GS NFV-IFA 027: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Performance Measurements Specification".

[7] Recommendation ITU-T X.733: "Information technology - Open Systems Interconnection - Systems Management: Alarm reporting function".

[8] ETSI GS NFV-SOL 013: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; Specification of common aspects for RESTful NFV MANO APIs".

[9] IETF RFC 7193: "The application/cms Media Type".

NOTE: Available from: https://tools.ietf.org/rfc/rfc7193.txt

[10] ETSI GS NFV-IFA 011: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; VNF Descriptor and Packaging Specification".

[11] IETF RFC 7232: "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests".

NOTE: Available at https://tools.ietf.org/html/rfc7232.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GS NFV 003: "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

[i.2] ETSI GS NFV-SOL 002: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; RESTful protocols specification for the Ve-Vnfm Reference Point".

[i.3] ETSI NFV registry of VimConnectionInfo information.

NOTE: Available at http://register.etsi.org/NFV.

[i.4] ETSI GS NFV-SOL 001: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; NFV descriptors based on TOSCA specification".

[i.5] OpenStack® documentation: "Disk and container formats for images".

NOTE 1: Available at https://docs.openstack.org/glance/pike/user/formats.html.

NOTE 2: The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. ETSI is not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

[i.6] JSON Schema: "Core definitions and terminology", Version draft-07, November 19, 2017.

NOTE 1: Draft-07 is available at https://json-schema.org/specification-links.html#draft-7.

NOTE 2: The specification is available as Internet Draft at https://tools.ietf.org/html/draft-handrews-json-schema-01.

[i.7] OpenAPI™ Specification.

NOTE: Available at https://github.com/OAI/OpenAPI-Specification.

[i.8] Void.

[i.9] Void.

[i.10] Void.

[i.11] ETSI GS NFV-SOL 015: "Network Functions Virtualisation (NFV); Protocols and Data Models; Specification of Patterns and Conventions for RESTful NFV-MANO APIs".

[i.12] Void.

[i.13] ETSI GS NFV-SOL 005: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point".

[i.14] ETSI GS NFV-SOL 010: "Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; VNF Snapshot Package specification".

[i.15] ETSI GS NFV-IFA 040: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Requirements for service interfaces and object model for OS container management and orchestration specification".

[i.16] ETSI GR NFV-IFA 029: "Network Functions Virtualisation (NFV) Release 3; Architecture; Report on the Enhancements of the NFV architecture towards 'Cloud-native' and 'PaaS'".

[i.17] ETSI GS NFV-SOL 018: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; Profiling specification of protocol and data model solutions for OS Container management and orchestration".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GS NFV 003 [i.1] and the following apply:

Compute MCIO: MCIO which declarative descriptor specifies compute infrastructure resource requests

LCM workflow: set of operations, including resource management operations towards the VIM, that are executed by the VNFM to perform a lifecycle management operation

NOTE: Examples for LCM workflows are VNFM-internal procedures associated with an LCM operation, and LCM scripts contained in the VNF package.

Network MCIO: MCIO which declarative descriptor specifies network infrastructure resource requests

Storage MCIO: MCIO which declarative descriptor specifies storage infrastructure resource requests

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

API	Application Programming Interface
CIR	Container Image Registry
CIS	Container Infrastructure Service
CISM	Container Infrastructure Service Management
CP	Connection Point
CPD	CP Descriptor
CSAR	Cloud Service ARchive
EM	Element Manager
ETSI	European Telecommunications Standards Institute
FM	Fault Management
GS	Group Specification
GUI	Graphical User Interface
HTTP	HyperText Transfer Protocol
IETF	Internet Engineering Task Force
IFA	InterFaces and Architecture
IP	Internet Protocol
ISO	International Organization for Standardization
ITU	International Telecommunication Union
JSON	JavaScript Object Notation
LCCN	Life Cycle Change Notifications
LCM	Lifecycle Management
MAC	Medium Access Control
MANO	Management and Orchestration
MCIO	Managed Container Infrastructure Object
MCIOP	Managed Container Infrastructure Object Package

NAD	Network Attachment Definition
NF	Network Function
NFV	Network Functions Virtualisation
NFVI	NFV Infrastructure
NFVI-PoP	Network Function Virtualisation Infrastructure Point of Presence
NFVO	NFV Orchestrator
NS	Network Service
NSD	Network Service Descriptor
PM	Performance Management
PNFD	Physical Network Function Descriptor
RAM	Random Access Memory
REST	Representational State Transfer
RFC	Request For Comments
SR-IOV	Single Root Input/Output Virtualisation
SSL	Secure Socket Layer
TLS	Transport Layer Security
TOSCA	Topology and Orchestration Specification for Cloud Applications
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
vCPU	Virtualised CPU
VDU	Virtualisation Deployment Unit
VIM	Virtualised Infrastructure Manager
VIP	Virtual IP
VL	Virtual Link
VLAN	Virtual LAN
VLD	VL Descriptor
VM	Virtual Machine
VNF	Virtualised Network Function
VNFC	VNF Component
VNFD	VNF Descriptor
VNFM	VNF Manager
VPN	Virtual Private Network
YAML	YAML Ain't Markup Language
YANG	Yet Another Next Generation

4 General aspects

4.1 Overview

The present document defines the protocol and data model for the following interfaces used over the Or-Vnfm reference point, in the form of RESTful Application Programming Interface (API) specifications:

- VNF Lifecycle Management interface (as produced by the VNFM towards the NFVO).
- VNF Performance Management interface (as produced by the VNFM towards the NFVO).
- VNF Fault Management interface (as produced by the VNFM towards the NFVO).
- VNF Indicator interface (as produced by the VNFM towards the NFVO).
- VNF Lifecycle Operation Granting interface (as produced by the NFVO towards the VNFM).
- VNF Package Management interface (as produced by the NFVO towards the VNFM).
- Virtualised Resources Quota Available Notification interface (as produced by the NFVO towards the VNFM).
- VNF Snapshot Package Management interface (as produced by the VNFM towards the NFVO).

Table 4.1-1 lists the versions of the APIs defined in the present document.

Table 4.1-1: Versions of the APIs specified in the present document

API	API version
VNF Lifecycle Management interface	2.10.0
VNF Performance Management interface	2.10.0
VNF Fault Management interface	1.10.0
VNF Indicator interface	1.10.0
VNF Lifecycle Operation Granting interface	1.10.0
VNF Package Management interface	2.10.0
Virtualised Resources Quota Available Notification interface	1.10.0
VNF Snapshot Package Management interface	1.10.0

The design of the protocol and data model for the above interfaces is based on the information model and requirements defined in ETSI GS NFV-IFA 007 [1]. In clause 4, general aspects are specified that apply to multiple APIs on the Or-Vnfm reference point. In addition, the provisions in clauses 4, 5, 6, 8 and 9 of ETSI GS NFV-SOL 013 [8] define common aspects of RESTful NFV-MANO APIs, and shall apply for all APIs defined in the present document.

In the subsequent clauses, the protocol and data model for the individual interfaces are specified. Per interface, the resource structure with associated HTTP methods is defined and applicable flows are provided. Further, the resources and the data model are specified in detail.

Annex A provides the mapping of the combination of resources and methods defined in the present document to the operations defined in ETSI GS NFV-IFA 007 [1]. Annex B contains explanations of key concepts. Annex C defines the structure of the VimConnectionInfo registry.

Even though the different interfaces defined in the present document (apart from the Virtualised Resources Quota Available Notification Interface) are related, implementations shall not assume a particular order of messages that arrive via different interfaces.

4.2 Void

4.3 Void

4.4 Common data types

4.4.1 Structured data types

4.4.1.1 Introduction

This clause defines data structures that are referenced from data structures in multiple interfaces. In addition, the structured data types defined in clause 7.1 of ETSI GS NFV-SOL 013 [8] shall apply.

4.4.1.2 Void

4.4.1.3 Void

4.4.1.3a Void

4.4.1.4 Void

4.4.1.5 Type: VnfInstanceSubscriptionFilter

This type represents subscription filter criteria to match VNF instances. It shall comply with the provisions defined in table 4.4.1.5-1.

Table 4.4.1.5-1: Definition of the VnfInstanceSubscriptionFilter data type

Attribute name	Data type	Cardinality	Description
vnfdIds	Identifier	0..N	If present, match VNF instances that were created based on a VNFD identified by one of the vnfdId values listed in this attribute. See note 1.
vnfProductsFromProviders	Structure (inlined)	0..N	If present, match VNF instances that belong to VNF products from certain providers. See note 1.
>vnfProvider	String	1	Name of the VNF provider to match.
>vnfProducts	Structure (inlined)	0..N	If present, match VNF instances that belong to VNF products with certain product names, from one particular provider.
>>vnfProductName	String	1	Name of the VNF product to match.
>>versions	Structure (inlined)	0..N	If present, match VNF instances that belong to VNF products with certain versions and a certain product name, from one particular provider.
>>>vnfSoftwareVersion	Version	1	Software version to match.
>>>vnfdVersions	Version	0..N	If present, match VNF instances that belong to VNF products with certain VNFD versions, a certain software version and a certain product name, from one particular provider.
vnfInstanceIds	Identifier	0..N	If present, match VNF instances with an instance identifier listed in this attribute. See note 2.
vnfInstanceNames	String	0..N	If present, match VNF instances with a VNF Instance Name listed in this attribute. See note 2.

NOTE 1: The attributes "vnfdIds" and "vnfProductsFromProviders" are alternatives to reference to VNF instances that are based on certain VNFDs in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.

NOTE 2: The attributes "vnfInstanceIds" and "vnfInstanceNames" are alternatives to reference to particular VNF instances in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.

4.4.1.6 Type: VimConnectionInfo

This type represents parameters to connect to a VIM, a CISM, a CIR or a MCIOP repository for managing the resources of a VNF instance. It shall comply with the provisions defined in table 4.4.1.6-1.

This structure is used to convey VIM-related, CISM-related, CIR-related, or MCIOP-repository-related parameters over the Or-Vnfm interface. Additional parameters for a VIM, a CISM, a CIR or a MCIOP repository may be configured into the VNFM by means outside the scope of the present document and bound to the identifier of that VIM.

Table 4.4.1.6-1: Definition of the VimConnectionInfo data type

Attribute name	Data type	Cardinality	Description
vimId	Identifier	0..1	The identifier of the VIM, CISM, CIR or MCIOP repository instance. This identifier is managed by the NFVO. Shall be present to address additional information about the VIM, CISM, CIR or MCIOP repository if such information has been configured into the VNFM by means outside the scope of the present document and should be absent otherwise.

Attribute name	Data type	Cardinality	Description
vimType	String	1	Discriminator for the different types of the VIM information. The value of this attribute determines the structure of the "interfaceInfo" and "accessInfo" attributes, based on the type of the VIM, CISM, CIR or MCIOP repository. The set of permitted values is expected to change over time as new types or versions of VIMs become available. The ETSI NFV registry of VIM-related information [i.3] provides access to information about VimConnectionInfo definitions for various VIM, CISM, CIR or MCIOP repository types. The structure of the registry is defined in annex C.
interfaceInfo	KeyValuePairs	0..1	Information about the interface or interfaces to the VIM, CISM, CIR or MCIOP repository, if applicable, such as the URI of an interface endpoint to communicate with the VIM, CISM, CIR or MCIOP repository. The applicable keys are dependent on the content of vimType. Alternatively, such information may have been configured into the VNFM and bound to the vimId.
accessInfo	KeyValuePairs	0..1	Authentication credentials for accessing the VIM, CISM, CIR or MCIOP repository and other access-related information such as tenants or infrastructure resource groups (see note 1). The applicable keys are dependent on the content of vimType. If the VimConnectionInfo structure is part of an HTTP response payload body, sensitive attributes that are children of this attributes (such as passwords) shall not be included. If the VimConnectionInfo structure is part of an HTTP request payload body, sensitive attributes that are children of this attribute (such as passwords) shall be present if they have not been provisioned out of band. See note 2.
extra	KeyValuePairs	0..1	VIM, CISM, CIR or MCIOP repository type specific additional information. The applicable structure, and whether or not this attribute is available, is dependent on the content of vimType.

NOTE 1: If applicable, this attribute also provides information about the resourceGroupIds that are accessible using a particular set of credentials. See definition of "resourceGroupId" in clause 9.5.3.3.

NOTE 2: Once the connectivity between VNFM and VIM, CISM, CIR or MCIOP repository is provided through a secure connection over HTTP Secure (HTTP over SSL/TLS), and the connection might also be established through a VPN (for example TLS-based VPN tunnelling) for site-to-site connection, the "accessInfo" JSON data structure, and the sensitive data related information ("username"/"password" as required properties for authentication purpose), will be transmitted as plain text through a TLS tunnel without additional encoding/encryption before transmitting it, making the sensitive data visible to the endpoint. The base64 encoded certificates are only used by the VNFM to verify the authenticity of the interface endpoint of the VIM, CISM, CIR or MCIOP repository.

4.4.1.7 Type: ResourceHandle

This type represents the information that allows addressing a virtualised resource that is used by a VNF instance. Information about the resource is available from the VIM. The ResourceHandle type shall comply with the provisions defined in table 4.4.1.7-1.

Table 4.4.1.7-1: Definition of the ResourceHandle data type

Attribute name	Data type	Cardinality	Description
vimConnectionId	Identifier	0..1	Identifier of the VIM or CISM connection to manage the resource. This attribute shall be supported when the resource is managed by a CISM. When the resource is managed by a VIM, this attribute shall only be supported and present if VNF-related resource management in direct mode is applicable. The applicable "VimConnectionInfo" structure, which is referenced by vimConnectionId, can be obtained from the "vimConnectionInfo" attribute of the "VnfInstance" structure.
resourceProviderId	Identifier	0..1	Identifier of the entity responsible for the management of the resource. This attribute shall only be supported and present when VNF-related resource management in indirect mode is applicable. The identification scheme is outside the scope of the present document.
resourceId	IdentifierInVim	1	Identifier of the resource in the scope of the VIM or the CISM or the resource provider. See note 2.
vimLevelResourceType	String	0..1	Type of the resource in the scope of the VIM or the CISM or the resource provider. See note 1.
vimLevelAdditionalResourceInfo	AdditionalResourceInfo	0..1	Additional resource information which is specific to this resource and its type, and which is available from the VIM or the CISM or the resource provider.
containerNamespace	String	0..1	The value of the namespace in which the MCIO corresponding to the resource is deployed. This attribute shall be present if the resource is managed by a CISM and it shall be absent otherwise.

NOTE 1: The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or CISM or the resource provider and can be used as information that complements the ResourceHandle. This value set is different from the value set of the "type" attribute in the ResourceDefinition (refer to clause 9.5.3.2). When the container infrastructure service management is a Kubernetes® instance the vimLevelResourceType is the type of resource, as would correspond to the 'kind' field if the resource is declared in its own Kubernetes® manifest, e.g.: Pod, PersistentVolumeClaim, NetworkAttachmentDefinition.

NOTE 2: When the container infrastructure service management is a Kubernetes® instance the resourceId shall be populated in the following way:

- For a compute MCIO, it is the instance identifier that Kubernetes® assigns, which is unique cluster wide per resource type.
- For a storage MCIO modelled as a persistent volume claim, it is the name of the persistent volume claim, i.e. the value of the 'claimName' field in the Kubernetes® manifest, or a compound name built by Kubernetes® if the persistent volume claim is defined inline in another template instead of in its own manifest.
- For a network MCIO representing a NetworkAttachmentDefinition, a Service or an Ingress, it is the value of the 'metadata.name' field in Kubernetes® manifest.

4.4.1.8 Void

4.4.1.9 Void

4.4.1.10 Type: VnfExtCpData

This type represents configuration information for external CPs created from a CPD. It shall comply with the provisions defined in table 4.4.1.10-1.

Table 4.4.1.10-1: Definition of the VnfExtCpData data type

Attribute name	Data type	Cardinality	Description
cpdId	IdentifierInVnfd	1	The identifier of the CPD in the VNFD. See note 1.
cpConfig	map(VnfExtCpConfig)	1..N	Map of instance data that need to be configured on the CP instances created from the respective CPD. The key of the map which identifies the individual VnfExtCpConfig entries is of type "IdentifierInVnf" and is managed by the NFVO. The entries shall be applied by the VNFM according to the rules of JSON Merge Patch (see IETF RFC 7396 [5]). See notes 2, 3 and 4.

NOTE 1: In case this identifier refers to a CPD with trunking enabled, the external CP instances created from this CPD will represent ports in a trunk.

NOTE 2: Within one VNF instance, all VNFC instances created from a particular VDU have the same external connectivity. Thus, given a particular value of the "cpdId" attribute, there shall be one "cpConfig" entry for each VNFC instance that has been or can be created from a VDU which includes a CPD identified by the "cpdId" attribute. If the cpConfig represents a subport in a trunk, all "cpConfig" entries in this list shall have the same segmentationId, which means they are connected to the same set of external VLs via the trunk.

NOTE 3: The map entry value shall be set to "null" in order to delete a "VnfExtCpConfig" entry identified by a particular key value from the map, i.e. for the disconnection of an existing external CP instance addressed by cpInstanceId in the deleted map entry from a particular external virtual link, and deletion of that instance in case it represents a subport. Deleting the last key from the map removes the affected instance of the "VnfExtCpData" structure from its parent data structure.

NOTE 4: If, as defined by the input parameters of a "ChangeVnfFlavour", "ChangeExtVnfConnectivity" or "ChangeCurrentVnfPkg" operation or as part of the Grant response for any of these operations, a cpConfig map entry identified by a particular map key value is moved into another "ExtVirtualLinkData" or "VnfExtCpData" structure, this particular cpConfig map entry may be used by an external CP instance different than the one that has used it before the operation, or by no external CP instance at all. Renaming a CPD identifier during the "changeCurrentVnfPkg" operation does not count as moving the related "cpConfig" map entries to a new "extCpData" structure.

4.4.1.10a Type: VnfExtCpConfig

This type represents an externally provided link port, or a network attachment definition resource of secondary container cluster network, or network address information per instance of an external connection point.

In the case of VM-based deployment of the VNFC exposing the external CP:

1) In case a link port is provided, the VNFM shall use that link port when connecting the external CP to the external VL.

2) In case a link port is not provided, the VNFM shall create a link port on the external VL and use that link port to connect the external CP to the external VL.

In the case of container-based deployment of the VNFC exposing the external CP, the VNFM shall use the network attachment definition resource of secondary container cluster network when connecting the CP to the external VL.

This type shall comply with the provisions defined in table 4.4.1.10a-1.

Table 4.4.1.10a-1: Definition of the VnfExtCpConfig data type

Attribute name	Data type	Cardinality	Description
parentCpConfigId	IdentifierInVnf	0..1	Value of the key that identifies the "VnfExtCpConfig" map entry which corresponds to the parent port of the trunk. Only present in "VnfExtCpConfig" structures that provide configuration information for a CP which represents a sub-port in a trunk, and if parent ports are supported.
linkPortId	Identifier	0..1	Identifier of a pre-configured link port to which the external CP will be associated. See notes 1 and 4.

Attribute name	Data type	Cardinality	Description
createExtLinkPort	Boolean	0..1	Indicates to the VNFM the need to create a dedicated link port for the external CP. If set to True, the VNFM shall create a link port. If set to False, the VNFM shall not create a link port. This attribute is only applicable for external CP instances without a floating IP address that expose a VIP CP instance for which a dedicated IP address is allocated. It shall be present in that case and shall be absent otherwise.
netAttDefResourceId	Identifier	0..N	Identifier of the "NetAttDefResourceData" structure that provides the specification of the interface to attach the external CP to a secondary container cluster network. It is only applicable if the external CP is connected or to be connected to a secondary container cluster network. It shall not be present if the external CP is related to a virtual network not categorized as secondary container cluster network. See notes 2, 3 and 4.
cpProtocolData	CpProtocolData	0..N	Parameters for configuring the network protocols on the link port that connects the CP to a VL. See notes 1 and 2.

NOTE 1: The following conditions apply to the attributes "linkPortId" and "cpProtocolData" for an external CP instance connected or to be connected to a virtual network not categorized as secondary container cluster network:
1) Void.
2) At least one of the "linkPortId" and "cpProtocolData" attributes shall be present for an external CP instance representing a subport that is to be created, or an external CP instance that is to be created by creating the corresponding VNFC or VNF instance during the current or a subsequent LCM operation, or for an existing external CP instance that is to be re-configured or added to a particular external virtual link.
3) If the "linkPortId" attribute is absent, the VNFM shall create a link port.
4) If the "cpProtocolData" attribute is absent, the "linkPortId" attribute shall be provided referencing a pre-created link port, and the VNFM can use means outside the scope of the present document to obtain the pre-configured address information for the connection point from the resource representing the link port.
5) If both "cpProtocolData" and "linkportId" are provided, the NFVO shall ensure that the cpProtocolData can be used with the pre-created link port referenced by "linkPortId".

NOTE 2: The following conditions apply to the attributes "netAttDefResourceId" and "cpProtocolData" for an external CP instance connected or to be connected to a secondary container cluster network:
1) The "netAttDefResourceId" and "cpProtocolData" attributes shall both be absent for the deletion of an existing external CP instance addressed by cpInstanceId.
2) The "netAttDefResourceId" attribute shall be present and the "cpProtocolData" attribute may be present for a to-be-created external CP instance or an existing external CP instance.

NOTE 3: Cardinality greater than 1 is only applicable for specific cases where more than one network attachment definition resource is needed to fulfil the connectivity requirements of the external CP, e.g. to build a link redundant mated pair in SR-IOV cases. When more than one netAttDefResourceId is indicated, all shall belong to the same namespace as defined by the corresponding "netAttDefResourceNamespace" attribute in the "NetAttDefResourceData".

NOTE 4: Either linkPortId or netAttDefResourceId may be included, but not both.

4.4.1.10b Type: CpProtocolData

This type represents network protocol data. It shall comply with the provisions defined in table 4.4.1.10b-1.

Table 4.4.1.10b-1: Definition of the CpProtocolData data type

Attribute name	Data type	Cardinality	Description
layerProtocol	Enum (inlined)	1	Identifier of layer(s) and protocol(s). Permitted values: - IP_OVER_ETHERNET - IP_FOR_VIRTUAL_CP See note.
ipOverEthernet	IpOverEthernetAddressData	0..1	Network address data for IP over Ethernet to assign to the extCP instance. Shall be present if layerProtocol is equal to "IP_OVER_ETHERNET" and shall be absent otherwise.
virtualCpAddress	VirtualCpAddressData	0..1	IP address data to assign to an external CP instance exposing a virtual CP. It shall be present if layerProtocol is equal to "IP_FOR_VIRTUAL_CP" and the external CP instance exposes a virtual CP and shall not be present otherwise.
NOTE: This attribute allows to signal the addition of further types of layer and protocol in future versions of the present document in a backwards-compatible way. In the current version of the present document, only IP over Ethernet is supported.			

4.4.1.10c Type: IpOverEthernetAddressData

This type represents network address data for IP over Ethernet. It shall comply with the provisions defined in table 4.4.1.10c-1.

Table 4.4.1.10c-1: Definition of the IpOverEthernetAddressData data type

Attribute name	Data type	Cardinality	Description
macAddress	MacAddress	0..1	MAC address. If this attribute is not present, it shall be chosen by the VIM. See note 1.
segmentationType	Enum	0..1	Specifies the encapsulation type for the traffics coming in and out of the trunk subport. Permitted values: - VLAN: the subport uses VLAN as encapsulation type. - INHERIT: the subport gets its segmentation type from the network it is connected to. This attribute may be present for CP instances that represent subports in a trunk and shall be absent otherwise. If this attribute is not present for a subport CP instance, default value VLAN shall be used.
segmentationId	String	0..1	Identification of the network segment to which the CP instance connects to. See note 3 and note 4.
ipAddresses	Structure (inlined)	0..N	List of IP addresses to assign to the CP instance. Each entry represents IP address data for fixed or dynamic IP address assignment per subnet. If this attribute is not present, no IP address shall be assigned. See note 1.
>type	Enum (inlined)	1	The type of the IP addresses. Permitted values: IPV4, IPV6.
>fixedAddresses	IpAddress	0..N	Fixed addresses to assign (from the subnet defined by "subnetId" if provided). See note 2.

Attribute name	Data type	Cardinality	Description
>numDynamicAddresses	Integer	0..1	Number of dynamic addresses to assign (from the subnet defined by "subnetId" if provided). See note 2.
>addressRange	Structure (inlined)	0..1	An IP address range to be used, e.g. in case of egress connections. In case this attribute is present, IP addresses from the range will be used. See note 2.
>>minAddress	IpAddress	1	Lowest IP address belonging to the range.
>>maxAddress	IpAddress	1	Highest IP address belonging to the range.
>subnetId	IdentifierInVim	0..1	Subnet defined by the identifier of the subnet resource in the VIM. In case this attribute is present, IP addresses from that subnet will be assigned; otherwise, IP addresses not bound to a subnet will be assigned.
NOTE 1: At least one of "macAddress" or "ipAddresses" shall be present. NOTE 2: Exactly one of "fixedAddresses", "numDynamicAddresses" or "ipAddressRange" shall be present. NOTE 3: If the CP instance represents a subport in a trunk, segmentationId shall be present. Otherwise it shall not be present. NOTE 4: Depending on the NFVI networking infrastructure, the segmentationId may indicate the actual network segment value (e.g. vlan Id, Vxlan segmentation id, etc.) used in the transport header of the packets or it may be an identifier used between the application and the NFVI networking infrastructure to identify the network sub-interface of the trunk port in question. In the latter case the NFVI infrastructure will map this local segmentationId to whatever segmentationId is actually used by the NFVI's transport technology.			

4.4.1.10d Type: VirtualCpAddressData

This type represents network address data for a virtual CP. It shall comply with the provisions defined in table 4.4.1.10d-1.

Table 4.4.1.10d-1: Definition of the VirtualCpAddressData data type

Attribute name	Data type	Cardinality	Description
type	Enum (inlined)	1	The type of the IP addresses. Permitted values: IPV4, IPV6.
loadBalancerIp	IpAddress	0..1	Fixed address to assign to an external load balancer. See notes 1 and 2.
NOTE 1: If the container cluster is set up to be able to configure an external load balancer this address will be used, otherwise it will be ignored by the CISM. NOTE 2: In case the cluster can configure an external load balancer but no loadBalancerIp is provided the container cluster will assign an IP address.			

4.4.1.11 Type: ExtVirtualLinkData

This type represents an external VL. It shall comply with the provisions defined in table 4.4.1.11-1.

Table 4.4.1.11-1: Definition of the ExtVirtualLinkData data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	The identifier of the external VL instance. The identifier is assigned by the NFV-MANO entity that manages this VL instance.
vimConnectionId	Identifier	0..1	Identifier of the VIM connection to manage this resource. This attribute shall only be supported and present if VNF-related resource management in direct mode is applicable.

Attribute name	Data type	Cardinality	Description
resourceProviderId	Identifier	0..1	Identifies the entity responsible for the management of this resource. This attribute shall only be supported and present if VNF-related resource management in indirect mode is applicable. The identification scheme is outside the scope of the present document.
resourceId	IdentifierInVim	1	The identifier of the resource in the scope of the VIM or the resource provider.
extCps	VnfExtCpData	1..N	External CPs of the VNF to be connected to this external VL. Entries in the list of external CP data that are unchanged need not be supplied if the ExtVirtualLinkData structure is part of a request or response that modifies the external connectivity.
extLinkPorts	ExtLinkPortData	0..N	Externally provided link ports to be used to connect external connection points to this external VL. If this attribute is not present, the VNFM shall create the link ports on the external VL except in the cases defined below. See note 1.
extNetAttDefResourceData	NetAttDefResourceData	0..N	Externally provided network attachment definition resource(s) that provide the specification of the interface to attach external CPs to this external VL. See note 2. It is only applicable if the external VL is realized by a secondary container cluster network. It shall not be present otherwise.

NOTE 1: A link port is not needed for an external CP instance that exposes a VIP CP in the following cases:
- 1 For a VIP CP directly exposed as an external CP:
  - 1.1 No dedicated IP address is allocated as VIP address, as indicated in the VNFD.
  - 1.2 A dedicated IP address is allocated as VIP address, but the NFVO indicates that no port is needed (createExtLinkPort in VnfExtCpConfig set to false).
- 2 For a VIP CP exposed as an external CP via a floating IP address:
  - 2.1 No dedicated IP address is allocated as VIP address, as indicated in the VNFD, and the VNFC CP associated to the VIP CP is also exposed via a floating IP address.
- 3 For a VIRTUAL CP exposed as an external CP.
- 4 For a VNFC CP exposed as an external CP in a secondary container cluster external network or a secondary container cluster internal network.

NOTE 2: An example of the network attachment definition resource when the container infrastructure service management is a Kubernetes® instance is a Network Attachment Definition (NAD).

4.4.1.12 Type: ExtManagedVirtualLinkData

This type represents an externally-managed internal VL. It shall comply with the provisions defined in table 4.4.1.12-1.

Table 4.4.1.12-1: Definition of the ExtManagedVirtualLinkData data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	The identifier of the externally-managed internal VL instance. The identifier is assigned by the NFV-MANO entity that manages this VL instance.
vnfVirtualLinkDescId	IdentifierInVnfd	1	The identifier of the VLD in the VNFD for this VL.
vimConnectionId	Identifier	0..1	Identifier of the VIM connection to manage this resource. This attribute shall only be supported and present if VNF-related resource management in direct mode is applicable.
resourceProviderId	Identifier	0..1	Identifies the entity responsible for the management of this resource. This attribute shall only be supported and present if VNF-related resource management in indirect mode is applicable. The identification scheme is outside the scope of the present document.
resourceId	IdentifierInVim	1	The identifier of the resource in the scope of the VIM or the resource provider.

Attribute name	Data type	Cardinality	Description
netAttDefResourceData	NetAttDefResourceData	0..N	Externally provided network attachment definition resource(s) that provide the specification of the interface to attach VNFC connection points to this externally-managed VL. See notes 1 and 3.
intCp	IntVnfCpData	0..N	Internal CPs of the VNF to be connected to this externally-managed VL. See note 1.
vnfLinkPort	VnfLinkPortData	0..N	Externally provided link ports to be used to connect VNFC connection points to this externally-managed VL on this network resource. If this attribute is not present, the VNFM shall create the link ports on the externally-managed VL. See note 2.
extManagedMultisiteVirtualLinkId	Identifier	0..1	Identifier of the externally-managed multi-site VL instance. The identifier is assigned by the NFV-MANO entity that manages the externally managed multi-site VL instance. It shall be present when the present externally-managed internal VL (indicated by extManagedVirtualLinkId) is part of a multi-site VL, e.g. in support of multi-site VNF spanning several VIMs. All externally-managed internal VL instances corresponding to an internal VL created based on the same virtualLinkDescId shall refer to the same extManagedMultisiteVirtualLinkId.
NOTE 1: It is only applicable if the externally-managed VL is realized by a secondary container cluster network. It shall not be present otherwise. NOTE 2: A link port is not needed for a VNFC internal connection point connected to a secondary container cluster network. NOTE 3: An example of the network attachment definition resource when the container infrastructure service management is a Kubernetes® instance is a network attachment definition (NAD)			

4.4.1.13 Void

4.4.1.14 Type: ExtLinkPortData

This type represents an externally provided link port to be used to connect an external connection point to an external VL. It shall comply with the provisions defined in table 4.4.1.14-1.

Table 4.4.1.14-1: Definition of the ExtLinkPortData data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this link port as provided by the entity that has created the link port.
resourceHandle	ResourceHandle	1	Reference to the virtualised resource realizing this link port.
trunkResourceId	IdentifierInVim	0..1	Identifier of the trunk resource in the VIM. Shall be present if the present link port corresponds to the parent port that the trunk resource is associated with. See note.
NOTE: The value of "trunkResourceId" is scoped by the value of "vimConnectionId" in the "resourceHandle" attribute.			

4.4.1.15 Type: ScaleInfo

This type represents the scale level of a VNF instance related to a scaling aspect. It shall comply with the provisions defined in table 4.4.1.15-1.

Table 4.4.1.15-1: Definition of the ScaleInfo data type

Attribute name	Data type	Cardinality	Description
aspectId	IdentifierInVnfd	1	Identifier of the scaling aspect.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case the value differs from the vnfdId attribute of the VnfInstance (e.g. during a "Change current VNF package" operation or due to its final failure).
scaleLevel	Integer	1	Indicates the scale level. The minimum value shall be 0 and the maximum value shall be ≤ maxScaleLevel as described in the VNFD.

4.4.1.16 Type: AdditionalResourceInfo

This type represents additional resource information which resource and resource type specific, and which is available from the VIM or the CISM or the resource provider. It shall comply with the provisions defined in table 4.4.1.16-1.

Table 4.4.1.16-1: Definition of the AdditionalResourceInfo data type

Attribute name	Data type	Cardinality	Description
hostName	String	0..1	Name of the host where the resource is allocated. It shall be present for compute resources in the scope of the CISM and shall be absent otherwise. See note.
persistentVolume	String	0..1	Name of the persistent volume to which the persistent volume claim representing the storage resource is bound. It may be present for storage resources in the scope of the CISM and shall be absent otherwise. See note.
additionalInfo	KeyValuePairs	0..1	Information related to other properties directly owned by the resource and available from the VIM or CISM or the resource provider. See note.
NOTE: At least one attribute shall be present.			

4.4.1.17 Type: NetAttDefResourceData

This type represents a network attachment definition resource that provides the specification of the interface to be used to connect one or multiple connection points to a secondary container cluster network realizing a VL. It shall comply with the provisions defined in table 4.4.1.17-1.

Table 4.4.1.17-1: Definition of the NetAttDefResourceData data type

Attribute name	Data type	Cardinality	Description
netAttDefResourceId	Identifier	1	Identifier of this network attachment definition resource as provided by the entity that has created it.
resourceHandle	ResourceHandle	1	Resource handle of the resource identifying the network attachment definition resource that provides the specification of the interface to attach the connection points to a secondary container cluster network.

4.4.1.18 Type: IntVnfCpData

This type represents input information related to one or more VNF internal CP instances created based on the same CPD. It shall comply with the provisions defined in table 4.4.1.18-1.

Table 4.4.1.18-1: Definition of the IntVnfCpData data type

Attribute name	Data type	Cardinality	Description
cpdId	Identifier	1	Identifier of the CPD in the VNFD.
netAttDefResourceId	Identifier	1..N	Identifiers of the "NetAttDefResourceData" structure that provides the specification of the interface to attach the VNF internal CP created from the CPD identified by cpdId to a secondary container cluster network. See note.
NOTE: Cardinality greater than 1 is only applicable for specific cases where more than one network attachment definition resource is needed to fulfil the connectivity requirements of the VNF internal CP, e.g. to build a link redundant mated pair in SR-IOV cases. When more than one netAttDefResourceId is indicated, all shall belong to the same namespace as defined by the corresponding "netAttDefResourceNamespace" attribute in the "NetAttDefResourceData".			

4.4.2 Simple data types and enumerations

4.4.2.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in multiple interfaces.

4.4.2.2 Simple data types

Table 4.4.2.2-1 defines simple data types for reference from data type definitions in the present document. In addition, the simple data types defined in clause 7.2.2 of ETSI GS NFV-SOL 013 [8] shall apply.

Table 4.4.2.2-1: Simple data types

Type name	Description
IdentifierInVnfd	An identifier that is unique within a VNF descriptor. Representation: string of variable length.
IdentifierInVim	An identifier maintained by the VIM or the CISM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.
IdentifierInVnf	An identifier that is unique for the respective type within a VNF instance, but that need not be globally unique. Representation: string of variable length.
IdentifierLocal	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.

4.4.2.3 Enumerations

4.4.2.3.1 Introduction

This clause defines enumerations that are referenced from data types in multiple interfaces. In addition, the enumerations defined in clause 7.2.3 of ETSI GS NFV-SOL 013 [8] shall apply to be available for referencing from data type definitions in the present document.

4.4.2.3.2 Enumeration: LcmCoordResultType

The enumeration LcmCoordResultType defines the permitted values to represent the result of executing an LCM coordination action. The coordination result also implies the action to be performed by the VNFM as the follow-up to this coordination. The LcmCoordResultType shall comply with the provisions defined in table 4.4.2.3.2-1.

Table 4.4.2.3.2-1: Enumeration LcmCoordResultType

Enumeration value	Description
CONTINUE	The related LCM operation shall be continued, staying in the state "PROCESSING".
ABORT	The related LCM operation shall be aborted by transitioning into the state "FAILED_TEMP".
CANCELLED	The coordination action has been cancelled upon request of the API consumer, i.e. the VNFM. The related LCM operation shall be aborted by transitioning into the state "FAILED_TEMP".

4.5 Void

4.6 Void

4.7 Void

4.8 HTTP conditional requests

Conditional requests are HTTP (POST, PUT, PATCH and DELETE) requests that include one or more header fields indicating a precondition to be tested before applying the method semantics to the target resource. They are supported by metainformation about the resource that was provided in earlier HTTP responses. The set of HTTP header fields to compose a conditional request allowed in the present version of the present document is defined in clause 4.2 of ETSI GS NFV-SOL 013 [8] with the following provisions applicable to the HTTP requests and responses.

POST: For resources that also support the PUT or PATCH method, the API producer should provide the "ETag" and the "Last-Modified" HTTP headers in the POST response when the HTTP response codes are "200 OK", "201 Created" or "204 No Content".

GET: For resources that also support the PUT or PATCH method, the API producer should provide the "ETag" and the "Last-Modified" HTTP headers in the GET response when the response code is "200 OK".

PATCH: The API producer should provide the "ETag" and the "Last-Modified" HTTP headers in the PATCH response when the response code are "200 OK" or "204 No Content". In case the related "Last-Modified" or "ETag" headers have been received in previous responses related to the target resource, the API consumer should provide the "If-Unmodified-Since" or the "If-Match" header fields as conditions (see sections 3.1 and 3.4 of IETF RFC 7232 [11]) in order to prevent conflicts with other changes to the resource.

When receiving the request, the API producer shall test if there is an "If-Unmodified-Since" or "If-Match" header included. If such header is included, the API producer shall further process the request according to sections 3.1, 3.4 and 4.2 of IETF RFC 7232 [11]. This includes to check if there is a mismatch between the content of that header and the last modification of the target resource and to return the HTTP response code "412 Precondition failed" in such a case, indicating that the resource was modified by another entity since the API consumer has obtained the representation of the resource.

PUT: The same provisions as for PATCH apply.

DELETE: The API producer shall not provide the "ETag" and the "Last-Modified" HTTP headers in the DELETE response as these headers are not applicable when the resource has ceased to exist.

5 VNF Lifecycle Management interface

5.1 Description

This interface allows the NFVO to invoke VNF lifecycle management operations of VNF instances towards the VNFM, and to subscribe to notifications regarding VNF lifecycle changes provided by the VNFM.

The operations provided through this interface are:

- Create VNF Identifier
- Query VNF
- Modify VNF Information

- Delete VNF Identifier
- Instantiate VNF
- Scale VNF
- Scale VNF to Level
- Change VNF Flavour
- Terminate VNF
- Heal VNF
- Operate VNF
- Change external VNF connectivity
- Change current VNF package
- Create VNF snapshot
- Revert to VNF snapshot
- Query VNF snapshot information
- Delete VNF snapshot information
- Fetch VNF state snapshot
- Get Operation Status
- Subscribe
- Query Subscription Information
- Terminate Subscription
- Notify

NOTE: The interface provides also the capability to modify information of an "Individual VNF snapshot" resource to fulfil the extraction of a VNF snapshot from a VNF snapshot package.

This interface also enables to invoke error handling procedures (Retry, Rollback, Cancel, Fail) on the actual VNF lifecycle management operation occurrences, and API version information retrieval.

5.1a API version

For the VNF lifecycle management interface version as specified in the present document, the MAJOR version field shall be 2, the MINOR version field shall be 10 and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [8] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v2".

5.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [8]. The string "vnflcm" shall be used to represent {apiName}. All resource URIs in clauses below are defined relative to the above base URI.

Figure 5.2-1 shows the overall resource URI structure defined for the VNF lifecycle management interface.

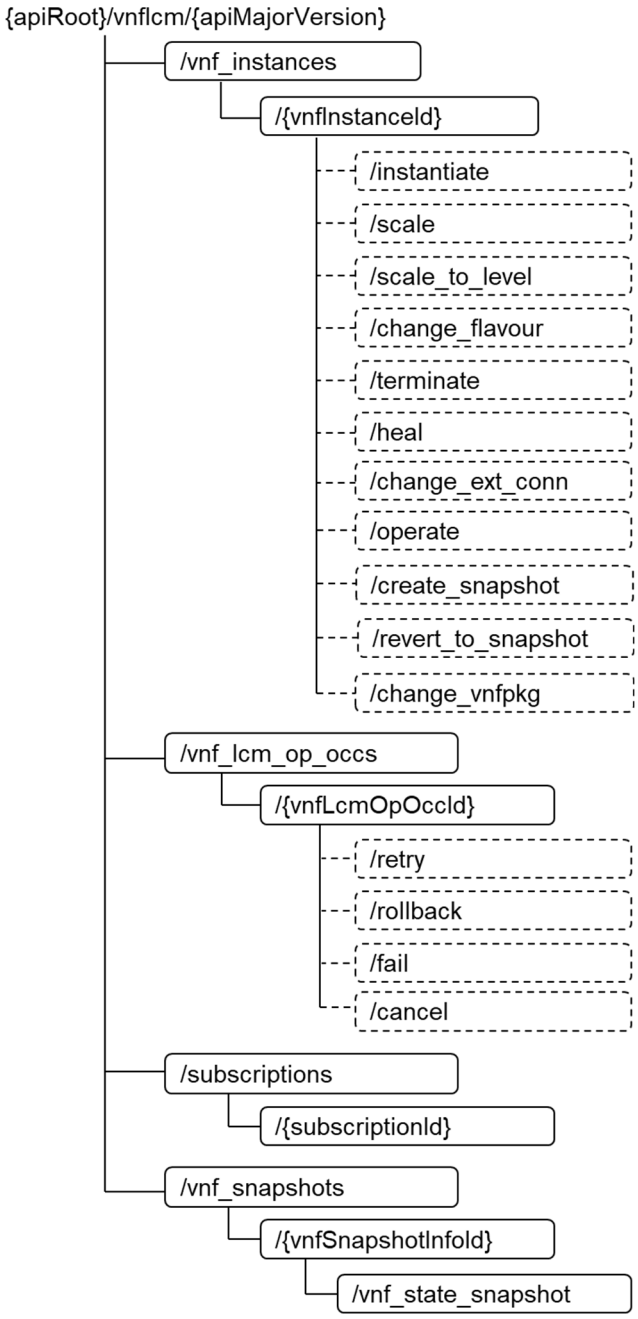

Figure 5.2-1: Resource URI structure of the VNF Lifecycle Management Interface

Table 5.2-1 lists the individual resources defined, and the applicable HTTP methods.

The VNFM shall support responding to requests for all HTTP methods on the resources in table 5.2-1 that are marked as "M" (mandatory) in the "Cat" column. The VNFM shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [8].

Table 5.2-1: Resources and methods overview of the VNF Lifecycle Management interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
VNF instances	/vnf_instances	GET	M	Query multiple VNF instances.
		POST	M	Create a new "Individual VNF instance" resource.
Individual VNF instance	/vnf_instances/{vnfInstanceId}	GET	M	Read an "Individual VNF instance" resource.
		PATCH	M	Modify VNF instance information.
		DELETE	M	Delete an "Individual VNF instance" resource.
Instantiate VNF task	/vnf_instances/{vnfInstanceId}/instantiate	POST	M	Instantiate a VNF.
Scale VNF task	/vnf_instances/{vnfInstanceId}/scale	POST	M	Scale a VNF instance incrementally.
Scale VNF to Level task	/vnf_instances/{vnfInstanceId}/scale_to_level	POST	M	Scale a VNF instance to a target level.
Change VNF flavour task	/vnf_instances/{vnfInstanceId}/change_flavour	POST	M	Change the deployment flavour of a VNF instance.
Terminate VNF task	/vnf_instances/{vnfInstanceId}/terminate	POST	M	Terminate a VNF instance.
Heal VNF task	/vnf_instances/{vnfInstanceId}/heal	POST	M	Heal a VNF instance.
Operate VNF task	/vnf_instances/{vnfInstanceId}/operate	POST	M	Operate a VNF instance.
Change external VNF connectivity task	/vnf_instances/{vnfInstanceId}/change_ext_conn	POST	M	Change the external connectivity of a VNF instance.
Change current VNF package task	/vnf_instances/{vnfInstanceId}/change_vnfpkg	POST	M	Change the current VNF package on which a VNF instance is based.
Create VNF snapshot task	/vnf_instances/{vnfInstanceId}/create_snapshot	POST	M	Create a VNF snapshot.
Revert to VNF snapshot task	/vnf_instances/{vnfInstanceId}/revert_to_snapshot	POST	M	Revert a VNF instance to a VNF snapshot.
VNF LCM operation occurrences	/vnf_lcm_op_occs	GET	M	Query information about multiple VNF lifecycle management operation occurrences.
Individual VNF LCM operation occurrence	/vnf_lcm_op_occs/{vnfLcmOpOccId}	GET	M	Read information about an individual VNF lifecycle management operation occurrence.
Retry operation task	/vnf_lcm_op_occs/{vnfLcmOpOccId}/retry	POST	M	Retry a VNF lifecycle management operation occurrence.
Rollback operation task	/vnf_lcm_op_occs/{vnfLcmOpOccId}/rollback	POST	M	Rollback a VNF lifecycle management operation occurrence.
Fail operation task	/vnf_lcm_op_occs/{vnfLcmOpOccId}/fail	POST	M	Mark a VNF lifecycle management operation occurrence as failed.
Cancel operation task	/vnf_lcm_op_occs/{vnfLcmOpOccId}/cancel	POST	M	Cancel a VNF lifecycle management operation occurrence.
Subscriptions	/subscriptions	POST	M	Subscribe to VNF lifecycle change notifications.
		GET	M	Query multiple subscriptions.
Individual subscription	/subscriptions/{subscriptionId}	GET	M	Read an "Individual subscription" resource.
		DELETE	M	Terminate a subscription.
Notification endpoint	(provided by API consumer)	POST	See note	Notify about VNF lifecycle change.
		GET	See note	Test the notification endpoint.

Resource name	Resource URI	HTTP Method	Cat	Meaning
VNF snapshots	/vnf_snapshots	GET	M	Query multiple VNF snapshots.
		POST	M	Create an "Individual VNF snapshot" resource.
Individual VNF snapshot	/vnf_snapshots/{vnfSnapshotInfoId}	GET	M	Read an "Individual VNF snapshot" resource.
		PATCH	M	Modify VNF snapshot resource information.
		DELETE	M	Delete an "Individual VNF snapshot" resource.
VNF state snapshot	/vnf_snapshots/{vnfSnapshotInfoId}/vnf_state_snapshot	GET	M	Fetch the content of the VNF state snapshot.
NOTE: The VNFM shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the NFVO. If the NFVO supports invoking the POST method on the "Subscriptions" resource towards the VNFM, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.				

Table 5.4.1.2-1 specifies the preconditions and postconditions applicable to the different VNF lifecycle management operations triggered by task resources.

5.3 Sequence diagrams (informative)

5.3.1 Flow of the creation of a VNF instance resource

This clause describes the procedure for the creation of an "Individual VNF instance" resource.

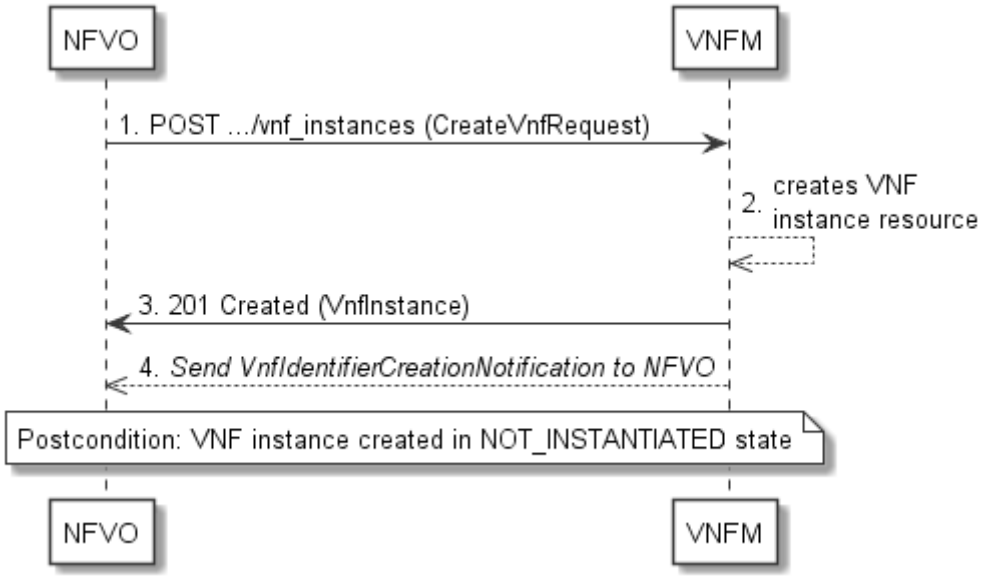

Figure 5.3.1-1: Flow of the creation of a VNF instance resource

NOTE: Due to possible race conditions, the 201 response and the VnfIdentifierCreationNotification can arrive in any order at the NFVO.

The procedure consists of the following steps as illustrated in figure 5.3.1-1:

1) The NFVO sends a POST request to the "VNF Instances" resource including in the payload body a data structure of type "CreateVnfRequest".

2) The VNFM creates a new "Individual VNF instance" resource in NOT_INSTANTIATED state, and the associated VNF instance identifier.

3) The VNFM returns a 201 Created response containing a representation of the "Individual VNF instance" resource just created by the VNFM and provides the URI of the newly-created resource in the "Location" HTTP header. See note.

4) The VNFM sends a VNF Identifier Creation Notification (see clause 5.3.9) to the NFVO to indicate the creation of the "Individual VNF instance" resource and the associated VNF instance identifier. See note.

Postcondition: Upon successful completion, a new "Individual VNF instance" resource has been created in "NOT_INSTANTIATED" state.

5.3.2 Flow of the deletion of a VNF instance resource

This clause describes the procedure for the deletion of an "Individual VNF instance" resource.

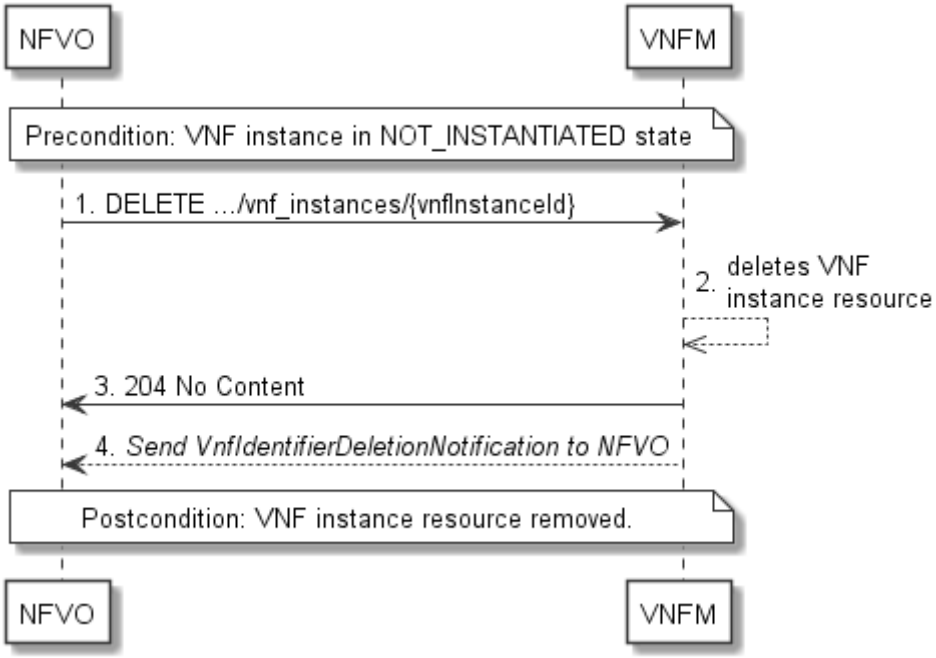

Figure 5.3.2-1: Flow of the deletion of a VNF instance resource

NOTE: Due to possible race conditions, the 204 response and the VnfIdentifierDeletionNotification can arrive in any order at the NFVO.

Precondition: The resource representing the VNF instance to be deleted needs to be in NOT_INSTANTIATED state.

The procedure consists of the following steps as illustrated in figure 5.3.2-1:

1) NFVO sends a DELETE request to the "Individual VNF Instance" resource.

2) The VNFM deletes the "Individual VNF instance" resource and the associated VNF instance identifier.

3) The VNFM returns a "204 No Content" response with an empty payload body. See note.

4) The VNFM sends to the NFVO a VnfIdentifierDeletionNotification to indicate the deletion of the "Individual VNF instance" resource and the associated VNF instance identifier. See note.

Postcondition: The resource representing the VNF instance has been removed from the list of VNF instance resources.

Error handling: If the "Individual VNF instance" resource is not in NOT_INSTANTIATED state, the VNFM rejects the deletion request.

5.3.3 Flow of VNF lifecycle management operations triggered by task resources

This clause describes the general sequence for VNF Lifecycle Management operations that operate on VNF instance resource and are triggered by task resources. The flows for these operations are very similar. The differences between the individual operations are covered in table 5.4.1.2-1.

This flow is applicable to the following operations:

- Instantiate VNF
- Scale VNF
- Scale VNF to Level
- Change VNF flavour
- Operate VNF
- Heal VNF
- Change external VNF connectivity
- Change current VNF package
- Create VNF snapshot
- Revert to VNF snapshot
- Terminate VNF

Figure 5.3.3-1 illustrates the general lifecycle management flow. Placeholders in this flow allow for differentiating between the operations and are marked with double angular brackets "<<…>>".

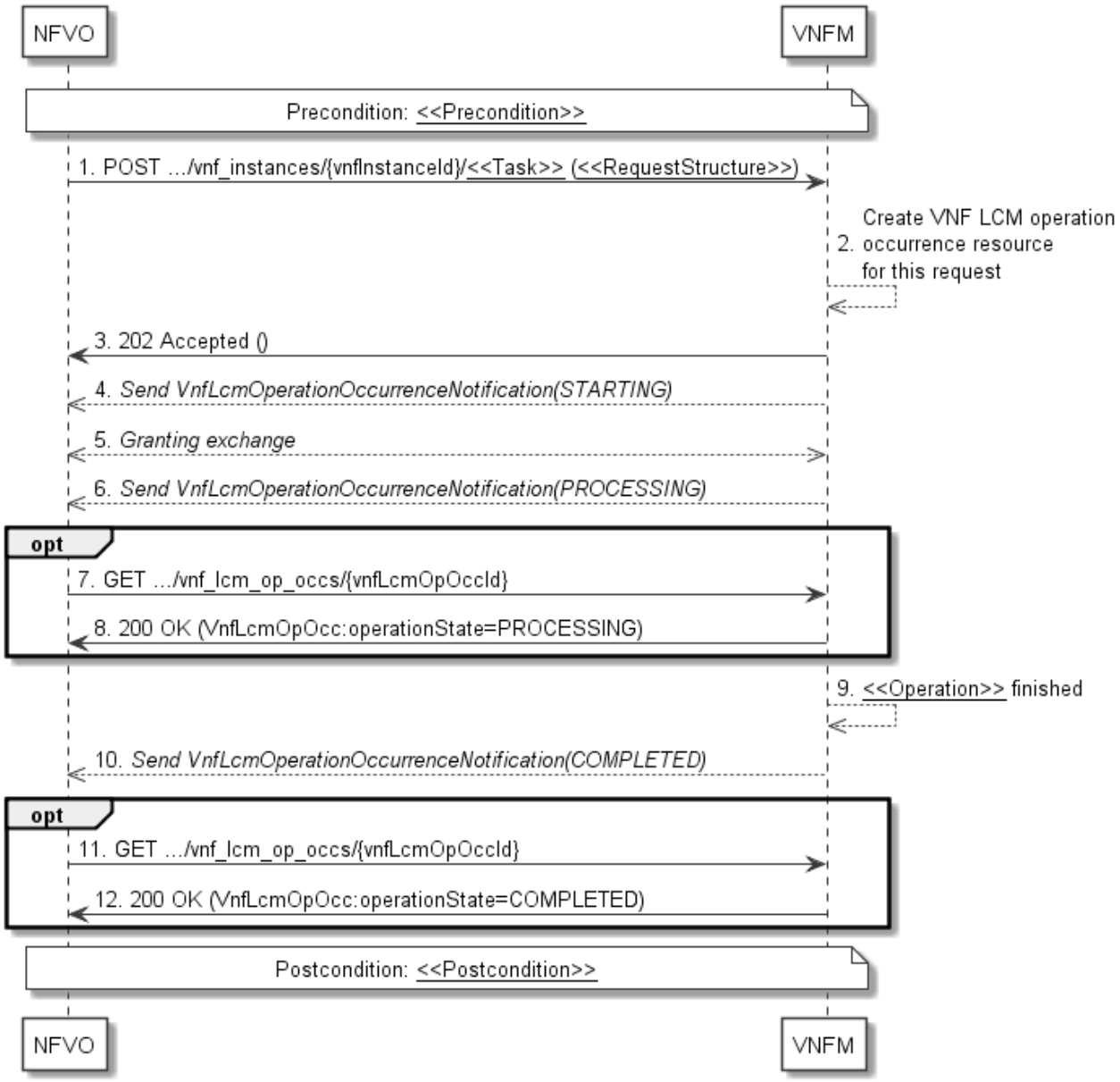

Figure 5.3.3-1: General flow of VNF lifecycle management operations triggered by task resources

NOTE: Due to possible race conditions, the 202 response, the "STARTING" VnfLcmOperationOccurrenceNotification and the request of the Granting exchange can arrive in any order at the NFVO.

Precondition: The precondition depends on the actual operation and is described by the template parameter <<Precondition>>. Table 5.4.1.2-1 specifies the applicable precondition.

A VNF lifecycle operation, as illustrated in figure 5.3.3-1, consists of the following steps:

1) The NFVO sends a POST request to the <<Task>> resource that represents the lifecycle operation to be executed on the VNF instance and includes in the payload body a data structure of type <<RequestStructure>>. The name <<Task>> of the task resource and the <<RequestStructure>> depend on the operation and are described in table 5.4.1.2-1.

2) The VNFM creates a new "Individual VNF LCM operation occurrence" resource for the request.

3) The VNFM returns a "202 Accepted" response with an empty payload body and a "Location" HTTP header that points to the new "Individual VNF LCM operation occurrence" resource, i.e. it includes the URI of that resource which is "…/vnf_lcm_op_occs/{vnfLcmOpOccId}". See note.

4) The VNFM sends to the NFVO a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate the start of the lifecycle management operation occurrence with the "STARTING" state. See note.

5) VNFM and NFVO exchange granting information (see VNF Lifecycle Operation Granting interface, clause 9.3). See note.

6) The VNFM sends to the NFVO a VNF lifecycle management operation occurrence notification (see clause 5.3.8) to indicate that the VNF LCM operation occurrence enters the "PROCESSING" state.

7) If desired, the NFVO can poll the "Individual VNF LCM operation occurrence" resource to obtain information about the ongoing operation by sending a GET request to the resource that represents the VNF LCM operation occurrence.

8) In the response to that request, the VNFM returns to the NFVO information of the operation, such as the operation status, by providing in the payload body a data structure of type "VnfLcmOpOcc".

9) The VNFM has finished the operation <<Operation>>.

10) The VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate the completion of the lifecycle management operation occurrence with the success state "COMPLETED".

11) If desired, the NFVO can send a new GET request to the "Individual VNF LCM operation occurrence" resource.

12) In the response to that request, the VNFM returns to the NFVO information about the result of the operation, by providing in the payload body a data structure of type "VnfLcmOpOcc".

Postcondition: The postcondition depends on the actual operation and is described by the template parameter <<Postcondition>>. Table 5.4.1.2-1 specifies the applicable postcondition.

Error handling: If the VNF lifecycle management operation fails, error information is provided in the notification message that reports the erroneous completion of the procedure and is also available in the resource that represents the actual VNF lifecycle management operation occurrence related to this VNF lifecycle management operation.

Table 5.4.1.2-1 defines how the flow described above is parameterized for the different VNF lifecycle management operations.

5.3.4 Flow of automatic invocation of VNF scaling and VNF healing

This clause describes the sequence for the automatic invocation of "Scale VNF", "Scale VNF to Level" and "Heal VNF" operations by the VNFM, also known as "auto-scale" and "auto-heal". The criteria based on which the VNFM decides when to invoke an automatic scaling or automatic healing are outside the scope of the present document and can include certain changes in monitoring parameters that are monitored by the VNFM by PM jobs or thresholds, changes in VNF indicator values that are polled by the VNFM or that are reported to the VNFM via "VnfIndicatorValueChangeNotification" messages. Auto-scaling and auto-healing can be enabled and disabled by the NFVO by modifying the appropriate "isAutoscaleEnabled" and "isAutohealEnabled" configurable properties of the VNF using the sequence flow according to clause 5.3.6.

This flow is applicable to the automatic invocation of the following operations:

- Scale VNF
- Scale VNF to Level
- Heal VNF

Figure 5.3.4-1 illustrates the flow.

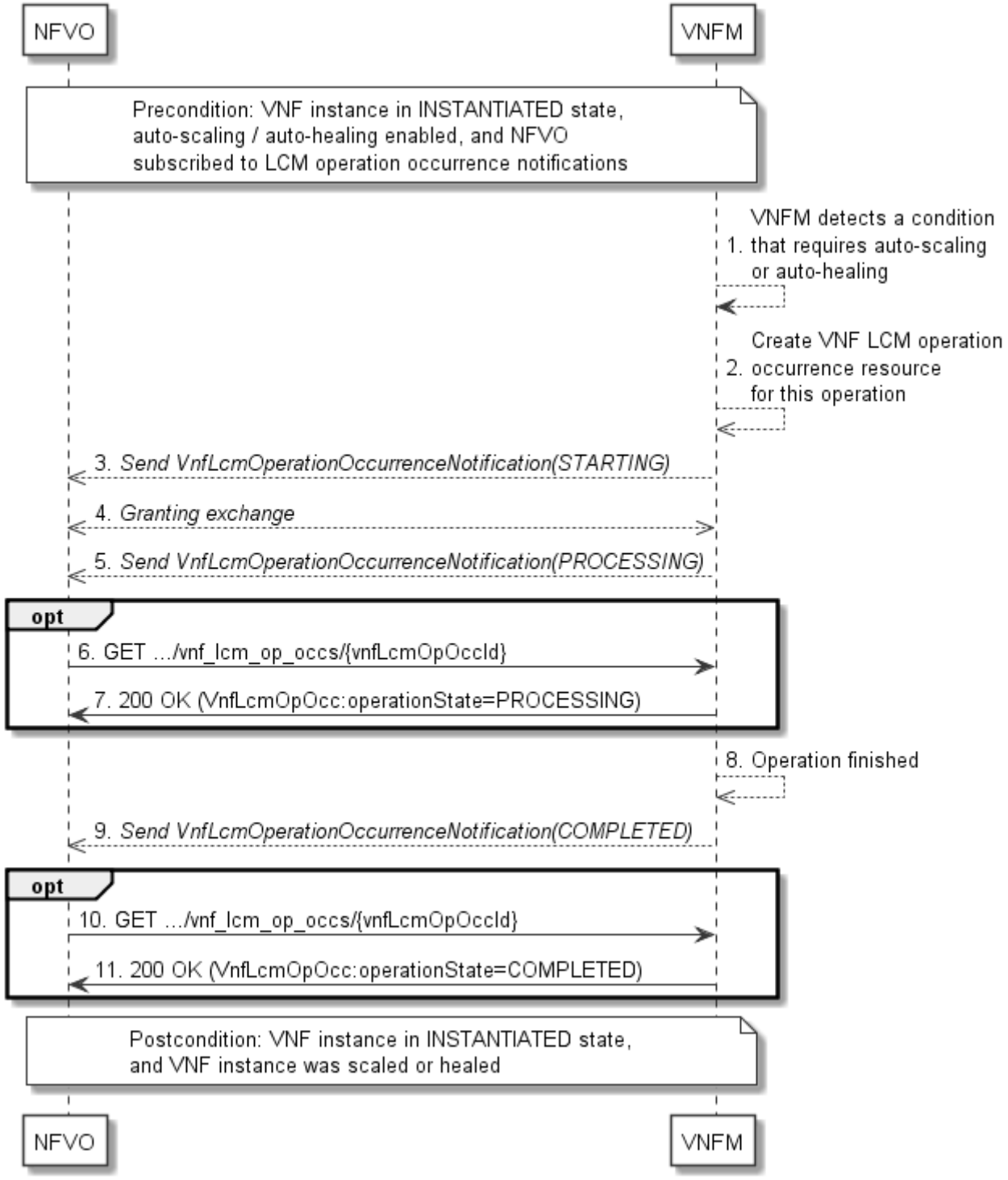

Figure 5.3.4-1: Flow of auto-scaling and auto-healing

NOTE: Due to possible race conditions, the "STARTING" VnfLcmOperationOccurrenceNotification and the request of the Granting exchange can arrive in any order at the NFVO.

Precondition: The VNF instance is in INSTANTIATED state, auto-scaling/auto-healing is enabled, and the NFVO is subscribed to VNF LCM operation occurrence notifications.

The automatic invocation of a VNF scaling or VNF healing operation, as illustrated in figure 5.3.4-1, consists of the following steps:

1) The VNFM detects a condition that triggers auto-scaling (Scale VNF or Scale VNF To Level) or auto-healing (Heal VNF) of the VNF and selects the appropriate parameters for the operation.

2) The VNFM creates an "Individual VNF LCM operation occurrence" resource for the operation.

3) The VNFM sends to the NFVO a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate the start of the lifecycle management operation occurrence. See note.

4) VNFM and NFVO exchange granting information (see VNF Lifecycle Operation Granting interface, clause 9.3). See note.

5) The VNFM sends to the NFVO a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate that the VNF LCM operation occurrence enters the "PROCESSING" state.

6) If desired, the NFVO can poll the "Individual VNF lifecycle management operation occurrence" resource to obtain information about the ongoing operation by sending a GET request to the resource that represents the VNF lifecycle management operation occurrence.

7) In the response to that request, the VNFM returns to the NFVO information of the operation, such as the operation status, by providing in the payload body a data structure of type "VnfLcmOpOcc".

8) The VNFM has finished the operation.

9) The VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate the completion of the lifecycle management operation occurrence.

10) If desired, the NFVO can send a new GET request to the "Individual VNF lifecycle management operation occurrence" resource.

11) In the response to that request, the VNFM returns to the NFVO information about the result of the operation, by providing in the payload body a data structure of type "VnfLcmOpOcc".

Postcondition: The VNF instance in INSTANTIATED state, and the VNF instance has been scaled or healed as appropriate.

Error handling: If the VNF lifecycle management operation fails, error information is provided in the notification message that reports the erroneous completion of the procedure and is also available in the resource that represents the actual VNF lifecycle management operation occurrence related to this VNF lifecycle management operation.

5.3.5 Flow of the Query VNF operation

This clause describes a sequence for querying/reading information about a VNF instance.

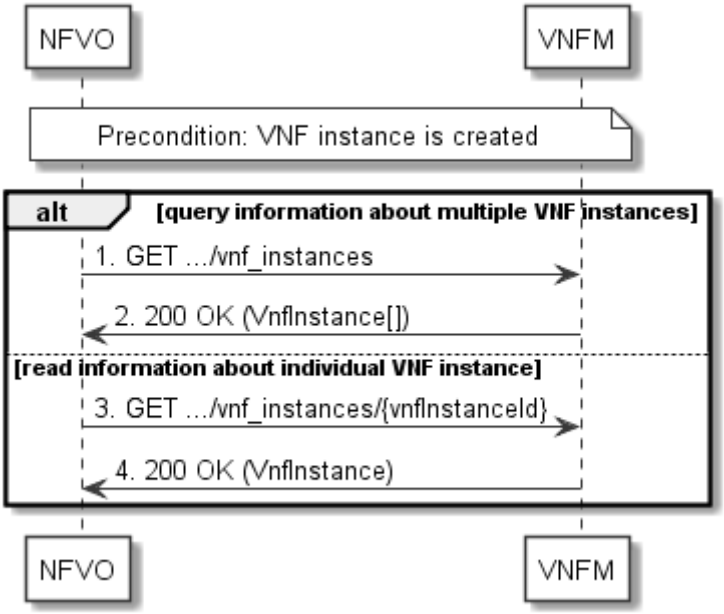

Figure 5.3.5-1: Flow of VNF instance query/read

Precondition: The resource representing the VNF instance has been created.

VNF instance query, as illustrated in figure 5.3.5-1, consists of the following steps:

1) If the NFVO intends to query all VNF instances, it sends a GET request to the "VNF instances" resource.

2) The VNFM returns a "200 OK" response to the NFVO and includes zero or more data structures of type "VnfInstance" in the payload body.

3) If the NFVO intends to read information about a particular VNF instance, it sends a GET request to the "Individual VNF instance" resource, addressed by the appropriate VNF instance identifier in its resource URI.

4) The VNFM returns a "200 OK" response to the NFVO and includes one data structure of type "VnfInstance" in the payload body.

Error handling: In case of failure, appropriate error information is provided in the response.

5.3.6 Flow of the Modify VNF Information operation

This clause describes a sequence for updating information about a VNF instance.

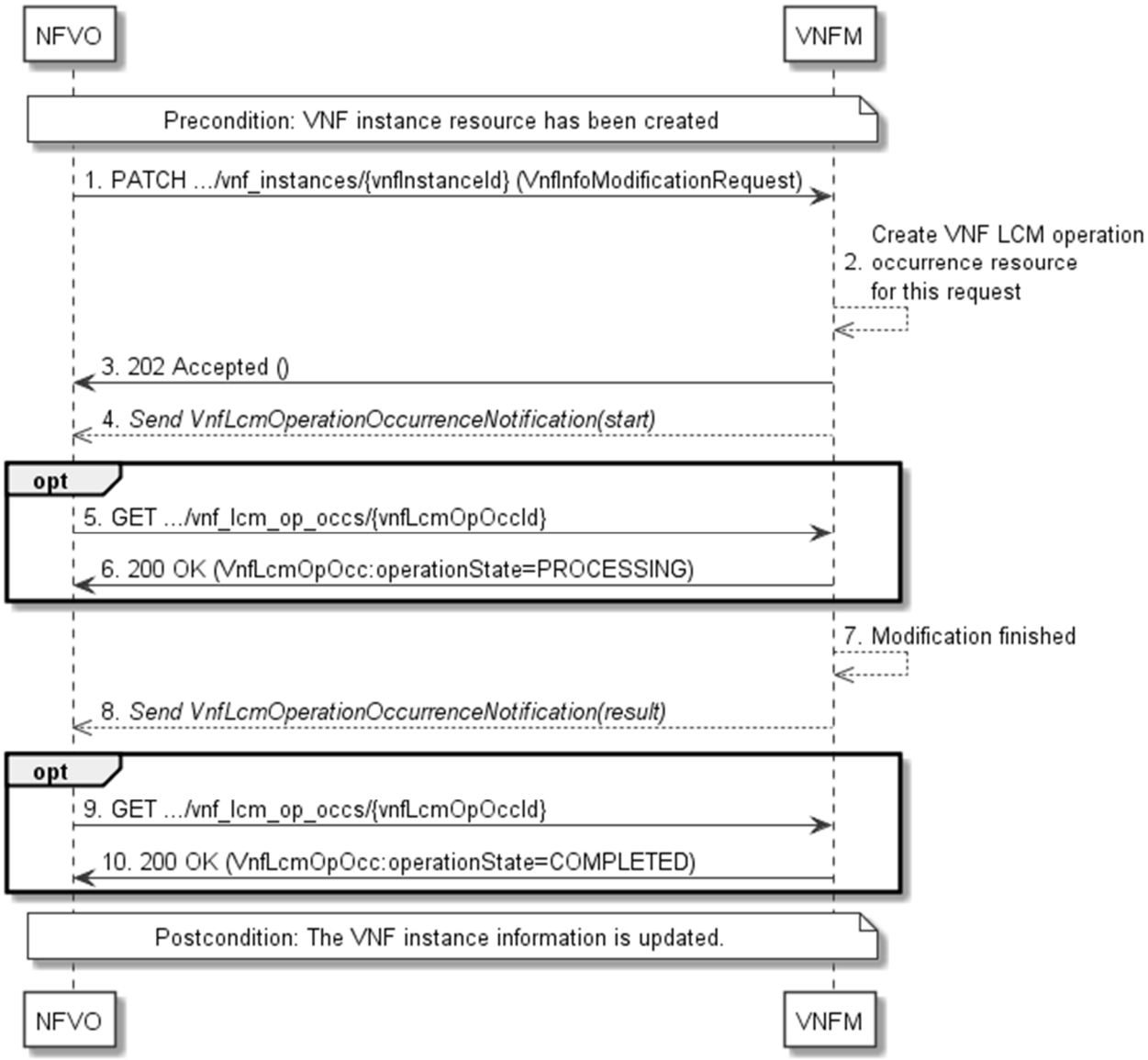

Figure 5.3.6-1: Flow of the modification of VNF instance information

NOTE: Due to possible race conditions, the 202 response and the VnfLcmOperationOccurrenceNotification can arrive in any order at the NFVO.

Precondition: The resource representing the VNF instance has been created.

Updating the VNF instance information, as illustrated in figure 5.3.6-1, consists of the following steps:

1) The NFVO sends a PATCH request to the "Individual VNF instance" resource of the VNF instance that is to be operated and includes in the payload body a data structure of type "VnfInfoModificationRequest".

2) The VNFM creates an "Individual VNF LCM operation occurrence" resource for the request.

3) The VNFM returns a "202 Accepted" response with an empty payload body and a "Location" HTTP header that points to the new "Individual VNF LCM operation occurrence" resource, i.e. it includes the URI of that resource which is "…/vnf_lcm_op_occs/{vnfLcmOpOccId}". See note.

4) The VNFM sends to the NFVO a lifecycle management operation occurrence notification (see clause 5.3.9) to indicate the start of the operation. See note.

5) If desired, the NFVO can poll the "Individual VNF LCM operation occurrence" resource to obtain information about the ongoing operation by sending a GET request to the resource that represents the VNF lifecycle management operation occurrence.

6) In the response to that request, the VNFM returns to the NFVO information of the operation, such as the operation status, by providing in the payload body a data structure of type "VnfLcmOpOcc".

7) The VNFM has finished the modification operation.

8) The VNFM sends a lifecycle management operation occurrence notification (see clause 5.3.9) to indicate the completion of the operation, and the performed changes.

9) If desired, the NFVO can send a new GET request to the "Individual VNF LCM operation occurrence" resource.

10) In the response to that request, the VNFM returns to the NFVO information about the result of the operation, by providing in the payload body a data structure of type "VnfLcmOpOcc".

Postcondition: Upon successful completion, information of the VNF instance is updated.

Error handling: If the updating of VNF instance information fails, error information is provided in the notification message that reports the erroneous completion of the procedure and is also available in the resource that represents the actual VNF lifecycle management operation occurrence related to this VNF LCM operation.

5.3.7 Flow of the Get Operation Status operation

This clause describes a sequence for obtaining the status of a VNF lifecycle management operation occurrence.

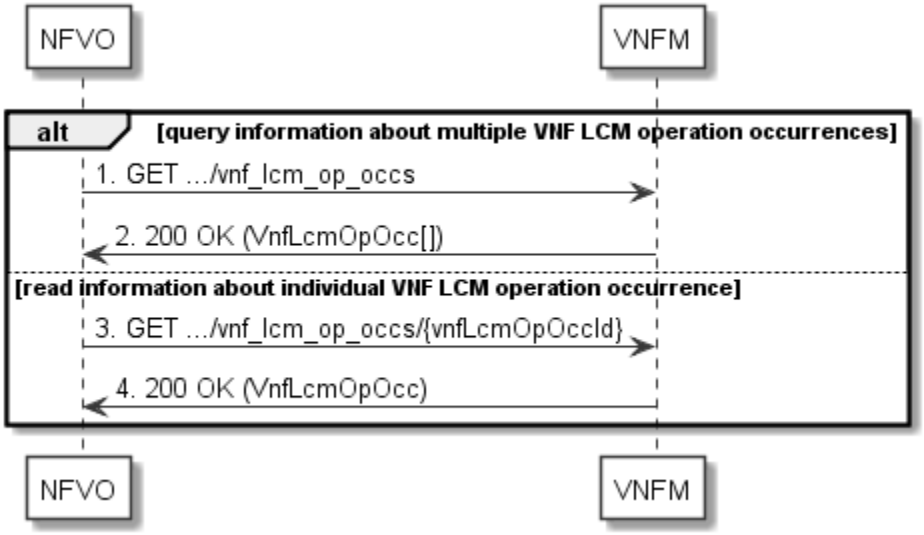

Figure 5.3.7-1: Flow of Get VNF lifecycle operation status

Obtaining the VNF lifecycle operation status, as illustrated in figure 5.3.7-1, consists of the following steps:

1) If the NFVO intends to query all VNF lifecycle management operation occurrences, it sends a GET request to the "VNF LCM operation occurrences" resource.

2) The VNFM returns a "200 OK" response to the NFVO and includes zero or more data structures of type "VnfLcmOpOcc" in the payload body.

3) If the NFVO intends to read information about a particular VNF LCM operation occurrence, it sends a GET request to the "Individual VNF LCM operation occurrence" resource, addressed by the appropriate VNF LCM operation occurrence identifier in its resource URI.

4) The VNFM returns a "200 OK" response to the NFVO and includes one data structure of type "VnfLcmOpOcc" in the payload body.

Error handling: In case of failure, appropriate error information is provided in the response.

5.3.8 Flow of managing subscriptions

This clause describes the procedure for creating, querying/reading and terminating subscriptions to notifications related to VNF lifecycle management.

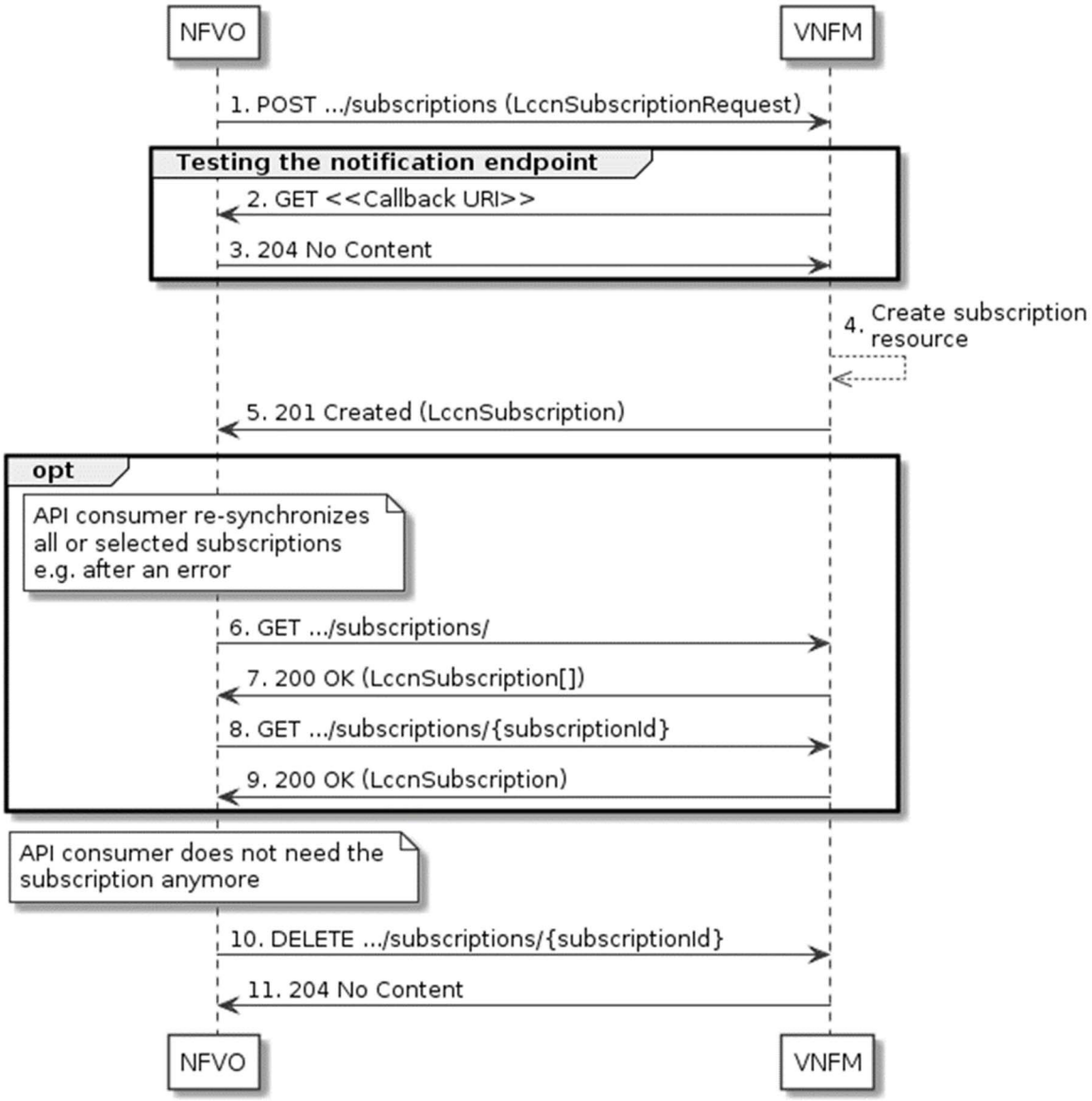

Figure 5.3.8-1: Flow of managing subscriptions

The procedure consists of the following steps as illustrated in figure 5.3.8-1:

1) The NFVO sends a POST request to the "Subscriptions" resource including in the payload body a data structure of type "LccnSubscriptionRequest". That data structure contains filtering criteria and a callback URI to which the VNFM will subsequently send notifications about events that match the filter.

2) To test the notification endpoint that has been registered by the NFVO as part of the subscription, the VNFM sends a GET request to the notification endpoint URI.

3) The NFVO returns a "204 No Content" response to indicate success.

4) The VNFM creates a new subscription to notifications related to VNF lifecycle changes, and an "Individual subscription" resource that represents this subscription.

5) The VNFM returns a 201 Created response containing a data structure of type "LccnSubscription" representing the "Individual subscription" resource just created by the VNFM and provides the URI of the newly-created resource in the "Location" HTTP header.

6) If desired, e.g. to recover from an error situation, the NFVO can query information about its subscriptions by sending a GET request to the resource representing the subscriptions.

7) In that case, the VNFM returns a "200 OK" response that contains zero or more representations of all existing subscriptions that were created by the NFVO.

8) If desired, e.g. to recover from an error situation, the NFVO can read information about a particular subscription by sending a GET request to the resource representing that individual subscription.

9) In that case, the VNFM returns a "200 OK" response that contains a representation of that individual subscription.

10) If the NFVO does not need the subscription anymore, it terminates the subscription by sending a DELETE request to the resource that represents the individual subscription to remove.

11) The NFVO acknowledges the successful termination of the subscription by returning a "204 No Content" response.

Error handling: The VNFM rejects a subscription if the subscription information is not valid: endpoint cannot be reached, subscription information is malformed, etc.

5.3.9 Flow of sending notifications

This clause describes the procedure for sending notifications.

NOTE 1: Notifications merely report to subscribed NFV-MANO entities the state changes of a VNF instance and/or LCM operation occurrence. They are triggered during the execution of the operation's flow or at its end but have no impact on the course of the procedure that has triggered them or on the state of the VNF instance. If this flow is invoked as part of another flow, the invoking procedure does not wait for the acknowledgement of the delivery of the notification.

NOTE 2: Race conditions between LCM operation requests/responses on one hand and notification delivery requests/responses on the other hand can occur as these are delivered through different HTTP connections.

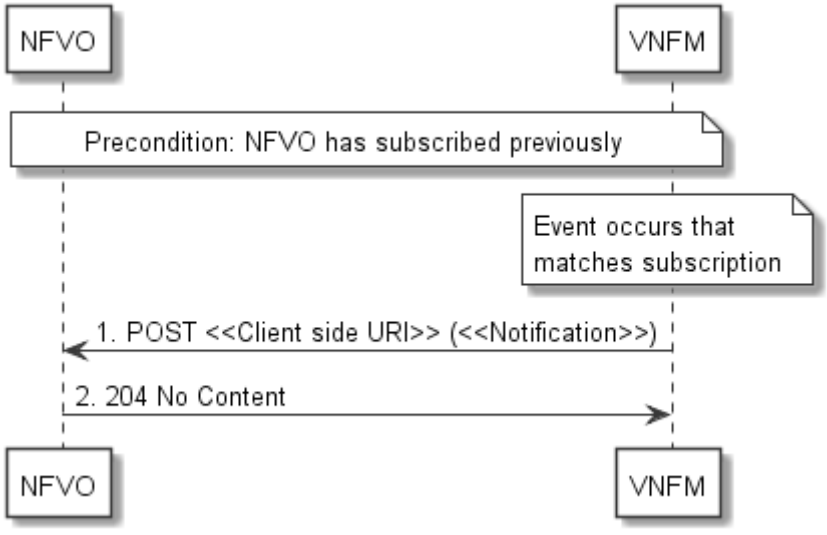

Figure 5.3.9-1: Flow of sending notifications

The procedure consists of the following steps as illustrated in figure 5.3.9-1.

Precondition: The NFVO has subscribed previously to notifications related to VNF lifecycle management.

1) If an event occurs that matches the filtering criteria which are part of the subscription, the VNFM generates a notification that includes information about the event and sends it in the body of a POST request to the URI which the NFVO has registered as part of the subscription request. The variable <<Notification>> in the flow is a placeholder for the different types of notifications that can be sent by this API (see clauses 5.5.2.17 to 5.5.2.19).

2) The NFVO acknowledges the successful delivery of the notification by returning a "204 No Content" response.

Error handling: If the VNFM does not receive the "204 No Content" response from the NFVO, it can retry sending the notification.

5.3.10 Flow of retrying a VNF lifecycle management operation

This clause describes a sequence for retrying a VNF lifecycle management operation occurrence that is represented by an "Individual VNF LCM operation occurrence" resource. Retry is used if an operation is in FAILED_TEMP state, and there is reason to believe that the operation will eventually succeed when retried, for instance because obstacle that led to an error during the execution of the LCM operation have been removed by an automated procedure, or by manual intervention. The "retry" operation is also called "idempotent retry" because it is possible to invoke retry multiple times, without side effects.

A comprehensive description of the handling of VNF lifecycle management errors is provided in clause 5.6.

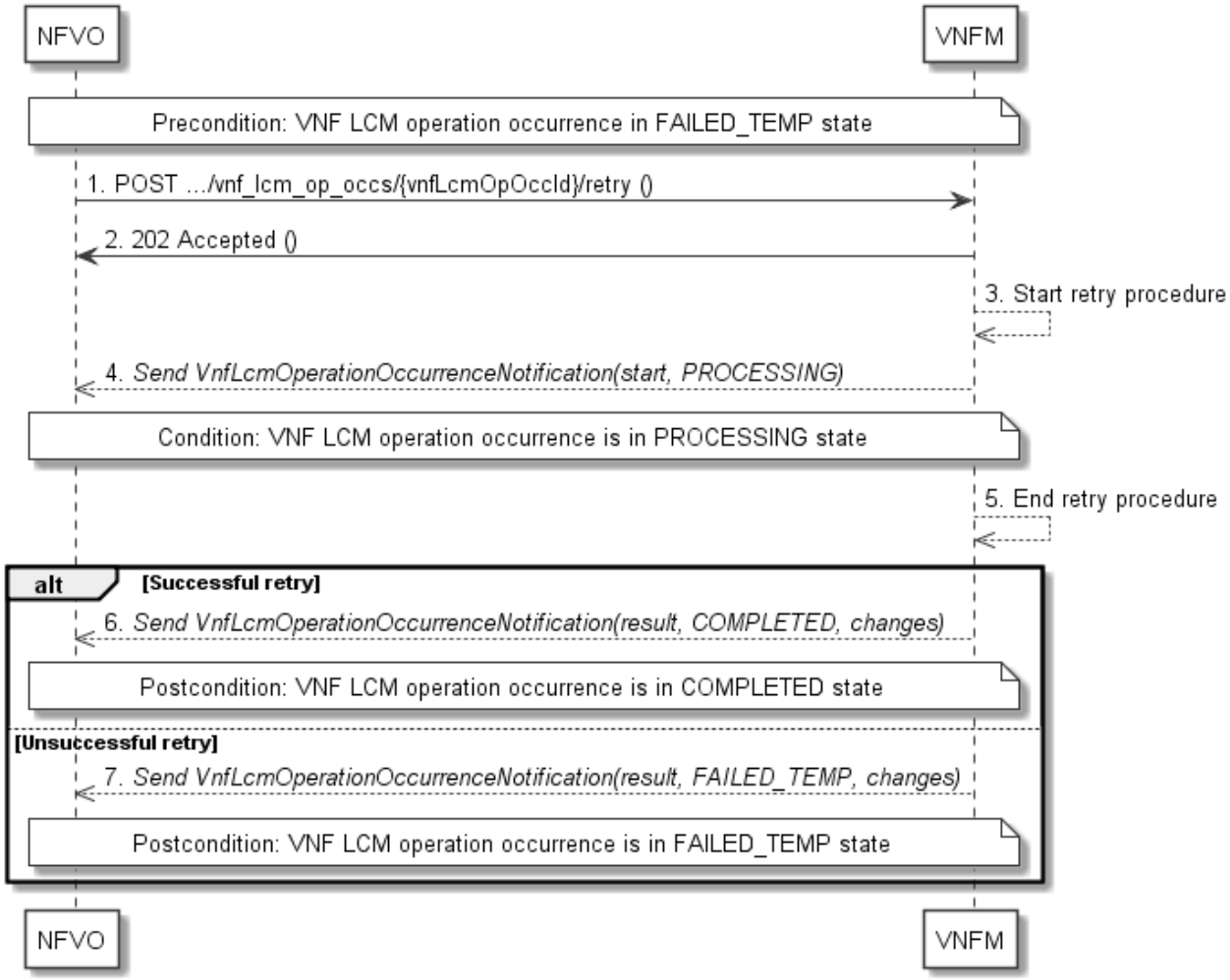

Figure 5.3.10-1: Flow of retrying a VNF lifecycle management operation

NOTE: Due to possible race conditions, the 202 response and the "PROCESSING" VnfLcmOperationOccurrenceNotification can arrive in any order at the NFVO.

Precondition: The VNF lifecycle management operation occurrence is in FAILED_TEMP state.

Retrying a VNF lifecycle operation, as illustrated in figure 5.3.10-1, consists of the following steps:

1) The NFVO sends a POST request with an empty body to the "Retry operation task" resource of the VNF LCM operation occurrence that is to be retried.

2) The VNFM returns a "202 Accepted" response. See note.

3) The VNFM starts the retry procedure.

4) The VNFM sends a VNF lifecycle management operation occurrence notification of type "start" to indicate that the VNF LCM operation occurrence enters the "PROCESSING" state. See note.

5) The VNFM finishes the retry procedure.

6) On successful retry, the VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate successful completion of the operation and to inform the NFVO about the virtualised resources changes.

7) On unsuccessful retry, the VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate an intermediate error (retry failed) of the operation and to inform the NFVO about the virtualised resources changes.

Postcondition: The VNF lifecycle management operation occurrence is in one of the following states: FAILED_TEMP, COMPLETED. COMPLETED is a terminal state (see clause 5.6.2.2).

Error handling: The operation is rejected in case the "Individual VNF LCM operation occurrence" resource is in any other state than FAILED_TEMP, or in case Retry is not supported by for the particular VNF LCM operation for the particular VNF.

5.3.11 Flow of rolling back a VNF lifecycle management operation

This clause describes a sequence for rolling back a VNF lifecycle management operation occurrence that is represented by an "Individual VNF LCM operation occurrence" resource. Rollback can be used for example if an operation is in FAILED_TEMP state, and there is no reason to believe that retrying the operation will eventually succeed.

A comprehensive description of the handling of VNF lifecycle management errors is provided in clause 5.6.

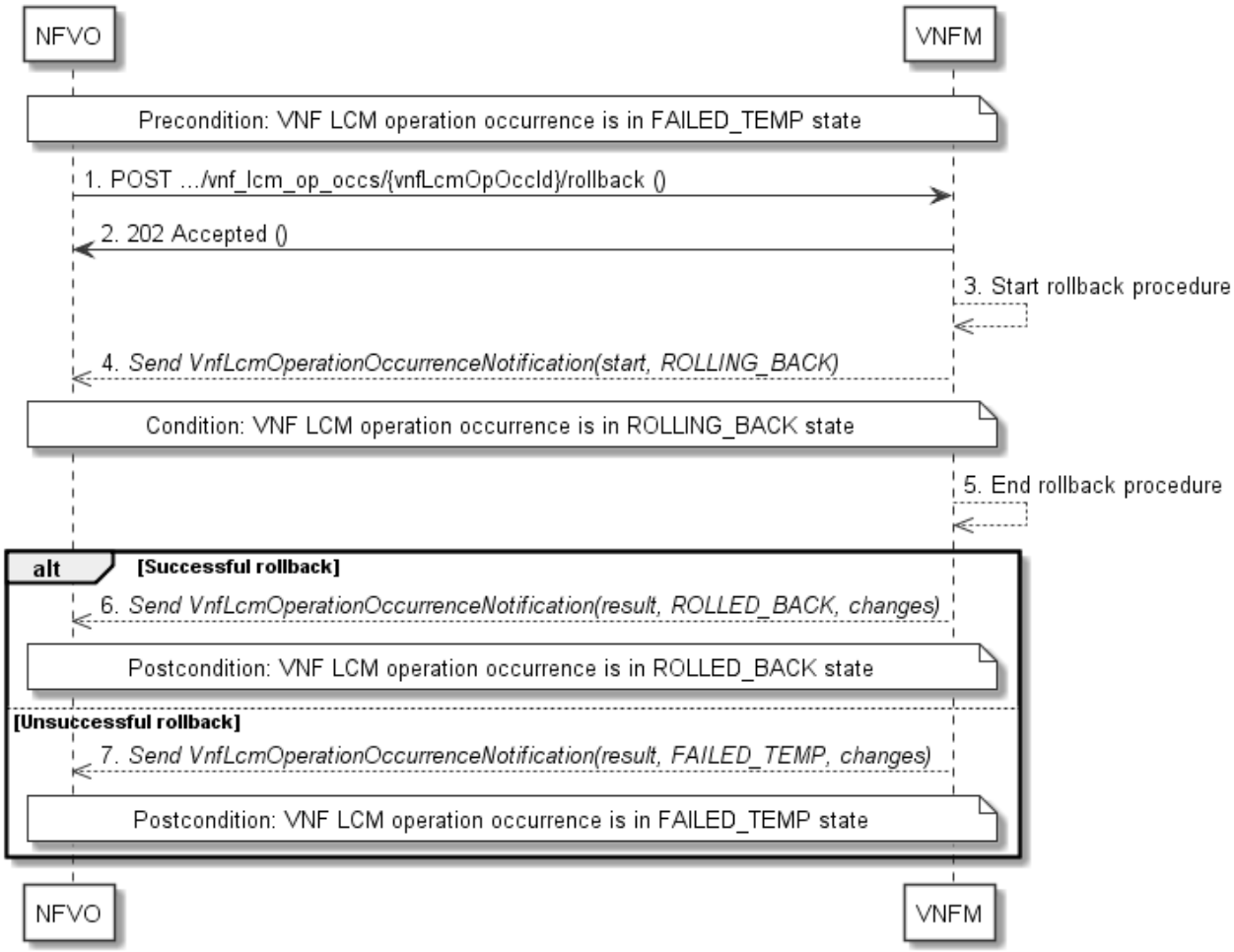

Figure 5.3.11-1: Flow of rolling back a VNF lifecycle management operation

NOTE: Due to possible race conditions, the 202 response and the "ROLLING_BACK" VnfLcmOperationOccurrenceNotification can arrive in any order at the NFVO.

Precondition: The VNF lifecycle management operation occurrence is in FAILED_TEMP state.

Initiating the rollback of a VNF lifecycle management operation, as illustrated in figure 5.3.11-1, consists of the following steps:

1) The NFVO sends a POST request with an empty body to the "Rollback operation task" resource of the VNF LCM operation occurrence that is to be rolled back.

2) The VNFM returns a "202 Accepted" response. See note.

3) The VNFM starts the rollback procedure.

4) The VNFM sends a VNF lifecycle management operation occurrence notification of type "start" to indicate that the VNF LCM operation occurrence enters the "ROLLING_BACK" state. See note.

5) The VNFM finishes the rollback procedure.

6) On successful rollback, the VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate successful completion of the operation and to inform the NFVO about the virtualised resources changes.

7) On unsuccessful retry, the VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate an intermediate error (rollback failed) of the operation and to inform the NFVO about the virtualised resources changes.

Postcondition: The VNF lifecycle management operation occurrence is in one of the following states: FAILED_TEMP, ROLLED_BACK. ROLLED_BACK is a terminal state (see clause 5.6.2.2).

Error handling: The operation is rejected in case the VNF lifecycle management operation occurrence is in any other state than FAILED_TEMP, or in case Rollback is not supported for the particular VNF LCM operation for the particular VNF.

5.3.12 Flow of failing a VNF lifecycle management operation

This clause describes a sequence for declaring as "failed" a VNF lifecycle management operation occurrence that is represented by an "Individual VNF LCM operation occurrence" resource. If there is neither an assumption that the operation can eventually succeed after further retries, nor that the operation can be successfully rolled back, the operation can be declared as "failed". This will unblock the invocation of other LCM operations, such as HealVnf, or non-graceful VNF termination, on the affected VNF instance.

A comprehensive description of the handling of VNF lifecycle management errors is provided in clause 5.6.

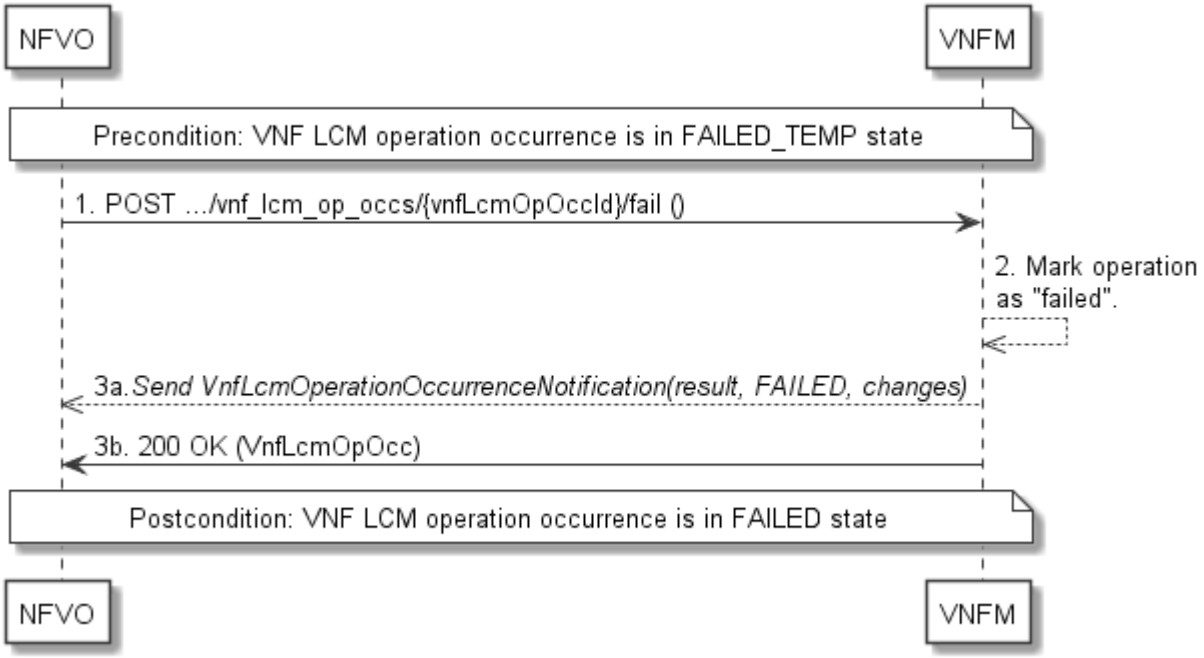

Figure 5.3.12-1: Flow of declaring a VNF lifecycle management operation as failed

NOTE: Due to possible race conditions, the 200 response and the "FAILED" VnfLcmOperationOccurrenceNotification can arrive in any order at the NFVO.

Precondition: The VNF lifecycle management operation occurrence is in FAILED_TEMP state.

Declaring a VNF lifecycle management operation as failed, as illustrated in figure 5.3.12-1, consists of the following steps:

1) The NFVO sends a POST request with an empty body to the "Fail operation task" resource of the VNF LCM operation occurrence that is to be marked as failed.

2) The VNFM marks the operation as failed.

3) The VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate the final failure of the operation and to inform the NFVO about the virtualised resources changes. Furthermore, it returns a "200 OK" response and includes in the body a VnfLcmOpOcc structure. The order in which the response and the notification arrive at the NFVO is not defined. See note.

Postcondition: The VNF lifecycle management operation occurrence is FAILED state. This is a terminal state (see clause 5.6.2.2).

Error handling: The operation is rejected in case the VNF lifecycle management operation occurrence is in any other state than FAILED_TEMP.

5.3.13 Flow of cancelling a VNF lifecycle management operation

This clause describes a sequence for cancelling an ongoing VNF LCM operation occurrence, or a rollback of a VNF LCM operation occurrence. The possibility and timing of cancellation is dependent on the implementation of the underlying lifecycle management operation.

A comprehensive description of the handling of VNF lifecycle management errors is provided in clause 5.6.

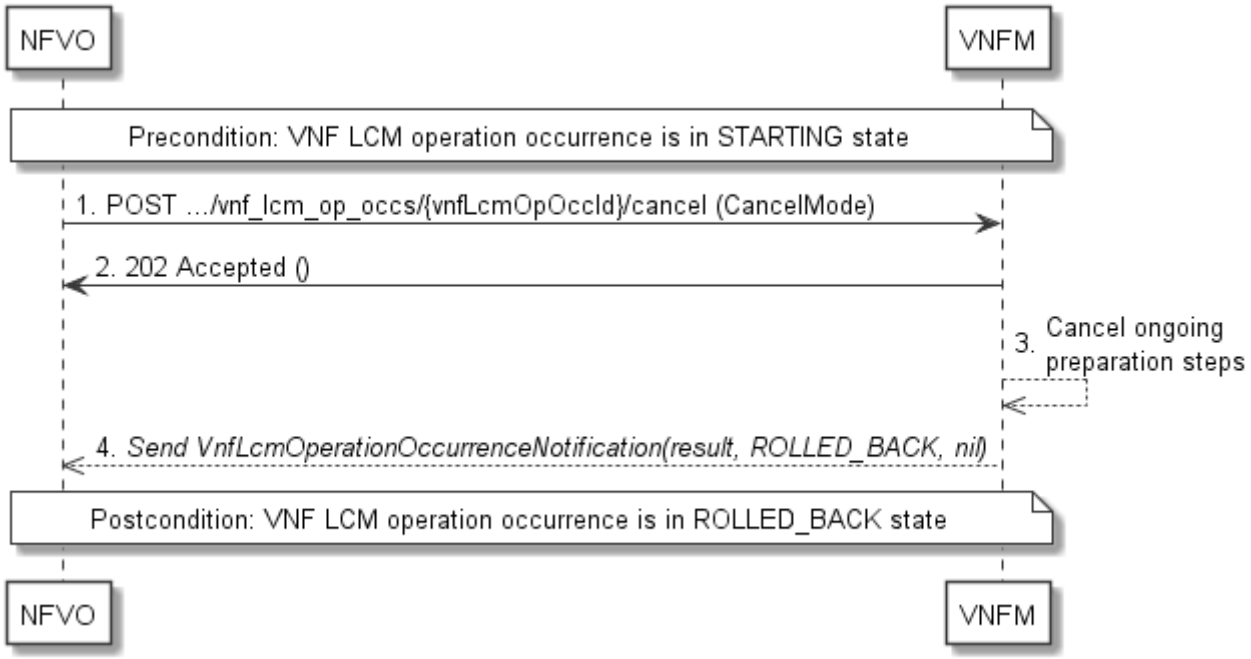

Figure 5.3.13-1: Flow of cancelling a VNF lifecycle management operation in "STARTING" state

NOTE 1: Due to possible race conditions, the 202 response and the "ROLLED_BACK" VnfLcmOperationOccurrenceNotification can arrive in any order at the NFVO.

Precondition: The VNF lifecycle management operation occurrence is in STARTING state.

Cancelling a VNF lifecycle operation when it is in STARTING state, as illustrated in figure 5.3.13-1, consists of the following steps:

1) The NFVO sends a POST request with "CancelMode" structure in the body to the "Cancel operation task" resource of the VNF LCM operation occurrence that is to be cancelled.

2) The VNFM returns a "202 Accepted" response. See note 1.

3) The VNFM cancels the ongoing preparation operations.

4) The VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate an intermediate error (cancelled) of the operation, and to inform the NFVO that there were no virtualised resources changes. See note 1.

Postcondition: The VNF lifecycle management operation occurrence is in ROLLED_BACK state.

Error handling: The operation is rejected in case the VNF lifecycle operation occurrence is in any other state than STARTING.

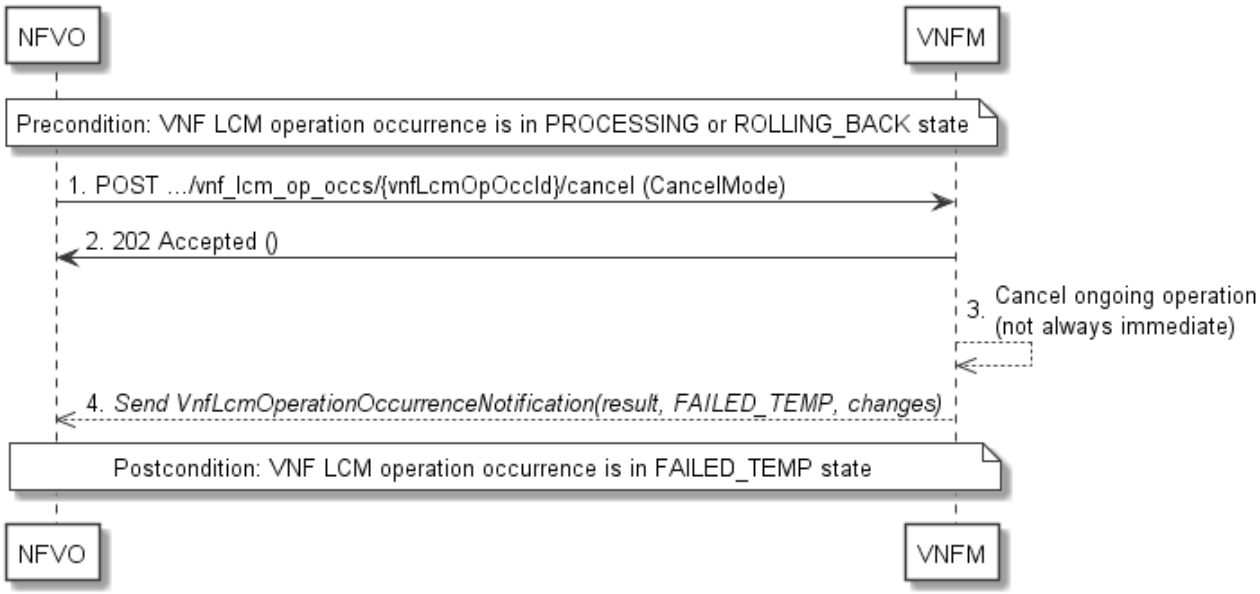

Figure 5.3.13-2: Flow of cancelling a VNF lifecycle management operation in "PROCESSING" or "ROLLING_BACK" state

NOTE 2: Due to possible race conditions, the 202 response and the "FAILED_TEMP" VnfLcmOperationOccurrenceNotification can arrive in any order at the NFVO.

Precondition: The VNF lifecycle management operation occurrence is in PROCESSING or ROLLING_BACK state.

Cancelling a VNF lifecycle operation when it is in "PROCESSING" or "ROLLING_BACK" state, as illustrated in figure 5.3.13-2, consists of the following steps:

1) The NFVO sends a POST request with a "CancelMode" structure in the body to the "Cancel operation task" resource of the VNF LCM operation occurrence that is to be cancelled.

2) The VNFM returns a "202 Accepted" response. See note 2.

3) The VNFM cancels the ongoing LCM operation. This can take some time.

4) The VNFM sends a VNF lifecycle management operation occurrence notification (see clause 5.3.9) to indicate an intermediate error (cancelled) of the operation and to inform the NFVO about the virtualised resources changes. See note 2.

Postcondition: The VNF lifecycle management operation occurrence is FAILED_TEMP state.

Error handling: The operation is rejected in case the VNF lifecycle management operation occurrence is in any other state than PROCESSING or ROLLING_BACK, or in case Cancel is not supported for the particular VNF LCM operation for the particular VNF.

5.3.14 Flow of creation of a VNF snapshot resource

This clause describes the procedure for the creation of an "Individual VNF snapshot" resource.

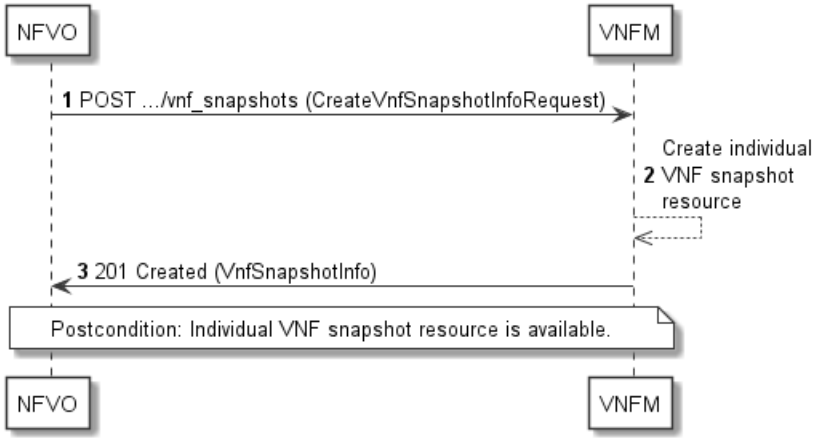

Figure 5.3.14-1: Flow of creation of a VNF snapshot resource

The procedure consists of the following steps as illustrated in figure 5.3.14-1:

1) The NFVO sends a POST request to the "VNF snapshots" resource and includes in the payload body a data structure of type "CreateVnfSnapshotInfoRequest".

2) The VNFM creates a new "Individual VNF snapshot" resource.

3) The VNFM returns a "201 Created" response containing a representation of the "Individual VNF snapshot" resource and a "Location" HTTP header that points to the new "Individual VNF snapshot" resource.

Postcondition: The resource representing the VNF snapshot has been created and is available.

Error handling: In case of failure, appropriate error information is provided in the response.

5.3.15 Flow of the Query VNF Snapshot operation

This clause describes a sequence for querying/reading information about one or more VNF snapshots.

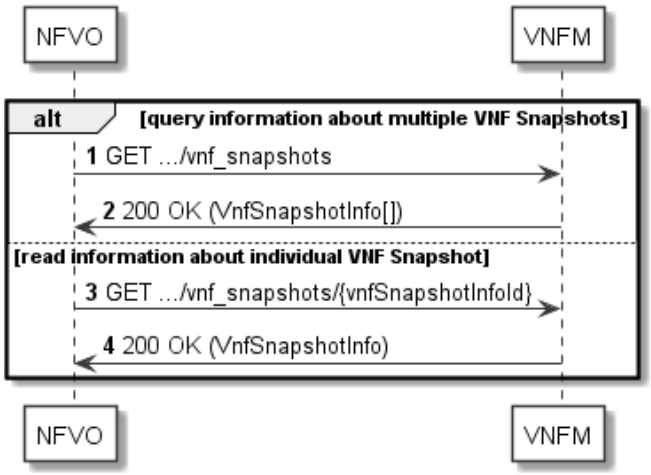

Figure 5.3.15-1: Flow of VNF snapshot query/read

VNF snapshot query/read, as illustrated in figure 5.3.15-1, consists of the following steps:

1) If the NFVO intends to query all VNF snapshots, it sends a GET request to the "VNF snapshots" resource.

2) The VNFM returns a "200 OK" response to the NFVO and includes zero or more data structures of type "VnfSnapshotInfo" in the payload body.

3) If the NFVO intends to read information about a particular VNF snapshot, it sends a GET request to the "Individual VNF snapshot" resource, addressed by the appropriate VNF snapshot information identifier in its resource URI.

4) The VNFM returns a "200 OK" response to the NFVO and includes one data structure of type "VnfSnapshotInfo" in the payload body.

Error handling: In case of failure, appropriate error information is provided in the response.

5.3.16 Flow of modify VNF snapshot resource information

This clause describes the procedure for modifying information of a VNF snapshot resource.

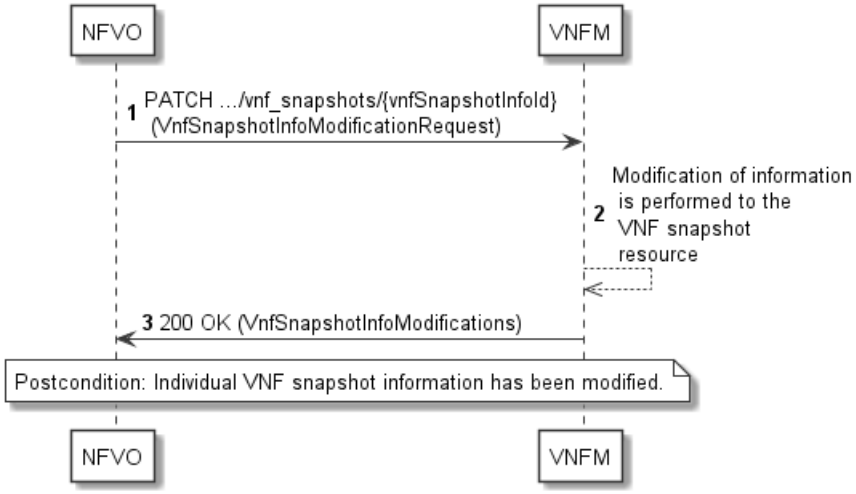

Figure 5.3.16-1: Flow of modify VNF snapshot resource information

The procedure consists of the following steps as illustrated in figure 5.3.16-1:

1) The NFVO sends a PATCH request to the "Individual VNF snapshot" resource and includes in the payload body a data structure of type "VnfSnapshotInfoModificationRequest".

2) The VNFM performs the modification of information to the VNF snapshot resource.

3) The VNFM returns a "200 OK" response and includes in the payload body a data structure of type "VnfSnapshotInfoModifications" to indicate the completion of the operation and the performed changes.

Postcondition: Upon successful completion, the information of the VNF snapshot resource is updated.

Error handling: In case of failure, appropriate error information is provided in the response.

5.3.17 Flow of fetching the VNF state snapshot content

This clause describes a sequence for fetching the VNF state snapshot content.

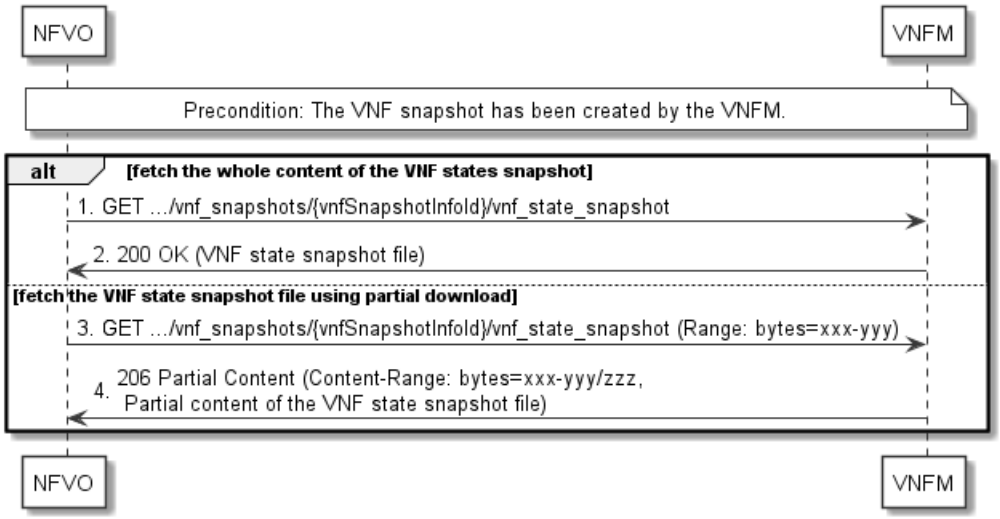

Figure 5.3.17-1: Flow of fetching the VNF state snapshot content

Precondition: The VNF snapshot has been created by the VNFM.

Fetching the VNF state snapshot content, as illustrated in figure 5.3.17-1, consists of the following steps:

1) If fetching the whole content of the VNF state snapshot, the NFVO sends a GET request to the "VNF state snapshot" resource.

2) The VNFM returns a "200 OK" response and includes a copy of the VNF state snapshot content in the payload body.

3) If fetching the content of the VNF state snapshot using partial download, the NFVO sends a GET request to the "VNF state snapshot" resource and includes a "Range" HTTP header indicating the partition of the file that needs to be transferred.

4) The VNFM returns a "206 Partial Content" response with a payload body containing the portion of the file with the VNF state snapshot content, and a "Content-Range" HTTP header indicating the byte range enclosed in the payload and the complete length of the file.

Error handling: In case of failure, appropriate error information is provided in the response.

5.3.18 Flow of the deletion of a VNF snapshot resource

This clause describes the procedure for the deletion of a VNF snapshot resource.

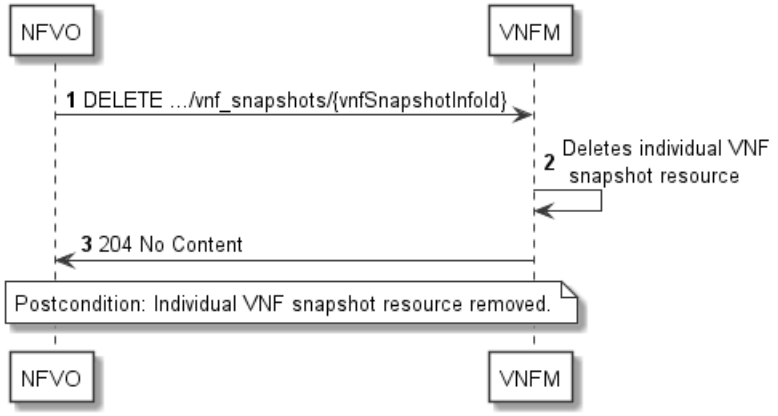

Figure 5.3.18-1: Flow of the deletion of a VNF snapshot resource

The procedure consists of the following steps as illustrated in figure 5.3.18-1:

1) NFVO sends a DELETE request to the "Individual VNF snapshot" resource.

2) The VNFM deletes the VNF snapshot resource and the associated VNF snapshot.

3) The VNFM returns a "204 No Content" response with an empty payload body.

Postcondition: The resource representing the VNF snapshot has been removed from the list of VNF snapshot resources, and the VNF snapshot has been deleted.

Error handling: In case of failure, appropriate error information is provided in the response.

5.4 Resources

5.4.1 Introduction

5.4.1.1 Overview

This clause defines all the resources and methods provided by the VNF lifecycle management interface.

5.4.1.2 Task resources that trigger VNF LCM operations

A number of resources are defined as task resources to trigger VNF LCM operations that are potentially long-running (e.g. Instantiate VNF, Scale VNF). To represent each occurrence of such a VNF LCM operation, an "Individual VNF LCM operation occurrence" resource is created as defined in clause 5.4.13.

When successfully executing the POST method on a task resource that triggers a VNF LCM operation, asynchronous processing of the request is started, which shall include the following:

1) Before returning the "202 Accepted" response to the POST method, a new "Individual VNF LCM operation occurrence" resource as defined in clause 5.4.13 shall be created, which represents the underlying VNF LCM operation occurrence that is executed by the VNFM. The VNFM shall set the "operationState" in the representation of the "Individual VNF LCM operation occurrence" resource to "STARTING".

2) Notifications of type "VnfLcmOperationOccurrenceNotification" shall be triggered as part of executing the underlying VNF LCM operation occurrence as defined in clauses 5.5.2.17 and 5.6.2.

3) If the VNFM has successfully completed the underlying VNF LCM operation occurrence:

   a) It shall update the representation of the "Individual VNF instance" resource which has been changed by the LCM operation to reflect the result of the operation. For individual operations, specific additional conditions can be specified in the following clauses, if applicable.

   b) It shall set the "operationState" attribute in the representation of the aforementioned "Individual VNF LCM operation occurrence" resource to the value "COMPLETED" and shall reflect the changes performed during the LCM operation in the representation of that resource.

   c) To indicate success, the VNFM shall send a notification of type "VnfLcmOperationOccurrenceNotification" with the "operationState" attribute set to "COMPLETED" as defined in clause 5.6.2.

4) If executing the underlying VNF LCM operation occurrence by the VNFM has failed in the "STARTING" phase, the VNFM shall send a notification of type "VnfLcmOperationOccurrenceNotification" with the "operationState" attribute set to "ROLLED_BACK" as defined in clause 5.6.2. It shall also set the "operationState" attribute in the representation of the aforementioned "Individual VNF LCM operation occurrence" resource to the value "ROLLED_BACK".

5) If executing the underlying VNF LCM operation occurrence by the VNFM has failed with no option to recover, the VNFM shall send a notification of type "VnfLcmOperationOccurrenceNotification" with the "operationState" attribute set to "FAILED" as defined in clause 5.6.2. It shall also set the "operationState" attribute in the representation of the aforementioned "Individual VNF LCM operation occurrence" resource to the value "FAILED", and shall reflect, at its best knowledge, the changes performed during the LCM operation in the representation of that resource.

6) If executing the underlying VNF LCM operation occurrence by the VNFM has failed temporarily, the VNFM shall send a notification of type "VnfLcmOperationOccurrenceNotification" with the "operationState" attribute set to "FAILED_TEMP" as defined in clause 5.6.2. It shall also set the "operationState" attribute in the representation of the aforementioned "Individual VNF LCM operation occurrence" resource to the value "FAILED_TEMP", and shall reflect, at its best knowledge, the changes performed so far during the LCM operation in the representation of that resource.

The preconditions and postconditions for a successful execution of each of the VNF lifecycle management operations triggered by the corresponding task resources shall be as defined in table 5.4.1.2-1.

Table 5.4.1.2-1: Preconditions, postconditions and parameterization of the flow for different VNF lifecycle management operations

Operation	Precondition	Task	RequestStructure	Postcondition
Instantiate VNF	VNF instance created and in NOT_INSTANTIATED state	instantiate	InstantiateVnfRequest	VNF instance in INSTANTIATED state
Scale VNF	VNF instance in INSTANTIATED state	scale	ScaleVnfRequest	VNF instance still in INSTANTIATED state and VNF has been scaled
Scale VNF to Level	VNF instance in INSTANTIATED state	scale_to_level	ScaleVnfToLevel Request	VNF instance still in INSTANTIATED state and VNF has been scaled
Change VNF flavour	VNF instance in INSTANTIATED state	change_flavour	ChangeVnfFlavour Request	VNF instance still in INSTANTIATED state and VNF deployment flavour changed
Operate VNF	VNF instance in INSTANTIATED state	operate	OperateVnfRequest	VNF instance still in INSTANTIATED state and VNF operational state changed
Heal VNF	VNF instance in INSTANTIATED state	heal	HealVnfRequest	VNF instance still in INSTANTIATED state
Change external VNF connectivity	VNF instance in INSTANTIATED state	change_ext_conn	ChangeExtVnfConnectivityRequest	VNF instance still in INSTANTIATED state and external connectivity of the VNF is changed
Change current VNF package	VNF instance in INSTANTIATED state	change_vnfpkg	ChangeCurrentVnfPkgRequest	VNF instance still in INSTANTIATED state and is now based on another VNF package
Create VNF snapshot	VNF instance in INSTANTIATED state and "Individual VNF snapshot" resource is available	create_snapshot	CreateVnfSnapshotRequest	VNF instance still in INSTANTIATED state and a VNF snapshot has been created
Revert to VNF snapshot	VNF instance in INSTANTIATED state	revert_to_snapshot	RevertToVnfSnapshotRequest	VNF instance still in INSTANTIATED state and VNF has been reverted to the snapshot status.
Terminate VNF	VNF instance in INSTANTIATED state	terminate	TerminateVnfRequest	VNF instance in NOT_INSTANTIATED state

5.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [8] are part of the VNF lifecycle management interface.

5.4.2 Resource: VNF instances

5.4.2.1 Description

This resource represents VNF instances. The API consumer can use this resource to create "Individual VNF instance" resources, and to query VNF instances.

5.4.2.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances

This resource shall support the resource URI variables defined in table 5.4.2.2-1.

Table 5.4.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.

5.4.2.3 Resource methods

5.4.2.3.1 POST

The POST method creates a new VNF instance resource based on a VNF package that is onboarded and in "ENABLED" state.

This method shall follow the provisions specified in tables 5.4.2.3.1-1 and 5.4.2.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual VNF instance" resource as defined in clause 5.4.3 shall have been created, and the value of the "instantiationState" attribute in the representation of that resource shall be "NOT_INSTANTIATED". A notification of type VnfIdentifierCreationNotification shall be triggered as part of successfully executing this method as defined in clause 5.5.2.18.

When initiating the creation of a VNF instance resource, the passed metadata values can differ from the default values for metadata, if any, declared in the VNFD.

The VNFM shall apply the "metadata" attributes in the "CreateVnfRequest" data structure in the payload body to the "metadata" attribute in the "VnfInstance" data structure on top of the default values that were obtained from the VNFD according to the rules of JSON Merge Patch (see IETF RFC 7396 [5]). For all metadata keys defined in the VNFD, the VNFM shall ensure that the content of the resulting "metadata" attributes is valid against the data type definitions in the VNFD. The absence of a metadata item that is marked "required" in the VNFD shall not be treated as an error. In case a "metadata" child attribute is not defined in the VNFD, the VNFM shall consider it valid in case it is a valid JSON structure.

In case of an error, the operation shall be rejected with a "422 Unprocessable Entity" error.

Table 5.4.2.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.2.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	CreateVnfRequest	1	The VNF creation parameters, as defined in clause 5.5.2.3	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfInstance	1	201 Created	Shall be returned when a new "Individual VNF instance" resource and the associated VNF instance identifier has been created successfully. The response body shall contain a representation of the created VNF instance, as defined in clause 5.5.2.2. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created VNF instance.
	ProblemDetails	1	422 Unprocessable Entity	Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this resource, the response code 422 shall also be returned if the VNF package referenced by the "vnfdId" attribute in the "CreateVnfRequest" structure is not in the "ENABLED" state or does not exist. In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.2.3.2 GET

The GET method queries information about multiple VNF instances.

This method shall follow the provisions specified in tables 5.4.2.3.2-1 and 5.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.2.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [8]. The VNFM shall support receiving this parameter as part of the URI query string. The NFVO may supply this parameter. All attribute names that appear in the VnfInstance and in data types referenced from it shall be supported by the VNFM in the filter expression.
all_fields	0..1	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The VNFM shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The VNFM should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The VNFM should support this parameter.

Name	Cardinality	Description
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The VNFM shall support this parameter. The following attributes shall be excluded from the VnfInstance structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: - vnfConfigurableProperties - vimConnectionInfo - instantiatedVnfInfo - metadata - extensions
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource.

Table 5.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfInstance	0..N	200 OK	Shall be returned when information about zero or more VNF instances has been queried successfully. The response body shall contain in an array the representations of zero or more VNF instances, as defined in clause 5.5.2.2. If the "filter" URI parameter or one of the "all_fields", "fields" (if supported), "exclude_fields" (if supported) or "exclude_default" URI parameters was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clauses 5.2.2 and 5.3.2 of ETSI GS NFV-SOL 013 [8], respectively. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute selector. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.3 Resource: Individual VNF instance

5.4.3.1 Description

This resource represents an individual VNF instance. The API consumer can use this resource to modify and delete the underlying VNF instance, and to read information about the VNF instance.

5.4.3.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}

The base resource URI variables for this resource are defined in table 5.4.3.2-1.

Table 5.4.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual VNF instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.3.3 Resource methods

5.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.3.3.2 GET

The GET method retrieves information about a VNF instance by reading an "Individual VNF instance" resource.

This method shall follow the provisions specified in tables 5.4.3.3.2-1 and 5.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.3.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response codes**	**Description**
	VnfInstance	1	200 OK	Shall be returned when information about an individual VNF instance has been read successfully. The response body shall contain a representation of the VNF instance, as defined in clause 5.5.2.2.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.3.3.4 PATCH

This method modifies an "Individual VNF instance" resource.

Changes to the VNF configurable properties are applied to the configuration in the VNF instance and are reflected in the representation of this resource. Other changes are applied to the VNF instance information managed by the VNFM and are reflected in the representation of this resource.

This method shall follow the provisions specified in tables 5.4.3.3.4-1 and 5.4.3.3.4-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 5.4.1.2.

The VNFM shall apply the "metadata", "extensions" and "vnfConfigurableProperties" attributes in the "VnfInfoModificationRequest" data structure in the payload body to the existing "extensions" and "vnfConfigurableProperties" attributes from the "VnfInstance" data structure according to the rules of JSON Merge Patch (see IETF RFC 7396 [5]).

The VNFM shall ensure that the content of the child attributes of the resulting "metadata", "extensions" and "vnfConfigurableProperties" attributes is valid against the data type definitions of these child attributes in the VNFD.

In case a "metadata" child attribute is not defined in the VNFD, the VNFM shall consider it valid in case it is a valid JSON structure.

NOTE 1: "Extensions" and "vnfConfigurableProperties" child attributes are always declared in the VNFD.

If the VNF instance is in the "INSTANTIATED" state, the validation shall also include ensuring the presence of all "extensions" and "vnfConfigurableProperties" child attributes that are marked as "required" in the VNFD.

NOTE 2: This allows to build the set of "extensions" and "vnfConfigurableProperties" incrementally prior VNF instantiation but ensures their completeness for an instantiated VNF instance.

The absence of a metadata item that is marked "required" in the VNFD shall not be treated as an error.

In case of an error during validation, the operation shall be automatically rolled back, and appropriate error information shall be provided in the "VnfLcmOperationOccurrenceNotification" message and the "VnfLcmOpOcc" data structure. The processing of changes to the "metadata"/"extensions"/"vnfConfigurableProperties" attributes shall be performed in the "PROCESSING" phase of the LCM operation. The change shall be atomic, i.e. the result of intermediate stages shall not be visible in the API. In case of successful completion of the processing and validation, the attributes shall be provided in the "VnfInstance" data structure and the operation shall proceed towards successful completion.

Table 5.4.3.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.3.3.4-2: Details of the PATCH request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	VnfInfoModificationRequest	1		Parameters for the VNF modification, as defined in clause 5.5.2.12. The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [5].
Response body	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. On success, the HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the operation. The response body shall be empty.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the "Individual VNF instance" resource. Typically, this is due to the fact that another LCM operation is ongoing. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	412 Precondition failed	Shall be returned upon the following error: A precondition given in an HTTP request header is not fulfilled. Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity. The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.3.3.5 DELETE

This method deletes an "Individual VNF instance" resource.

This method shall follow the provisions specified in tables 5.4.3.3.5-1 and 5.4.3.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual VNF instance" resource shall not exist any longer. A notification of type "VnfIdentifierDeletionNotification" shall be triggered as part of successfully executing this method as defined in clause 5.5.2.19.

Table 5.4.3.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.3.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned when the "Individual VNF instance" resource and the associated VNF identifier were deleted successfully. The response body shall be empty.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that the "Individual VNF instance" resource is in INSTANTIATED state. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.4 Resource: Instantiate VNF task

5.4.4.1 Description

This task resource represents the "Instantiate VNF" operation. The API consumer can use this resource to instantiate a VNF instance.

5.4.4.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/instantiate

This resource shall support the resource URI variables defined in table 5.4.4.2-1.

Table 5.4.4.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	The identifier of the VNF instance to be instantiated. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual VNF instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.4.3 Resource methods

5.4.4.3.1 POST

The POST method instantiates a VNF instance.

This method shall follow the provisions specified in tables 5.4.4.3.1-1 and 5.4.4.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 5.4.1.2.

In addition, once the VNFM has successfully completed the underlying VNF LCM operation occurrence, it shall set the "instantiationState" attribute to the value "INSTANTIATED" and the "vnfState" attribute to the value "STARTED" in the representation of the "Individual VNF instance" resource.

When instantiating a VNF instance, the values of the extensions and/or VNF configurable properties passed in the instantiation request can differ from the values in the "VnfInstance" data structure that were initialized from default values, if any, declared in the VNFD.

The VNFM shall apply the "extensions" and "vnfConfigurableProperties" attributes in the "InstantiateVnfRequest" data structure in the payload body to the existing "extensions" and "vnfConfigurableProperties" attributes from the "VnfInstance" data structure according to the rules of JSON Merge Patch (see IETF RFC 7396 [5]). The VNFM shall ensure that the content of the resulting "extensions" and "vnfConfigurableProperties" attributes is valid against the VNFD (including the presence of all child attributes that are marked as "required" in the VNFD). In case of an error during validation, the operation shall be automatically rolled back, and appropriate error information shall be provided in the "VnfLcmOperationOccurrenceNotification" message and the "VnfLcmOpOcc" data structure. The processing of changes to the "extensions"/"vnfConfigurableProperties" attributes shall be performed in the "STARTING" phase of the LCM operation. The change shall be atomic, i.e. the result of intermediate stages shall not be visible in the API. In case of successful completion of the processing and validation, the attributes shall be provided in the "VnfInstance" data structure and the operation shall proceed to obtain the grant.

Table 5.4.4.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.4.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	InstantiateVnfRequest	1		Parameters for the VNF instantiation, as defined in clause 5.5.2.4.
Response body	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the operation.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that the "Individual VNF instance" resource is in INSTANTIATED state, or that a required (see note) child attribute of the "extensions" attribute has not been set. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.
NOTE: Required attributes are marked as "required" in the VNFD.				

5.4.4.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.5 Resource: Scale VNF task

5.4.5.1 Description

This task resource represents the "Scale VNF" operation. The API consumer can use this resource to request scaling a VNF instance.

It depends on the VNF capabilities, and is declared in the VNFD, whether this operation is supported for a particular VNF.

See clause B.2 for an explanation of VNF scaling.

5.4.5.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/scale

This resource shall support the resource URI variables defined in table 5.4.5.2-1.

Table 5.4.5.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance to be scaled. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual VNF instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.5.3 Resource methods

5.4.5.3.1 POST

The POST method requests to scale a VNF instance resource incrementally.

This method shall follow the provisions specified in tables 5.4.5.3.1-1 and 5.4.5.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 5.4.1.2.

In addition, once the VNFM has successfully completed the underlying VNF LCM operation occurrence, it shall reflect the result of scaling the VNF instance by updating the "scaleStatus" attribute in the representation of the "Individual VNF instance" resource.

Table 5.4.5.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.5.3.1-2: Details of the POST request/response on this resource

Request body	**Data type**	**Cardinality**	**Description**	
	ScaleVnfRequest	1	Parameters for the scale VNF operation, as defined in clause 5.5.2.5.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the operation.
	ProblemDetails	0..1	404 Not Found	Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this task resource, the response code 404 shall also be returned if the task is not supported for the VNF instance represented by the parent resource, which means that the task resource consequently does not exist. In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that the "Individual VNF instance" resource is in NOT_INSTANTIATED state, that another lifecycle management operation is ongoing, or that a required (see note) child attribute of the "extensions" attribute has not been set. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.
NOTE: Required attributes are marked as "required" in the VNFD.				

5.4.5.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.6 Resource: Scale VNF to Level task

5.4.6.1 Description

This task resource represents the "Scale VNF to Level" operation. The API consumer can use this resource to request scaling of a VNF instance to a target level.

It depends on the VNF capabilities, and is declared in the VNFD, whether this operation is supported for a particular VNF.

See clause B.2 for an explanation of VNF scaling.

5.4.6.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/scale_to_level

This resource shall support the resource URI variables defined in table 5.4.6.2-1.

Table 5.4.6.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance to be scaled to a target level. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual VNF instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.6.3 Resource methods

5.4.6.3.1 POST

The POST method requests to scale a VNF instance resource to a target level.

This method shall follow the provisions specified in tables 5.4.6.3.1-1 and 5.4.6.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 5.4.1.2.

In addition, once the VNFM has successfully completed the underlying VNF LCM operation occurrence, it shall reflect the result of scaling the VNF instance by updating the "scaleStatus" attribute in the representation of the "Individual VNF instance" resource.

Table 5.4.6.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.6.3.1-2: Details of the POST request/response on this resource

Request body	**Data type**	**Cardinality**	**Description**	
	ScaleVnfToLevelRequest	1	Parameters for the scale VNF to Level operation, as defined in clause 5.5.2.6.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the operation.
	ProblemDetails	0..1	404 Not Found	Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this task resource, the response code 404 shall also be returned if the task is not supported for the VNF instance represented by the parent resource, which means that the task resource consequently does not exist. In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that the VNF instance resource is in NOT_INSTANTIATED state, that another lifecycle management operation is ongoing, or that a required (see note) child attribute of the "extensions" attribute has not been set. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.
NOTE: Required attributes are marked as "required" in the VNFD.				

5.4.6.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.6.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.6.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.7 Resource: Change VNF Flavour task

5.4.7.1 Description

This task resource represents the "Change VNF Flavour" operation. The API consumer can use this resource to change the deployment flavour for a VNF instance.

It depends on the VNF capabilities, and is declared in the VNFD, whether this operation is supported for a particular VNF. This operation may be service-disruptive.

5.4.7.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/change_flavour

This resource shall support the resource URI variables defined in table 5.4.7.2-1.

Table 5.4.7.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	The identifier of the VNF instance of which the deployment flavour is requested to be changed. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual VNF instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.7.3 Resource methods

5.4.7.3.1 POST

The POST method changes the deployment flavour of a VNF instance.

This method shall follow the provisions specified in tables 5.4.7.3.1-1 and 5.4.7.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 5.4.1.2.

In addition, once the VNFM has successfully completed the underlying VNF LCM operation occurrence, it shall set the "flavourId" attribute in the representation of the "Individual VNF instance" resource to the value of the "newFlavourId" attribute passed in the "ChangeVnfFlavourRequest" data in the POST request.

When initiating a change of the current VNF flavour, the values of the extensions and/or VNF configurable properties, can differ between the new flavour and the old flavour of the VNF instance.

The VNFM shall apply the "extensions" and "vnfConfigurableProperties" attributes in the "ChangeVnfFlavourRequest" data structure in the payload body to the existing "extensions" and "vnfConfigurableProperties" attributes from the "VnfInstance" data structure according to the rules of JSON Merge Patch (see IETF RFC 7396 [5]). The VNFM shall ensure that the content of the resulting "extensions" and "vnfConfigurableProperties" attributes is valid against the VNFD (which includes ensuring the presence of all child attributes that are marked as "required" in the VNFD). In case of an error, the operation shall be automatically rolled back, and appropriate error information shall be provided in the "VnfLcmOperationOccurrenceNotification" message and the "VnfLcmOpOcc" data structure. The processing of changes to the "extensions"/"vnfConfigurableProperties" attributes shall be performed in the "STARTING" phase of the LCM operation. The change shall be atomic, i.e. the result of intermediate stages shall not be visible in the API. In case of successful completion of the processing and validation, the attributes shall be provided in the "VnfInstance" data structure and the operation shall proceed to obtain the grant.

Table 5.4.7.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.7.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	ChangeVnfFlavourRequest	1	Parameters for the Change VNF Flavour operation, as defined in clause 5.5.2.7.	
Response body	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the operation.
	ProblemDetails	0..1	404 Not Found	Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this task resource, the response code 404 shall also be returned if the task is not supported for the VNF instance represented by the parent resource, which means that the task resource consequently does not exist. In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that the "Individual VNF instance" resource is in NOT_INSTANTIATED state, that another lifecycle management operation is ongoing, or that a required (see note) child attribute of the "extensions" attribute has not been set. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.
NOTE: Required attributes are marked as "required" in the VNFD.				

5.4.7.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.7.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.7.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.7.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.8 Resource: Terminate VNF task

5.4.8.1 Description

This task resource represents the "Terminate VNF" operation. The API consumer can use this resource to terminate a VNF instance.

5.4.8.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/terminate

This resource shall support the resource URI variables defined in table 5.4.8.2-1.

Table 5.4.8.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	The identifier of the VNF instance to be terminated. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual VNF instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.8.3 Resource methods

5.4.8.3.1 POST

The POST method triggers the VNFM to terminate a VNF instance and to request to the VIM the release of its used virtualised resources.

This method shall follow the provisions specified in tables 5.4.8.3.1-1 and 5.4.8.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 5.4.1.2.

In addition, once the VNFM has successfully completed the underlying VNF LCM operation occurrence, it shall set the "instantiationState" attribute in the representation of the "Individual VNF instance" resource to the value "NOT_INSTANTIATED".

Table 5.4.8.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.8.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	TerminateVnfRequest	1		Parameters for the VNF termination, as defined in clause 5.5.2.8.
Response body	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the operation.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that the "Individual VNF instance" resource is in NOT_INSTANTIATED state, that another lifecycle management operation is ongoing, or that a required (see note) child attribute of the "extensions" attribute has not been set. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.
NOTE: Required attributes are marked as "required" in the VNFD.				

5.4.8.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.8.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.8.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.8.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.9 Resource: Heal VNF task

5.4.9.1 Description

This task resource represents the "Heal VNF" operation. The API consumer can use this resource to request healing a VNF instance.

It depends on the VNF capabilities, and is declared in the VNFD, whether this operation is supported for a particular VNF.

5.4.9.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/heal

This resource shall support the resource URI variables defined in table 5.4.9.2-1.

Table 5.4.9.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance to be healed. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual VNF instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.9.3 Resource methods

5.4.9.3.1 POST

The POST method requests to heal a VNF instance.

This method shall follow the provisions specified in tables 5.4.9.3.1-1 and 5.4.9.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 5.4.1.2.

Table 5.4.9.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.9.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	HealVnfRequest	1		Parameters for the Heal VNF operation, as defined in clause 5.5.2.9.
Response body	n/a		202 Accepted	Shall be returned when the request has been accepted for processing, but the processing has not been completed. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the operation.
	ProblemDetails	0..1	404 Not Found	Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this task resource, the response code 404 shall also be returned if the task is not supported for the VNF instance represented by the parent resource, which means that the task resource consequently does not exist. In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that the "Individual VNF instance" resource is in NOT_INSTANTIATED state, that another lifecycle management operation is ongoing, or that a required (see note) child attribute of the "extensions" attribute has not been set. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.
NOTE: Required attributes are marked as "required" in the VNFD.				

5.4.9.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.9.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.9.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.9.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.10 Resource: Operate VNF task

5.4.10.1 Description

This task resource represents the "Operate VNF" operation. The API consumer can use this resource to operate a VNF instance.

It depends on the VNF capabilities, and is declared in the VNFD, whether this operation is supported for a particular VNF.

The "Operate VNF" operation enables requesting to change the operational state of a VNF instance, including starting and stopping the VNF instance.

NOTE 1: These operations are complementary to instantiating and terminating a VNF.

NOTE 2: In the present document, only starting and stopping the VNF instances is supported. Extension of this operation to support other VNF state changes is left for future specification.

A VNF instance can be in the following states:

STARTED: the VNF instance is up and running.

STOPPED: the VNF instance has been shut down, i.e. all its VNFC instances have been stopped.

In the state STOPPED, the virtualisation containers, where the VNFC instances of the VNF run, are shut down but not deleted. In addition, if the workflow requires a graceful stop, as part of this process the VNFM (API producer) will interact with VNF/EM to gracefully stop the VNF application. Once a VNF is instantiated, i.e. all instantiation steps have been completed, the VNF instance is in the state STARTED.

5.4.10.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/operate

This resource shall support the resource URI variables defined in table 5.4.10.2-1.

Table 5.4.10.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance to be operated. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual VNF instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.10.3 Resource methods

5.4.10.3.1 POST

The POST method changes the operational state of a VNF instance.

This method shall follow the provisions specified in tables 5.4.10.3.1-1 and 5.4.10.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 5.4.1.2.

In addition, once the VNFM has successfully completed the underlying VNF LCM operation occurrence, it shall set the "vnfState" attribute in the representation of the "Individual VNF instance" resource to the value of the "changeStateTo" attribute passed in the "OperateVnfRequest" data in the POST request.

Table 5.4.10.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.10.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	OperateVnfRequest	1		Parameters for the Operate VNF operation, as defined in clause 5.5.2.10.
Response body	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the operation.

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	0..1	404 Not Found	Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this task resource, the response code 404 shall also be returned if the task is not supported for the VNF instance represented by the parent resource, which means that the task resource consequently does not exist. In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that the VNF instance resource is in NOT_INSTANTIATED state, that another lifecycle management operation is ongoing, or that a required (see note) child attribute of the "extensions" attribute has not been set. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.
NOTE: Required attributes are marked as "required" in the VNFD.				

5.4.10.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.10.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.10.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.10.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.11 Resource: Change external VNF connectivity task

5.4.11.1 Description

This task resource represents the "Change external VNF connectivity" operation. The API consumer can use this resource to change the external connectivity of a VNF instance. The types of changes that this operation supports are:

- Disconnect external CPs that are connected to a particular external VL and connect them to a different external VL.
- Disconnect and delete external CPs that are connected to a particular external VL and that represent subports in a trunk, i.e. CP instances that are created from external CPDs that have trunk mode configured according to clause 7.1.6.3 in ETSI GS NFV-IFA 011 [10]. If the parent port is exposed as an external CP, the VNFM shall ensure that the parent port is not deleted. If the parent port is exposed as an external CP and there are other subports connected, the VNFM shall ensure that the parent port is not disconnected, unless it is reconnected to a different external VL in the same operation.
- Change the connectivity parameters of existing external CPs, including changing addresses.
- Create new CPs that represent subports in a trunk, i.e. CP instances that are created from external CPDs that have trunk mode configured according to clause 7.1.6.3 in ETSI GS NFV-IFA 011 [10] and connect them to a particular external VL. Creation of the parent port with this operation is not supported.

NOTE: Depending on the capabilities of the underlying VIM resources, certain changes (e.g. modifying the IP address assignment) might not be supported without deleting the resource and creating another one with the modified configuration.

VNFs shall support this operation. This operation may be service-disruptive.

5.4.11.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/change_ext_conn

This resource shall support the resource URI variables defined in table 5.4.11.2-1.

Table 5.4.11.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance of which the external connectivity is requested to be changed. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual VNF instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.11.3 Resource methods

5.4.11.3.1 POST

The POST method changes the external connectivity of a VNF instance.

This method shall follow the provisions specified in tables 5.4.11.3.1-1 and 5.4.11.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 5.4.1.2.

Table 5.4.11.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.11.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Description	
Request body	ChangeExtVnfConnectivity Request	1	Parameters for the Change external VNF connectivity operation, as defined in clause 5.5.2.11.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the instantiation operation.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that another lifecycle management operation is ongoing, or that a required (see note) child attribute of the "extensions" attribute has not been set. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.
NOTE: Required attributes are marked as "required" in the VNFD.				

5.4.11.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.11.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.11.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.11.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.11a Resource: Change current VNF package task

5.4.11a.1 Description

This operation enables the NFVO to request the VNFM to change the current VNF Package, i.e. the VNF package on which a VNF instance is based. Clause B.3 of the ETSI GS NFV-IFA 007 [1] illustrates the variants of changes to the current VNF Package and information flow procedures.

This operation encompasses the following scenarios:

- Changes of the VNF virtualised resources, such as requirements, composition and structure between the VNF versions, without changing the VNF software version.
- Changes of both the VNF software version and the VNF virtualised resources. This case includes replacing the VNF software version by means of virtualised resources management, such as terminating the current virtualised resource instances running the current software version and instantiating new virtualised resource instances with the destination VNF software version. The new virtualised resource instances may have the same characteristics as the current virtualised resource instances.
- Changes related to the VNFD, such as correction of bugs in the VNFD, changes in the naming scheme of VNFD components (e.g. name of the VDU, vduId), and adding/removing descriptors of VNF Package changes (VnfPackageChangeInfo).

NOTE: For software updates that are executed by functional entities outside NFV-MANO and that require synchronization of the information held by the NFV-MANO entities with a new VNF package that reflects the same changes, an alternative procedure using the PATCH method on the "Individual VNF instance" resource has been defined, as illustrated in clause B.2 of ETSI GS NFV-IFA 007 [1]. This procedure assumes certain restrictions on the characteristics of the new VNF package, as defined in note 1 in table 5.5.2.2-1.

As part of changing the current VNF Package, the VNFM shall be capable to add temporary virtualised resources used in the modification process, e.g. virtualised resources for a VNFC which will be responsible for handling or supporting the change of the current VNF Package process. The need for temporary virtualised resources shall be indicated as "tempResource" to the NFVO during the VNF LCM operation granting exchange. In addition, the VNFM shall be capable to add and remove virtualised resources as required for the "change of current VNF Package" process. The need for addition and removal of existing virtualised resources shall be indicated as "addResource" and "removeResource" in the VNF LCM operation granting exchange.

The following applies to the existing resources of the VNF instance: In the course of the successful execution of this operation, the VNFM shall replace or update those resources of the VNF instance that are based on descriptors (e.g. VDUs, VLDs, CPDs) that have changed between source and destination VNFD to align them with the updated descriptors, with the only allowed exception that the references to software images need not be updated if the resources are not replaced. Further, the VNFM shall remove resources that relate to descriptors in the source VNFD that have no corresponding descriptor in the destination VNFD. For newly-created resources, the VNFM shall use the descriptors of the destination VNFD.

All VNFs shall support this operation. This operation may be service-disruptive.

It is declared in the VNFD whether a change from a particular "source" VNF package to a particular "destination" VNF package is possible. The evaluation of this information shall take place in the "STARTING" phase of the LCM operation. In case the evaluation shows that such change is not possible, the operation shall be automatically rolled back.

In the representation of the VNF instance (see clause 5.5.2.2), there are a number of structures that relate to a particular VNFD, which is reflected by these structures having an attribute of type "IdentifierInVnfd". During the course of the execution of this operation, or due to its final failure, these structures may either refer to the source VNFD or to the destination VNFD of the operation and are accompanied by a "vnfdId" attribute to signal which VNFD they relate to. If that attribute is present, it signals the VNFD that applies to the data structure. If that attribute is absent and the operation is in the "STARTING" phase, the source VNF package is referenced by default. If that attribute is absent and the operation is in any of the phases after "STARTING", the destination VNF package is referenced by default.

5.4.11a.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/change_vnfpkg

This resource shall support the resource URI variables defined in table 5.4.11a.2-1.

Table 5.4.11a.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance of which the underlying VNF package is requested to be changed. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual VNF instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.11a.3 Resource methods

5.4.11a.3.1 POST

The POST method changes the current VNF package on which the VNF instance is based.

This method shall follow the provisions specified in tables 5.4.11a.3.1-1 and 5.4.11a.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 5.4.1.2.

During a change of the current VNF package, the allowed and required extensions and/or VNF configurable properties and their data types, as well as the metadata data types, can differ between the source and the destination VNFD.

The VNFM shall process the child attributes of extensions and VNF configurable properties during the execution of the "Change current VNF package" as follows:

1) First, "extensions" and "vnfConfigurableProperties" child attributes which are not defined in the source VNFD but are defined in the destination VNFD with initial values shall be created automatically and shall be populated by these values.

2) Second, the "extensions" and "vnfConfigurableProperties" attributes in the "ChangeCurrentVnfPkgRequest" data structure in the payload body shall be applied to the existing "extensions" and "vnfConfigurableProperties" attributes from the "VnfInstance" data structure according to the rules of JSON Merge Patch (see IETF RFC 7396 [5]):

    a) For those "extensions" and "vnfConfigurableProperties" child attributes that were already defined in the source VNFD and of which the data type has changed in the destination VNFD and whose current value is not compatible with the new data type, input information is expected to be provided by the API consumer in a way that is compatible with applying the new information on top of the current value using JSON Merge Patch.

    b) For those new "extensions" and "vnfConfigurableProperties" child attributes that are not defined in the source VNFD but are defined in the destination VNFD without initial values and that are required, all information needed to populate them is expected to be provided by the API consumer.

c) Additional changed values can be provided by the API consumer.

3) To clean up, "extensions" and "vnfConfigurableProperties" child attributes that are no longer supported in the destination VNFD and that have not been deleted by explicit input shall be deleted automatically by the VNFM.

4) The VNFM shall validate the resulting "extensions" and "vnfConfigurableProperties" against the destination VNFD (which includes ensuring the presence of all child attributes that are marked as "required" in the VNFD). In case of an error, the operation shall be automatically rolled back, and appropriate error information shall be provided in the "VnfLcmOperationOccurrenceNotification" message and the "VnfLcmOpOcc" data structure.

In addition, the VNFM shall process the metadata attributes during the execution of the "Change current VNF package" as follows:

NOTE 1: Metadata changes cannot be signalled as part of invoking the "Change current VNF package" operation.

1) "Metadata" child attributes which are not defined in the source VNFD, are defined in the destination VNFD with initial values and do not exist in the VNF instance shall be created automatically and shall be populated by these initial values.

2) "Metadata" child attributes that are defined in the source VNFD but not in the destination VNFD shall be kept in the "VnfInstance" data structure unchanged.

3) "Metadata" child attributes that are not defined in the source VNFD, are defined in the destination VNFD and exist in the "VnfInstance" structure at the time when the "Change current VNF package" operation is in the "STARTING" phase shall be handled as follows:

   3a) If these child attributes in the VnfInstance have a data type compatible with the definition in the destination VNFD, the VNFM shall keep the existing value.

   3b) If these child attributes in the VnfInstance have a data type *in*compatible with the definition in the destination VNFD, the VNFM shall automatically roll back the operation and shall provide appropriate error information in the "VnfLcmOperationOccurrenceNotification" message and the "VnfLcmOpOcc" data structure.

4) "Metadata" child attributes that are defined in both the source VNFD and the destination VNFD and whose data type is changed in the destination VNFD compared to the data type defined in the source VNFD, in a way that validation would fail against the data type definition of that attribute in the destination VNFD, shall be handled as follows: The VNFM shall automatically roll back the operation and shall provide appropriate error information in the "VnfLcmOperationOccurrenceNotification" message and the "VnfLcmOpOcc" data structure.

NOTE 2: To address the cases 3b and 4, the API consumer can delete the affected colliding "metadata" child attributes or update their content by using the PATCH operation on the "individual VNF instance" resource prior to invoking the "Change current VNF package" operation.

The validation that the changes to the "extensions"/"vnfConfigurableProperties"/"metadata" attributes can be processed without issues shall be performed in the "STARTING" phase of the LCM operation. In case of successful completion of the validation, the operation shall proceed to obtain the grant.

Further, in the "VnfExtCpData" structure under the "ExtVirtualLinkData" structure in the "ChangeCurrentVnfPkgRequest", the API consumer need not explicitly delete (by setting them to null) those "cpConfig" entries that relate to CPDs which are present in the source VNFD but not in the destination VNFD. Before the successful completion of the operation, the VNFM shall remove these entries from the list that is exposed in the "currentVnfExtCpData" attribute of the "ExtVirtualLinkInfo".

Further, in the "extManagedVirtualLinks" attribute in the "ChangeCurrentVnfPkgRequest", the API consumer may still provide those entries that relate to virtual link descriptors which are present in the source VNFD but not in the destination VNFD. Before the successful completion of the operation, the VNFM shall remove these entries from the list that is exposed in the "extManagedVirtualLinks" attribute of the "VnfInstance" structure.

Table 5.4.11a.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.11a.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Description	
Request body	ChangeCurrentVnfPkgRequest	1	Parameters for the Change current VNF package operation, as defined in clause 5.5.2.11a.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual VNF LCM operation occurrence" resource corresponding to the instantiation operation.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that another lifecycle management operation is ongoing. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.11a.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.11a.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.11a.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.11a.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.12 Resource: VNF LCM operation occurrences

5.4.12.1 Description

This resource represents VNF lifecycle management operation occurrences. The API consumer can use this resource to query status information about multiple VNF lifecycle management operation occurrences.

5.4.12.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_lcm_op_occs

The base resource URI variables for this resource are defined in table 5.4.12.2-1.

Table 5.4.12.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.

5.4.12.3 Resource methods

5.4.12.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.12.3.2 GET

The API consumer can use this method to query status information about multiple VNF lifecycle management operation occurrences.

This method shall follow the provisions specified in tables 5.4.12.3.2-1 and 5.4.12.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.12.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [8]. The VNFM shall support receiving this parameter as part of the URI query string. The NFVO may supply this parameter. All attribute names that appear in the VnfLcmOpOcc and in data types referenced from it shall be supported by the VNFM in the filter expression.
all_fields	0..1	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The VNFM shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The VNFM should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The VNFM should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The VNFM shall support this parameter. The following attributes shall be excluded from the VnfLcmOpOcc structure in the response body if this parameter is provided, or none of the parameters "all_fields," "fields", "exclude_fields", "exclude_default" are provided: - operationParams - error - resourceChanges - changedInfo - changedExtConnectivity - lcmCoordinations - modificationsTriggeredByVnfPkgChange - warnings
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource.

Table 5.4.12.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfLcmOpOcc	0..N	200 OK	Shall be returned when status information for zero or more VNF lifecycle management operation occurrences has been queried successfully. The response body shall contain in an array the status information about zero or more VNF lifecycle operation occurrences, as defined in clause 5.5.2.13. If the "filter" URI parameter or one of the "all_fields", "fields" (if supported), "exclude_fields" (if supported) or "exclude_default" URI parameters was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clauses 5.2.2 and 5.3.2 of ETSI GS NFV-SOL 013 [8], respectively. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute selector. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.12.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.12.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.12.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.13 Resource: Individual VNF LCM operation occurrence

5.4.13.1 Description

This resource represents a VNF lifecycle management operation occurrence. The API consumer can use this resource to read status information about an individual VNF lifecycle management operation occurrence. Further, the API consumer can use task resources which are children of this resource to request cancellation of an operation in progress, and to request the handling of operation errors via retrying the operation, rolling back the operation, or permanently failing the operation.

The VNFM may remove an "Individual VNF LCM operation occurrence" resource some time after it has reached one of the terminal states (i.e. the "operationState" attribute of its representation is equal to one of the values "COMPLETED", "FAILED" or "ROLLED_BACK"). The minimum time how long the VNFM waits before deleting such a resource is defined by means outside the scope of the present document.

5.4.13.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_lcm_op_occs/{vnfLcmOpOccId}

The base resource URI variables for this resource are defined in table 5.4.13.2-1.

Table 5.4.13.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfLcmOpOccId	Identifier of a VNF lifecycle management operation occurrence. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a PATCH or POST request triggering a VNF LCM operation. It can also be retrieved from the "vnfLcmOpOccId" attribute in the VnfLcmOperationOccurrenceNotification.	

5.4.13.3 Resource methods

5.4.13.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.13.3.2 GET

The API consumer can use this method to retrieve status information about a VNF lifecycle management operation occurrence by reading an "Individual VNF LCM operation occurrence" resource.

This method shall follow the provisions specified in tables 5.4.13.3.2-1 and 5.4.13.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.13.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.13.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfLcmOpOcc	1	200 OK	Shall be returned when information about a VNF LCM operation occurrence has been read successfully. The response body shall contain status information about a VNF lifecycle management operation occurrence (see clause 5.5.2.13).
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.13.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.13.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.13.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

See clause 5.4.13.1 for a definition related to the removal of an "Individual VNF LCM operation occurrence" resource.

5.4.14 Resource: Retry operation task

5.4.14.1 Description

This task resource represents the "Retry operation" operation. The API consumer can use this resource to initiate retrying a VNF lifecycle operation that is in a transient failure state. See also clause 5.6.2.3.

5.4.14.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/ vnf_lcm_op_occs/{vnfLcmOpOccId}/retry

This resource shall support the resource URI variables defined in table 5.4.14.2-1.

Table 5.4.14.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfLcmOpOccId	Identifier of a VNF lifecycle management operation occurrence to be retried. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a PATCH or POST request triggering a VNF LCM operation. It can also be retrieved from the "vnfLcmOpOccId" attribute in the VnfLcmOperationOccurrenceNotification.	

5.4.14.3 Resource methods

5.4.14.3.1 POST

The POST method initiates retrying a VNF lifecycle operation if that operation has experienced a temporary failure, i.e. the related "Individual VNF LCM operation occurrence" resource is in "FAILED_TEMP" state.

This method shall follow the provisions specified in tables 5.4.14.3.1-1 and 5.4.14.3.1-2 for URI query parameters, request and response data structures, and response codes.

In case of success of processing the asynchronous request, the "operationState" attribute in the representation of the parent resource shall be changed to "PROCESSING" and the applicable "start" notification according to clause 5.6.2.2 shall be emitted to indicate that the underlying VNF LCM operation occurrence proceeds.

Table 5.4.14.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.14.3.1-2: Details of the POST request/response on this resource

Request body	**Data type**	**Cardinality**	**Description**	
	n/a		The POST request to this resource has an empty payload body.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response shall have an empty payload body.
	ProblemDetails	0..1	404 Not Found	Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this task resource, the response code 404 shall also be returned if the task is not supported for the VNF LCM operation occurrence represented by the parent resource, which means that the task resource consequently does not exist. In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the VNF LCM operation occurrence. Typically, this is due to the fact that the VNF LCM operation occurrence is not in FAILED_TEMP state, or another error handling action is starting, such as rollback or fail. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.14.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.14.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.14.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.14.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.15 Resource: Rollback operation task

5.4.15.1 Description

This task resource represents the "Rollback operation" operation. The API consumer can use this resource to initiate rolling back a VNF lifecycle operation. See also clause 5.6.2.3.

5.4.15.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_lcm_op_occs/{vnfLcmOpOccId}/rollback

This resource shall support the resource URI variables defined in table 5.4.15.2-1.

Table 5.4.15.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfLcmOpOccId	Identifier of a VNF lifecycle management operation occurrence to be rolled back. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a PATCH or POST request triggering a VNF LCM operation. It can also be retrieved from the "vnfLcmOpOccId" attribute in the VnfLcmOperationOccurrenceNotification.	

5.4.15.3 Resource methods

5.4.15.3.1 POST

The POST method initiates rolling back a VNF lifecycle operation if that operation has experienced a temporary failure, i.e. the related "Individual VNF LCM operation occurrence" resource is in "FAILED_TEMP" state. In case of rolling back an occurrence of the "InstantiateVnf" operation, the VNFM shall request to the VIM the release of the virtualised resources that were allocated for the related VNF instance. The "rollback" task shall be supported by the VNFM for any VNF LCM operation occurrence that represents an "InstantiateVnf" operation in FAILED_TEMP state.

This method shall follow the provisions specified in tables 5.4.15.3.1-1 and 5.4.15.3.1-2 for URI query parameters, request and response data structures, and response codes.

In case of success of processing the asynchronous request, the "operationState" attribute in the representation of the parent resource shall be changed to "ROLLING_BACK" and the applicable "start" notification according to clause 5.6.2.2 shall be emitted to indicate that rollback of the underlying VNF LCM operation occurrence is attempted.

Table 5.4.15.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.15.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a		The POST request to this resource has an empty payload body.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response shall have an empty payload body.
	ProblemDetails	0..1	404 Not Found	Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this task resource, the response code 404 shall also be returned if the task is not supported for the VNF LCM operation occurrence represented by the parent resource, which means that the task resource consequently does not exist. In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the VNF LCM operation occurrence. Typically, this is due to the fact that the VNF LCM operation occurrence is not in FAILED_TEMP state, or another error handling action is starting, such as retry or fail. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.15.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.15.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.15.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.15.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.16 Resource: Fail operation task

5.4.16.1 Description

This task resource represents the "Fail operation" operation. The API consumer can use this resource to mark a VNF lifecycle management operation occurrence as "finally failed", i.e. change the state of the related VNF LCM operation occurrence to "FAILED", if it is not assumed that a subsequent retry or rollback will succeed. Once the operation is marked as "finally failed", it cannot be retried or rolled back anymore. See also clause 5.6.2.3.

5.4.16.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_lcm_op_occs/{vnfLcmOpOccId}/fail

This resource shall support the resource URI variables defined in table 5.4.16.2-1.

Table 5.4.16.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfLcmOpOccId	Identifier of a VNF lifecycle management operation occurrence to be marked as "failed". See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a PATCH or POST request triggering a VNF LCM operation. It can also be retrieved from the "vnfLcmOpOccId" attribute in the VnfLcmOperationOccurrenceNotification.	

5.4.16.3 Resource methods

5.4.16.3.1 POST

The POST method marks a VNF lifecycle management operation occurrence as "finally failed" if that operation occurrence is in "FAILED_TEMP" state.

This method shall follow the provisions specified in tables 5.4.16.3.1-1 and 5.4.16.3.1-2 for URI query parameters, request and response data structures, and response codes.

In case of success, the "operationState" attribute in the representation of the parent resource shall be changed to "FAILED" and the applicable "result" notification according to clause 5.6.2.2 shall be emitted to indicate that the execution of the underlying VNF LCM operation occurrence has finally and unrecoverably failed.

Table 5.4.16.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.16.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a		The POST request to this resource has an empty payload body.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfLcmOpOcc	1	200 OK	Shall be returned when the state of the VNF lifecycle management operation occurrence has been changed successfully The response body shall include a representation of the "Individual VNF lifecycle operation occurrence" resource.
	ProblemDetails	0..1	404 Not Found	Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this task resource, the response code 404 shall also be returned if the task is not supported for the VNF LCM operation occurrence represented by the parent resource, which means that the task resource consequently does not exist. In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the VNF LCM operation occurrence. Typically, this is due to the fact that the VNF LCM operation occurrence is not in FAILED_TEMP state, or another error handling action is starting, such as retry or rollback. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.16.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.16.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.16.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.16.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.17 Resource: Cancel operation task

5.4.17.1 Description

This task resource represents the "Cancel operation" operation. The API consumer can use this resource to cancel an ongoing VNF lifecycle operation. See also clause 5.6.2.3.

5.4.17.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_lcm_op_occs/{vnfLcmOpOccId}/cancel

This resource shall support the resource URI variables defined in table 5.4.17.2-1.

Table 5.4.17.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfLcmOpOccId	Identifier of a VNF lifecycle management operation occurrence to be cancelled. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a PATCH or POST request triggering a VNF LCM operation. It can also be retrieved from the "vnfLcmOpOccId" attribute in the VnfLcmOperationOccurrenceNotification.	

5.4.17.3 Resource methods

5.4.17.3.1 POST

The POST method initiates cancelling an ongoing VNF lifecycle operation while it is being executed or rolled back, i.e. the related "Individual VNF LCM operation occurrence" resource is either in "STARTING" or "PROCESSING" or "ROLLING_BACK" state.

This method shall follow the provisions specified in tables 5.4.17.3.1-1 and 5.4.17.3.1-2 for URI query parameters, request and response data structures, and response codes.

Before returning the "202 Accepted" response, the VNFM shall update the "isCancelPending" and "cancelMode" attributes in the representation of the parent resource according to the provisions in clause 5.5.2.13.

In case of success of processing the asynchronous request:

1) If the request has been processed in "STARTING" state, the "operationState" attribute in the representation of the parent resource shall be changed to "ROLLED_BACK".

2) If the request has been processed in "PROCESSING" or "ROLLING_BACK" state, the "operationState" attribute in the representation of the parent resource shall be changed to "FAILED_TEMP".

In both cases, the VNFM shall update the "isCancelPending" and "cancelMode" attributes in the representation of the parent resource according to the provisions in clause 5.5.2.13 to reflect the new status, and the applicable "result" notification according to clause 5.6.2.2 shall be emitted to indicate that the execution of the underlying VNF LCM operation occurrence has temporarily failed.

Due to race conditions, the processing of the actual operation that is to be cancelled may eventually still succeed, in which case the "operationState" attribute in the representation of the parent resource shall represent the result of that operation, rather than the result of the cancellation.

Table 5.4.17.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.17.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	CancelMode	1	The POST request to this resource shall include a CancelMode structure in the payload body to choose between "graceful" and "forceful" cancellation.	
Response body	n/a		202 Accepted	Shall be returned when the request has been accepted for processing. The response shall have an empty payload body.
	ProblemDetails	0..1	404 Not Found	Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this task resource, the response code 404 shall also be returned if the task is not supported for the VNF LCM operation occurrence represented by the parent resource, which means that the task resource consequently does not exist. In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the VNF LCM operation occurrence. Typically, this is due to the fact that the operation occurrence is not in STARTING, PROCESSING or ROLLING_BACK state. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.17.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.17.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.17.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.17.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.18 Resource: Subscriptions

5.4.18.1 Description

This resource represents subscriptions. The API consumer can use this resource to subscribe to notifications related to VNF lifecycle management, and to query its subscriptions.

5.4.18.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/subscriptions

This resource shall support the resource URI variables defined in table 5.4.18.2-1.

Table 5.4.18.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.

5.4.18.3 Resource methods

5.4.18.3.1 POST

The POST method creates a new subscription.

This method shall follow the provisions specified in tables 5.4.18.3.1-1 and 5.4.18.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual subscription" resource as defined in clause 5.4.19 shall have been created. This method shall not trigger any notification.

Creation of two "Individual subscription" resources with the same callback URI and the same filter can result in performance degradation and will provide duplicates of notifications to the NFVO, and might make sense only in very rare use cases. Consequently, the VNFM may either allow creating an "Individual subscription" resource if another "Individual subscription" resource with the same filter and callback URI already exists (in which case it shall return the "201 Created" response code), or may decide to not create a duplicate "Individual subscription" resource (in which case it shall return a "303 See Other" response code referencing the existing "Individual subscription" resource with the same filter and callback URI).

Table 5.4.18.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.18.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	LccnSubscriptionRequest	1		Details of the subscription to be created, as defined in clause 5.5.2.15.
Response body	LccnSubscription	1	201 Created	Shall be returned when the subscription has been created successfully. The response body shall contain a representation of the created "Individual subscription" resource. The HTTP response shall include a "Location" HTTP header that points to the created "Individual subscription" resource.
	n/a		303 See Other	Shall be returned if a subscription with the same callback URI and the same filter already exists and the policy of the VNFM is to not create redundant subscriptions. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the existing "Individual subscription" resource. The response body shall be empty.
	ProblemDetails	1	422 Unprocessable Entity	Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this resource, the response code 422 shall also be returned if the VNFM has tested the Notification endpoint as described in clause 5.4.20.3.2 and the test has failed. In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.18.3.2 GET

The GET method queries the list of active subscriptions of the functional block that invokes the method. It can be used e.g. for resynchronization after error situations.

This method shall follow the provisions specified in tables 5.4.18.3.2-1 and 5.4.18.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.18.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [8]. The VNFM shall support receiving this parameter as part of the URI query string. The NFVO may supply this parameter. All attribute names that appear in the LccnSubscription and in data types referenced from it shall be supported by the VNFM in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource.

Table 5.4.18.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Response Codes	Description
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	LccnSubscription	0..N	200 OK	Shall be returned when the list of subscriptions has been queried successfully. The response body shall contain in an array the representations of all active subscriptions of the functional block that invokes the method, i.e. zero or more representations of lifecycle change notification subscriptions as defined in clause 5.5.2.16. If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [8]. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.18.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.18.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.18.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.19 Resource: Individual subscription

5.4.19.1 Description

This resource represents an individual subscription. The API consumer can use this resource to read and to terminate a subscription to notifications related to VNF lifecycle management.

5.4.19.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/subscriptions/{subscriptionId}

This resource shall support the resource URI variables defined in table 5.4.19.2-1.

Table 5.4.19.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
subscriptionId	Identifier of this subscription. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual subscription" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.19.3 Resource methods

5.4.19.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.19.3.2 GET

The GET method retrieves information about a subscription by reading an "Individual subscription" resource.

This method shall follow the provisions specified in tables 5.4.19.3.2-1 and 5.4.19.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.19.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.19.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	LccnSubscription	1	200 OK	Shall be returned when information about an individual subscription has been read successfully. The response body shall contain a representation of the "Individual subscription" resource.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.19.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.19.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.19.3.5 DELETE

The DELETE method terminates an individual subscription.

This method shall follow the provisions specified in tables 5.4.19.3.5-1 and 5.4.19.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual subscription" resource shall not exist any longer. This means that no notifications for that subscription shall be sent to the formerly-subscribed API consumer.

NOTE: Due to race conditions, some notifications might still be received by the formerly-subscribed API consumer for a certain time period after the deletion.

Table 5.4.19.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.19.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned when the "Individual subscription" resource has been deleted successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.20 Resource: Notification endpoint

5.4.20.1 Description

This resource represents a notification endpoint. The API producer can use this resource to send notifications related to VNF lifecycle changes to a subscribed API consumer, which has provided the URI of this resource during the subscription process.

5.4.20.2 Resource definition

The resource URI is provided by the API consumer when creating the subscription.

This resource shall support the resource URI variables defined in table 5.4.20.2-1.

Table 5.4.20.2-1: Resource URI variables for this resource

Name	Definition
none supported	

5.4.20.3 Resource methods

5.4.20.3.1 POST

The POST method delivers a notification from the API producer to an API consumer. The API consumer shall have previously created an "Individual subscription" resource with a matching filter.

This method shall follow the provisions specified in tables 5.4.20.3.1-1 and 5.4.20.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.20.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Each notification request body shall include exactly one of the alternatives defined in table 5.4.20.3.1-2.

Table 5.4.20.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	VnfLcmOperationOccurrenceNotification	1		A notification about lifecycle changes triggered by a VNF LCM operation occurrence.
	VnfIdentifierCreationNotification	1		A notification about the creation of a VNF identifier and the related "Individual VNF instance" resource.
	VnfIdentifierDeletionNotification	1		A notification about the deletion of a VNF identifier and the related "Individual VNF instance" resource.
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned when the notification has been delivered successfully.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.20.3.2 GET

The GET method allows the API producer to test the notification endpoint that is provided by the API consumer, e.g. during subscription.

This method shall follow the provisions specified in tables 5.4.20.3.2-1 and 5.4.20.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.20.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.20.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned to indicate that the notification endpoint has been tested successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.20.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.20.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.20.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.21 Resource: Create VNF snapshot task

5.4.21.1 Description

This task resource represents the "Create VNF Snapshot" operation. The API consumer can use this resource to request creating a VNF snapshot from a VNF instance.

It depends on the VNF capabilities, and is declared in the VNFD, whether this operation is supported for a particular VNF.

5.4.21.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/create_snapshot

This resource shall support the resource URI variables defined in table 5.4.21.2-1.

Table 5.4.21.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance from which a VNF snapshot is to be created. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new VNF instance resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.21.3 Resource methods

5.4.21.3.1 POST

The POST method requests taking a snapshot a VNF instance and populating a previously created VNF snapshot resource (refer to clause 5.4.23.3.1) with the snapshot content.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 5.4.1.2.

In addition, once the VNFM has successfully completed the underlying VNF LCM operation occurrence, it shall reflect the result of the VNF snapshot creation by updating the corresponding "Individual VNF snapshot" resource indicated by the "vnfSnapshotInfoId" attribute of the "CreateVnfSnapshotRequest" that is included in the payload body of the request.

This method shall follow the provisions specified in tables 5.4.21.3.1-1 and 5.4.21.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.21.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.21.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Description	
Request body	CreateVnfSnapshotRequest	1	Parameters for the "Create VNF Snapshot" operation, as defined in clause 5.5.2.21.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		202 Accepted	Shall be returned when the request was accepted for processing, but the processing has not been completed. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "VNF LCM operation occurrence" resource corresponding to the operation.

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	0..1	404 Not Found	Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this task resource, the response code 404 shall also be returned if the task is not supported for the VNF instance represented by the parent resource, which means that the task resource consequently does not exist. In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that the VNF instance resource is in NOT_INSTANTIATED state, or that another lifecycle management operation is ongoing. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	1	422 Unprocessable Entity	Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this resource, the response code 422 shall also be returned if the provided identifier of the target "Individual VNF snapshot" resource for the VNF snapshot is invalid. In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.21.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.21.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.21.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.21.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.22 Resource: Revert to VNF snapshot task

5.4.22.1 Description

This task resource represents the "Revert to VNF Snapshot" operation. The API consumer can use this resource to request reverting a VNF instance to a VNF snapshot.

During the revert to VNF snapshot process, the VNFM shall perform and record the changes on the VNF components and related resources, via the corresponding AffectedVnfc, AffectedVirtualLink, and AffectedVirtualStorage as follows:

- A component instance whose identifier is the same in between the "to be reverted" VNF instance and the snapshot information, its change shall be signalled as "MODIFIED".
- A component instance whose snapshot information is present in the VNF snapshot, but such component is not present in the "to be reverted" VNF instance, its change shall be signalled as "ADDED".
- A component instance which is present in the "to be reverted" VNF instance, but whose snapshot information is not present in the VNF snapshot, the component shall be terminated, and its change shall be signalled as "REMOVED".

During the "revert to VNF snapshot" process, for VNF constituents (e.g. VNFC, connection points, etc.) from the VNF snapshot that are added or modified in the "to be reverted" VNF instance, the VNFM shall assign the original identifier value present in the VNF snapshot in the case that the identifier value setting for such a VNF constituent is the responsibility of the VNFM. The identifier of the VNF instance shall not be modified in the reversion process.

It depends on the VNF capabilities, and is declared in the VNFD, whether this operation is supported for a particular VNF. This operation may be service-disruptive.

5.4.22.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_instances/{vnfInstanceId}/revert_to_snapshot

This resource shall support the resource URI variables defined in table 5.4.22.2-1.

Table 5.4.22.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfInstanceId	Identifier of the VNF instance for the VNF snapshot to be reverted to. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new VNF instance resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.22.3 Resource methods

5.4.22.3.1 POST

The POST method requests reverting a VNF instance to a VNF snapshot.

This method shall follow the provisions specified in tables 5.4.22.3.1-1 and 5.4.22.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.22.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.22.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	RevertToVnfSnapshotRequest	1		Parameters for the Revert to VNF snapshot operation, as defined in clause 5.5.2.26.
Response body	n/a		202 Accepted	Shall be returned when the request was accepted for processing, but the processing has not been completed. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "VNF LCM operation occurrence" resource corresponding to the operation.
	ProblemDetails	0..1	404 Not Found	Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this task resource, the response code 404 shall also be returned if the task is not supported for the VNF instance represented by the parent resource, which means that the task resource consequently does not exist. In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that the VNF instance resource is in NOT_INSTANTIATED state, or that another lifecycle management operation is ongoing. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.22.3.2 GET

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.22.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.22.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.22.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.23 Resource: VNF snapshots

5.4.23.1 Description

This resource represents VNF snapshots. The API consumer can use this resource create "Individual VNF snapshot" resources and to query information of the VNF snapshots.

5.4.23.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_snapshots

This resource shall support the resource URI variables defined in table 5.4.23.2-1.

Table 5.4.23.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8]
apiMajorVersion	See clause 5.1a

5.4.23.3 Resource methods

5.4.23.3.1 POST

The POST method creates a new "Individual VNF snapshot" resource.

As a result of successfully executing this method, a new "Individual VNF snapshot" resource as defined in clause 5.4.24 shall have been created.

The creation of an "Individual VNF snapshot" resource can be performed for two reasons:

- To create an "Individual VNF snapshot" resources that can later be populated by a new VNF snapshot taken from a VNF instance (refer to clause 5.4.21.3.1).
- To create an "Individual VNF snapshot" resource that can be populated with information gathered from a VNF snapshot package extraction. In this case, the API consumer indicates the source of the VNF snapshot package in the payload body of the POST request to the present resource.

In the second case, for a successful execution of the operation, the values in the "VnfSnapshotInfo" data structure representing the "Individual VNF snapshot" resource shall be applied as follows:

- If the request (refer to clause 5.5.2.20) includes the "vnfSnapshot" attribute, the VNFM shall apply the "VnfSnapshotInfo" with such provided information.
- If the request (refer to clause 5.5.2.20) does not include the "vnfSnapshot" attribute, the VNFM shall first fetch the VNF snapshot record from the source VNF snapshot package signalled by the "vnfSnapshotPkgId" attribute in the request and then apply the "VnfSnapshotInfo" from the fetched VNF snapshot record.

This method shall follow the provisions specified in tables 5.4.23.3.1-1 and 5.4.23.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.23.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.23.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	CreateVnfSnapshotInfoRequest	1	The VNF snapshot resource creation parameters, as defined in clause 5.5.2.20.	
Response body	VnfSnapshotInfo	1	201 Created	Shall be returned when an "Individual VNF snapshot" resource has been created successfully. The response body shall contain a representation of the new "Individual VNF snapshot" resource, as defined in clause 5.5.2.22. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the "Individual VNF snapshot" resource.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.23.3.2 GET

The GET method queries information about multiple VNF snapshots.

This method shall follow the provisions specified in tables 5.4.23.3.2-1 and 5.4.23.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.23.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [8]. The VNFM shall support receiving this parameter as part of the URI query string. The NFVO may supply this parameter. All attribute names that appear in the VnfSnapshot and in data types referenced from it shall be supported by the VNFM in the filter expression.
all_fields	0..1	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The VNFM shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The VNFM should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The VNFM should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The VNFM shall support this parameter. The following attributes shall be excluded from the VnfSnapshot structure in the response body if this parameter is provided, or none of the parameters "all_fields," "fields", "exclude_fields", "exclude_default" are provided: - vnfInstance - vnfcSnapshots
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource.

Table 5.4.23.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfSnapshotInfo	0..N	200 OK	Shall be returned when information about zero or more VNF snapshots was queried successfully. The response body shall contain in an array the representations of zero or more "Individual VNF snapshot" resources, as defined in clause 5.5.2.22. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute selector. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.23.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.23.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.23.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.24 Resource: Individual VNF snapshot

5.4.24.1 Description

This resource represents an individual VNF snapshot. The API consumer can use this resource to read information about the VNF snapshot, and to delete the VNF snapshot.

5.4.24.2 Resource definition

The resource URI is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_snapshots/{vnfSnapshotInfoId}

The base resource URI variables for this resource are defined in table 5.4.24.2-1.

Table 5.4.24.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1. of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfSnapshotInfoId	Identifier of the "Individual VNF snapshot" resource. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new VNF snapshot resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.24.3 Resource methods

5.4.24.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.24.3.2 GET

The GET method retrieves information about a VNF snapshot by reading an "Individual VNF snapshot" resource.

This method shall follow the provisions specified in tables 5.4.24.3.2-1 and 5.4.24.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.24.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.24.3.2-2: Details of the GET request/response on this resource

Request body	**Data type**	**Cardinality**	**Description**	
	n/a			
Response body	**Data type**	**Cardinality**	**Response codes**	**Description**
	VnfSnapshotInfo	1	200 OK	Shall be returned when information about an individual VNF snapshot was read successfully. The response body shall contain a representation of the "Individual VNF snapshot" resource, as defined in clause 5.5.2.22.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.24.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.24.3.4 PATCH

This method modifies an "Individual VNF snapshot" resource.

Changes are applied to the VNF snapshot information managed by the VNFM and are reflected in the representation of this resource. The VNFM shall reject the modification request if the "vnfSnapshot" attribute in the "VnfSnapshotInfo" structure representing the "Individual VNF snapshot" resource is not empty, or the resource is associated to an ongoing VNF snapshot operation (e.g. a VNF snapshot creation process has started).

This method shall follow the provisions specified in tables 5.4.24.3.4-1 and 5.4.24.3.4-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.24.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.24.3.4-2: Details of the PATCH request/response on this resource

	Data type	Cardinality	Description	
Request body	VnfSnapshotInfoModificationRequest	1	Parameters for the VNF snapshot information modification, as defined in clause 5.5.2.24. The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [5].	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfSnapshotInfoModifications	1	200 OK	Shall be returned when the modification of VNF snapshot information has been accepted and completed.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the "Individual VNF snapshot" resource. Typically, this is due to the fact another modification is ongoing or that the "Individual VNF snapshot" resource information is not empty due to a previously successful modification or currently being modified due to an underlying VNF snapshot operation. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	412 Precondition failed	Shall be returned upon the following error: A precondition given in an HTTP request header is not fulfilled. Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity. The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.24.3.5 DELETE

This method deletes an "Individual VNF snapshot" resource and the associated VNF snapshot information managed by the VNFM, and any resource associated to the VNF snapshot managed by the VIM.

As the result of successfully executing this method, the "Individual VNF snapshot" resource shall not exist any longer.

This method shall follow the provisions specified in tables 5.4.24.3.5-1 and 5.4.24.3.5-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.24.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.24.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned when the VNF snapshot resource and the associated VNF snapshot were deleted successfully. The response body shall be empty.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that the VNF snapshot is in use by some operation such as reverting a VNF instance to a VNF snapshot or creating a VNF snapshot package. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.25 Resource: VNF state snapshot

5.4.25.1 Description

This resource represents the content of VNF-specific state data associated to a VNF snapshot.

As part of the VNF snapshot creation, VNF-specific state data associated to the VNF snapshot can be created by the VNFM. Such data can be used during VNF snapshot reversions, root cause analysis, etc. and might need to be also compiled by the NFVO into a VNF snapshot package.

The API consumer can use this resource to fetch the content of the VNF state snapshot.

5.4.25.2 Resource definition

The resource URIs is:

{apiRoot}/vnflcm/{apiMajorVersion}/vnf_snapshots/{vnfSnapshotInfoId}/vnf_state_snapshot

This resource shall support the resource URI variables defined in table 5.4.25.2-1.

Table 5.4.25.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 5.1a.
vnfSnapshotInfoId	Identifier of the "Individual VNF snapshot" resource. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new VNF snapshot resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.25.3 Resource methods

5.4.25.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.25.3.2 GET

The GET method fetches the content of the VNF state snapshot.

This method shall follow the provisions specified in tables 5.4.25.3.2-1 and 5.4.25.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.25.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.25.3.2-2: Details of the GET request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	n/a			The request may contain a "Range" HTTP header to obtain single range of bytes from a VNF state snapshot file. This can be used to continue an aborted transmission. If the "Range" header is present in the request and the VNFM does not support responding to range requests with a 206 response, it shall return a 200 OK response instead as defined below.
Response body	n/a	1	200 OK	Shall be returned when the whole content of the VNF state snapshot file has been read successfully. The payload body shall contain a copy of the VNF state snapshot file and the "Content-Type" HTTP header shall be set according to the content type of the VNF state snapshot file. If the VNF state snapshot content is encrypted, the header shall be set to the value "application/cms" (IETF RFC 7193 [9]). If the content type cannot be determined, the header shall be set to the value "application/octet-stream".
	n/a	1	206 Partial Content	If the VNFM supports range requests, this response shall be returned when a single consecutive byte range from the content of the VNF state snapshot file has been read successfully according to the request. The response body shall contain the requested part of the VNF state snapshot file. The "Content-Type" HTTP header shall be set according to the content type of the VNF state snapshot file. If the content type cannot be determined, the header shall be set to the value "application/octet-stream". The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [4].

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that the VNF snapshot creation process is not completed. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	416 Range Not Satisfiable	Shall be returned upon the following error: The byte range passed in the "Range" header did not match any available byte range in the VNF state snapshot file (e.g. "access after end of file"). The response body may contain a ProblemDetails structure.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

5.4.25.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.25.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.4.25.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

5.5 Data model

5.5.1 Introduction

This clause defines the request and response data structures of the VNF Lifecycle management interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error and may choose to ignore them.

5.5.2 Resource and notification data types

5.5.2.1 Introduction

This clause defines the data structures to be used in resource representations and notifications.

5.5.2.2 Type: VnfInstance

This type represents a VNF instance. It shall comply with the provisions defined in table 5.5.2.2-1.

NOTE: Clause B.3.2 provides examples illustrating the relationship among the different run-time information elements (CP, VL and link ports) used to represent the connectivity of a VNF.

Table 5.5.2.2-1: Definition of the VnfInstance data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the VNF instance.
vnfInstanceName	String	0..1	Name of the VNF instance. This attribute can be modified with the PATCH method.
vnfInstanceDescription	String	0..1	Human-readable description of the VNF instance. This attribute can be modified with the PATCH method.
vnfdId	Identifier	1	Identifier of the VNFD on which the VNF instance is based. See note 1.
vnfProvider	String	1	Provider of the VNF and the VNFD. The value is copied from the VNFD.
vnfProductName	String	1	Name to identify the VNF Product. The value is copied from the VNFD.
vnfSoftwareVersion	Version	1	Software version of the VNF. The value is copied from the VNFD.
vnfdVersion	Version	1	Identifies the version of the VNFD. The value is copied from the VNFD.
vnfConfigurableProperties	KeyValuePairs	0..1	Additional VNF-specific attributes that provide the current values of the configurable properties of the VNF instance. These attributes represent values that are stored persistently in the VnfInstance structure and that correspond to configuration parameters of the VNF instance. Modifying these attributes affects the configuration of the VNF instance either directly (if the VNF instance is in INSTANTIATED state at the time of the modification) or as part of the subsequent VNF instantiation operation (if the VNF instance is in NOT_INSTANTIATED state at the time of the modification). Configurable properties referred in these attributes are declared in the VNFD. The declaration of configurable properties in the VNFD can optionally contain the specification of initial values. See notes 2, 3 and 4. The VNFM shall reject requests to write configurable properties that are not declared in the VNFD with a "422 Unprocessable entity" error response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8]. These configurable properties include the following standard attributes, which are declared in the VNFD if auto-scaling and/or auto-healing are supported by the VNF: - isAutoscaleEnabled: If present, the VNF supports auto-scaling. If set to true, auto-scaling is currently enabled. If set to false, auto-scaling is currently disabled. - isAutohealEnabled: If present, the VNF supports auto-healing. If set to true, auto-healing is currently enabled. If set to false, auto-healing is currently disabled.

Attribute name	Data type	Cardinality	Description
			These configurable properties can be initialized with default values from the VNFD (see note 4). Configurable properties can be modified with values passed in the request structures of certain LCM operations, such as the InstantiateVnfRequest structure. Further, these configurable properties can be created, modified or deleted with the PATCH method. In addition, the provisions in clause 5.7 shall apply.
vimConnectionInfo	map(VimConnectionInfo)	0..N	Information about VIM or CISM connections to be used for managing the resources for the VNF instance. The keys of the map, each of which identifies information about a particular VIM connection, are managed by the NFVO and referenced from other data structures via the "vimConnectionId" attribute. This attribute shall only be supported and present if - the resources of at least of the VNFCs are managed by a VIM and VNF-related resource management in direct mode is applicable. - the resources of at least of the VNFCs are managed by a CISM. This attribute can be modified with the PATCH method.
cirConnectionInfo	map(VimConnectionInfo)	0..N	Information about the CIR connection for managing OS container images for the VNF instance. Shall be present when all or part of the VNF is realized by a set of OS containers and shall be absent otherwise.
mciopRepositoryInfo	map(VimConnectionInfo)	0..N	Information about the MCIOP repository for the VNF instance. Shall be present when all or part of the VNF is realized by a set of OS containers and shall be absent otherwise. See note 1.
instantiationState	Enum (inlined)	1	The instantiation state of the VNF. Permitted values: - NOT_INSTANTIATED: The VNF instance is terminated or not instantiated. - INSTANTIATED: The VNF instance is instantiated.
instantiatedVnfInfo	Structure (inlined)	0..1	Information specific to an instantiated VNF instance. This attribute shall be present if the instantiateState attribute value is INSTANTIATED.

Attribute name	Data type	Cardinality	Description
>flavourId	IdentifierInVnfd	1	Identifier of the VNF deployment flavour applied to this VNF instance.
>vnfState	VnfOperationalStateType	1	State of the VNF instance.
>scaleStatus	ScaleInfo	0..N	Scale status of the VNF, one entry per aspect. Represents for every scaling aspect how "big" the VNF has been scaled with reference to that aspect. This attribute shall be present if the VNF supports scaling. See clause B.2 for an explanation of VNF scaling.
>maxScaleLevels	ScaleInfo	0..N	Maximum allowed scale levels of the VNF, one entry per aspect. This attribute shall be present if the VNF supports scaling.
>extCpInfo	VnfExtCpInfo	1..N	Information about the external CPs exposed by the VNF instance. When trunking is enabled, the list of entries includes both, external CPs corresponding to parent ports of a trunk, and external CPs associated to sub-ports of a trunk.
>vipCpInfo	VipCpInfo	0..N	VIP CPs that are part of the VNF instance. Shall be present when that particular VIP CP of the VNFC instance is associated to an external CP of the VNF instance. May be present otherwise.
>virtualCpInfo	VirtualCpInfo	0..N	Virtual CPs that are part of the VNF instance. Shall be present when a particular Virtual CP is associated to an external CP of the VNF instance. May be present otherwise.
>extVirtualLinkInfo	ExtVirtualLinkInfo	0..N	Information about the external VLs the VNF instance is connected to.
>extManagedVirtualLinkInfo	ExtManagedVirtualLinkInfo	0..N	Information about the externally-managed internal VLs of the VNF instance. See notes 5 and 6.
>monitoringParameters	MonitoringParameter	0..N	Active monitoring parameters.
>localizationLanguage	String	0..1	Information about localization language of the VNF (includes e.g. strings in the VNFD). The localization languages supported by a VNF can be declared in the VNFD, and localization language selection can take place at instantiation time. The value shall comply with the format defined in IETF RFC 5646 [3].
>vnfcResourceInfo	VnfcResourceInfo	0..N	Information about the virtualised compute and storage resources used by the VNFCs of the VNF instance.
>vnfVirtualLinkResourceInfo	VnfVirtualLinkResourceInfo	0..N	Information about the virtualised network resources used by the VLs of the VNF instance. See note 6.

Attribute name	Data type	Cardinality	Description
>virtualStorageResourceInfo	VirtualStorageResourceInfo	0..N	Information about the virtualised storage resources used as storage for the VNF instance.
>mcioInfo	McioInfo	0..N	Information on the MCIO(s) representing VNFC instance(s) realized by one or a set of OS containers and created from the same VDU for the VNF instance.
metadata	KeyValuePairs	0..1	Additional VNF-specific attributes that provide metadata describing the VNF instance. These attributes represent values that are stored persistently in the VnfInstance structure for consumption by functional blocks that invoke the VNF lifecycle management interface. They are not consumed by the VNFM, or the lifecycle management scripts. Modifying the values of these attributes has no effect on the VNF instance, it only affects the information represented in the VnfInstance structure. Metadata that the VNF provider foresees are expected to be declared in the VNFD. The declaration of metadata in the VNFD can optionally contain the specification of initial values. See notes 2 and 4. The VNFM shall accept requests to write metadata that are not declared in the VNFD. These attributes can be initialized with default values from the VNFD (see note 4) or with values passed in the CreateVnfRequest structure (see clause 5.4.2.3.1). These attributes can be created, modified or removed with the PATCH method.

Attribute name	Data type	Cardinality	Description
extensions	KeyValuePairs	0..1	Additional VNF-specific attributes that affect the lifecycle management of this VNF instance. These attributes represent values that are stored persistently in the VnfInstance structure for consumption by the VNFM or the lifecycle management scripts during the execution of VNF lifecycle management operations. All extensions that are allowed for the VNF are declared in the VNFD. The declaration of an extension in the VNFD contains information on whether its presence is optional or required, and optionally can specify an initial value. See notes 2 and 4. The VNFM shall reject requests to write extension attributes that are not declared in the VNFD with a "422 Unprocessable entity" error response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8]. Modifying the values of these attributes has no direct effect on the VNF instance; however, the modified attribute values can be considered during subsequent VNF lifecycle management operations, which means that the modified values can indirectly affect the configuration of the VNF instance. These attributes can be initialized with default values from the VNFD (see note 4). These attributes can be modified with values passed in the request structures of certain LCM operations, such as the InstantiateVnfRequest structure. Further, these attributes can be created, modified or deleted with the PATCH method. In addition, the provisions in clause 5.7 shall apply.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>indicators	Link	0..1	Indicators related to this VNF instance, if applicable.
>instantiate	Link	0..1	Link to the "Instantiate VNF task" resource, if the related operation is possible based on the current status of this VNF instance resource (i.e. VNF instance in NOT_INSTANTIATED state).
>terminate	Link	0..1	Link to the "Terminate VNF task" resource, if the related operation is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).
>scale	Link	0..1	Link to the "Scale VNF task" resource, if the related operation is supported for this VNF instance and is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).

Attribute name	Data type	Cardinality	Description
>scaleToLevel	Link	0..1	Link to the "Scale VNF to level task" resource, if the related operation is supported for this VNF instance and is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).
>changeFlavour	Link	0..1	Link to the "Change VNF flavour task" resource, if the related operation is supported for this VNF instance and is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).
>heal	Link	0..1	Link to the "Heal VNF task" resource, if the related operation is supported for this VNF instance and is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).
>operate	Link	0..1	Link to the "Operate VNF task" resource, if the related operation is supported for this VNF instance and is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).
>changeExtConn	Link	0..1	Link to the "Change external VNF connectivity task" resource, if the related operation is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).
>createSnapshot	Link	0..1	Link to the "Create VNF snapshot task" resource, if the related operation is supported for this VNF instance and is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).
>revertToSnapshot	Link	0..1	Link to the "Revert to VNF snapshot task" resource, if the related operation is supported for this VNF instance and is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).
>changeCurrentVnfPkg	Link	0..1	Link to the "Change current VNF package task" resource, if the related operation is possible based on the current status of this VNF instance resource (i.e. VNF instance is in INSTANTIATED state).

NOTE 1: Modifying the value of this attribute shall not be performed when conflicts exist between the previous and the newly referred VNF package, i.e. when the new VNFD is changed with respect to the previous VNFD in other aspects than merely referencing to other VNF software images. In order to avoid misalignment of the VnfInstance with the current VNF's on-boarded VNF Package, the values of attributes in the VnfInstance that have corresponding attributes in the VNFD shall be kept in sync with the values in the VNFD.

NOTE 2: ETSI GS NFV-SOL 001 [i.4] specifies the structure and format of the VNFD based on TOSCA specifications.

NOTE 3: VNF configurable properties are sometimes also referred to as configuration parameters applicable to a VNF. Some of these are set prior to instantiation and cannot be modified if the VNF is instantiated, some are set prior to instantiation (are part of initial configuration) and can be modified later, and others can be set only after instantiation. The applicability of certain configuration may depend on the VNF and the required operation of the VNF at a certain point in time.

NOTE 4: Upon creation of the VnfInstance structure, the VNFM shall create and initialize all child attributes of "vnfConfigurableProperties", "metadata" and "extensions" that were declared in the VNFD with a defined initial value. The defined initial values can be declared in the VNFD, and/or, in case of "metadata", obtained from the "CreateVnfRequest" structure. Child attributes of "vnfConfigurableProperties", "metadata" and "extensions" that have no defined initial value shall not be created, in order to be consistent with the semantics of the JSON Merge Patch method (see IETF RFC 7396 [5]) that interprets null values as deletion request.

Attribute name	Data type	Cardinality	Description
NOTE 5: It is possible to have several ExtManagedVirtualLinkInfo for the same VNF internal VL in case of a multi-site VNF spanning several VIMs. The set of ExtManagedVirtualLinkInfo corresponding to the same VNF internal VL shall indicate so by referencing to the same VnfVirtualLinkDesc and externally-managed multi-site VL instance (refer to clause 5.5.3.3). NOTE 6: Even though externally-managed internal VLs are also used for VNF-internal connectivity, they shall not be listed in the "vnfVirtualLinkResourceInfo" attribute as this would be redundant.			

5.5.2.3 Type: CreateVnfRequest

This type represents request parameters for the "Create VNF identifier" operation. It shall comply with the provisions defined in table 5.5.2.3-1.

Table 5.5.2.3-1: Definition of the CreateVnfRequest data type

Attribute name	Data type	Cardinality	Description
vnfdId	Identifier	1	Identifier that identifies the VNFD which defines the VNF instance to be created.
vnfInstanceName	String	0..1	Human-readable name of the VNF instance to be created.
vnfInstanceDescription	String	0..1	Human-readable description of the VNF instance to be created.
metadata	KeyValuePairs	0..1	If present, this attribute provides additional initial values, overriding those obtained from the VNFD, for the "metadata" attribute in "VnfInstance", as defined in clause 5.5.2.2. Provisions for handling metadata during the operation are defined in clause 5.4.2.3.1.

5.5.2.4 Type: InstantiateVnfRequest

This type represents request parameters for the "Instantiate VNF" operation. It shall comply with the provisions defined in table 5.5.2.4-1.

Table 5.5.2.4-1: Definition of the InstantiateVnfRequest data type

Attribute name	Data type	Cardinality	Description
flavourId	IdentifierInVnfd	1	Identifier of the VNF deployment flavour to be instantiated.
instantiationLevelId	IdentifierInVnfd	0..1	Identifier of the instantiation level of the deployment flavour to be instantiated. See note 3.
targetScaleLevelInfo	ScaleInfo	0..N	This attribute is applicable if VNF supports target scale level instantiation. For each scaling aspect of the current deployment flavour, the attribute specifies the scale level of VNF constituents (e.g. VDU level) to be instantiated. See notes 3 and 4.
extVirtualLinks	ExtVirtualLinkData	0..N	Information about external VLs to connect the VNF to, including configuration information for the CPs via which the VNF instance can attach to this VL. The following applies to the "ExtVirtualLinkData" information provided in this request, together with the the related overriding information provided in the "Grant" structure (see clause 9.5.2.3): Even if the VNF is not instantiated in fully scaled-out state, the API consumer shall provide enough CP configuration records to allow connecting the VNF instance, fully scaled out in all scaling aspects, to the external VLs.
extManagedVirtualLinks	ExtManagedVirtualLinkData	0..N	Information about internal VLs that are managed by the NFVO. See note 1 and note 2.

Attribute name	Data type	Cardinality	Description
vimConnectionInfo	map(VimConnectionInfo)	0..N	Information about VIM or CISM connections to be used for managing the resources for the VNF instance, or refer to external/externally-managed virtual links. This attribute shall only be supported and may be present if - the resources for at least one of the VNFCs shall be managed by a VIM and VNF-related resource management in direct mode is applicable. - the resources for at least one of the VNFCs shall be managed by a CISM. The VNFM shall apply the content of this attribute to the "vimConnectionInfo" attribute of "VnfInstance" according to the rules of JSON Merge Patch (see IETF RFC 7396 [5]).
localizationLanguage	String	0..1	Localization language of the VNF to be instantiated. The value shall comply with the format defined in IETF RFC 5646 [3].
additionalParams	KeyValuePairs	0..1	Additional input parameters for the instantiation process, specific to the VNF being instantiated, as declared in the VNFD as part of "InstantiateVnfOpConfig" defined in ETSI GS NFV-IFA 011 [10].
extensions	KeyValuePairs	0..1	If present, this attribute provides modifications to the default values, as obtained from the VNFD, of the "extensions" attribute in "VnfInstance", as defined in clause 5.5.2.2. Provisions for handling extensions during the operation are defined in clause 5.4.4.3.1.
vnfConfigurableProperties	KeyValuePairs	0..1	If present, this attribute provides modifications to the default values, as obtained from the VNFD, of the "vnfConfigurableProperties" attribute in "VnfInstance", as defined in clause 5.5.2.2. Provisions for handling configurable properties during the operation are defined in clause 5.4.4.3.1.

NOTE 1: The indication of externally-managed internal VLs is needed in case networks have been pre-configured for use with certain VNFs, for instance to ensure that these networks have certain properties such as security or acceleration features, or to address particular network topologies. The present document assumes that externally-managed internal VLs are managed by the NFVO and created towards the VIM.

NOTE 2: It is possible to have several ExtManagedVirtualLinkData for the same VNF internal VL in case of a multi-site VNF spanning several VIMs. The set of ExtManagedVirtualLinkData corresponding to the same VNF internal VL shall indicate so by referencing to the same VnfVirtualLinkDesc and externally-managed multi-site VL instance (refer to clause 4.4.1.12).

NOTE 3: The target size for VNF instantiation may be specified in either instantiationLevelId or targetScaleLevelInfo, but not both. If none of the two attributes (instantiationLevelId or targetScaleLevelInfo) are present, the default instantiation level as declared in the VNFD shall be used.

NOTE 4: If targetScaleLevelInfo is specified, information provided in targetScaleLevelInfo shall be used for instantiating scalable constituents of the VNF (e.g, VDUs/VLs). For scaling aspects not specified in targetScaleLevelInfo or for the VNF constituents (e.g.VDUs/VLs) that are not scalable, the default instantiation level as declared in the VNFD shall be used for instantiation.

5.5.2.5 Type: ScaleVnfRequest

This type represents request parameters for the "Scale VNF" operation. It shall comply with the provisions defined in table 5.5.2.5-1. See clause B.2 in annex B for an explanation of VNF scaling.

Table 5.5.2.5-1: Definition of the ScaleVnfRequest data type

Attribute name	Data type	Cardinality	Description
type	Enum (inlined)	1	Indicates the type of the scale operation requested. Permitted values: - SCALE_OUT: adding additional VNFC instances to the VNF to increase capacity. - SCALE_IN: removing VNFC instances from the VNF in order to release unused capacity.
aspectId	IdentifierInVnfd	1	Identifier of the scaling aspect.
numberOfSteps	Integer	0..1	Number of scaling steps to be executed as part of this Scale VNF operation. It shall be a positive number and the default value shall be 1.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the NFVO as input to the scaling process, specific to the VNF being scaled, as declared in the VNFD as part of "ScaleVnfOpConfig" defined in ETSI GS NFV-IFA 011 [10].

5.5.2.6 Type: ScaleVnfToLevelRequest

This type represents request parameters for the "Scale VNF to Level" operation. It shall comply with the provisions defined in table 5.5.2.6-1. See clause B.2 for an explanation of VNF scaling.

Table 5.5.2.6-1: Definition of the ScaleVnfToLevelRequest data type

Attribute name	Data type	Cardinality	Description
instantiationLevelId	IdentifierInVnfd	0..1	Identifier of the target instantiation level of the current deployment flavour to which the VNF is requested to be scaled. See note.
scaleInfo	ScaleInfo	0..N	For each scaling aspect of the current deployment flavour, indicates the target scale level to which the VNF is to be scaled. See note.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the NFVO as input to the scaling process, specific to the VNF being scaled, as declared in the VNFD as part of "ScaleVnfToLevelOpConfig" defined in ETSI GS NFV-IFA 011 [10].
NOTE: Either the instantiationLevelId attribute or the scaleInfo attribute shall be included.			

5.5.2.7 Type: ChangeVnfFlavourRequest

This type represents request parameters for the "Change VNF flavour" operation. It shall comply with the provisions defined in table 5.5.2.7-1.

Table 5.5.2.7-1: Definition of the ChangeVnfFlavourRequest data type

Attribute name	Data type	Cardinality	Description
newFlavourId	IdentifierInVnfd	1	Identifier of the VNF deployment flavour to be instantiated.
instantiationLevelId	IdentifierInVnfd	0..1	Identifier of the instantiation level of the deployment flavour to be instantiated. See note 3.If not present, the default instantiation level as declared in the VNFD is instantiated.

Attribute name	Data type	Cardinality	Description
targetScaleLevelInfo	ScaleInfo	0..N	This attribute is applicable if VNF supports target scale level instantiation. For each scaling aspect of the current deployment flavour, the attribute specifies the scale level of VNF constituents (e.g. VDU level) to be instantiated. See notes 3 and 4.
extVirtualLinks	ExtVirtualLinkData	0..N	Information about external VLs to connect the VNF to, including configuration information for the CPs via which the VNF instance can attach to this VL. Entries in the list of external VLs that are unchanged need not be supplied as part of this request. The following applies to the "ExtVirtualLinkData" information provided in this request, together with the related "ExtVirtualLinkInfo" information known to the VNFM represented in the "VnfInstance" structure (see clause 5.5.2.2) and the related overriding information provided in the "Grant" structure (see clause 9.5.2.3): Even if the VNF is not in fully scaled-out state after changing the flavour, the API consumer shall provide enough CP configuration records to allow connecting the VNF instance, fully scaled out in all scaling aspects, to the external VLs.
extManagedVirtualLinks	ExtManagedVirtualLinkData	0..N	Information about internal VLs that are managed by the NFVO. See notes 1 and 2.
vimConnectionInfo	map(VimConnectionInfo)	0..N	Information about VIM or CISM connections to be used for managing the resources for the VNF instance, or refer to external/externally-managed virtual links. This attribute shall only be supported and may be present if - the resources for at least one of the VNFCs shall be managed by a VIM and VNF-related resource management in direct mode is applicable. - the resources for at least one of the VNFCs shall be managed by a CISM. The VNFM shall apply the content of this attribute to the "vimConnectionInfo" attribute of "VnfInstance" according to the rules of JSON Merge Patch (see IETF RFC 7396 [5]).
additionalParams	KeyValuePairs	0..1	Additional input parameters for the flavour change process, specific to the VNF being modified, as declared in the VNFD as part of "ChangeVnfFlavourOpConfig" defined in ETSI GS NFV-IFA 011 [10].
extensions	KeyValuePairs	0..1	If present, this attribute provides modifications to the values of the "extensions" attribute in "VnfInstance", as defined in clause 5.5.2.2. Provisions for handling extensions during the operation, are defined in clause 5.4.7.3.1.
vnfConfigurableProperties	KeyValuePairs	0..1	If present, this attribute provides modifications to the values of the "vnfConfigurableProperties" attribute in "VnfInstance", as defined in clause 5.5.2.2. Provisions for handling VNF configurable properties during the operation, are defined in clause 5.4.7.3.1.

NOTE 1: The indication of externally-managed internal VLs is needed in case networks have been pre-configured for use with certain VNFs, for instance to ensure that these networks have certain properties such as security or acceleration features, or to address particular network topologies. The present document assumes that externally-managed internal VLs are managed by the NFVO and created towards the VIM.

NOTE 2: It is possible to have several ExtManagedVirtualLinkData for the same VNF internal VL in case of a multi-site VNF spanning several VIMs. The set of ExtManagedVirtualLinkData corresponding to the same VNF internal VL shall indicate so by referencing to the same VnfVirtualLinkDesc and externally-managed multi-site VL instance (refer to clause 4.4.1.12).

Attribute name	Data type	Cardinality	Description
NOTE 3: The target size for VNF instantiation may be specified in either instantiationLevelId or targetScaleLevelInfo, but not both. If none of the two attributes (instantiationLevelId or targetScaleLevelInfo) are present, the default instantiation level as declared in the VNFD shall be used. NOTE 4: If targetScaleLevelInfo is specified, information provided in targetScaleLevelInfo shall be used for instantiating scalable constituents of the VNF (e.g, VDUs/VLs). For scaling aspects not specified in targetScaleLevelInfo or for the VNF constituents (e.g.VDUs/VLs) that are not scalable, the default instantiation level as declared in the VNFD shall be used for instantiation.			

5.5.2.8 Type: TerminateVnfRequest

This type represents request parameters for the "Terminate VNF" operation. It shall comply with the provisions defined in table 5.5.2.8-1.

Table 5.5.2.8-1: Definition of the TerminateVnfRequest data type

Attribute name	Data type	Cardinality	Description
terminationType	Enum (inlined)	1	Indicates whether forceful or graceful termination is requested. See note. Permitted values: - FORCEFUL: The VNFM will shut down the VNF and release the resources immediately after accepting the request. - GRACEFUL: The VNFM will first arrange to take the VNF out of service after accepting the request. Once the operation of taking the VNF out of service finishes (irrespective of whether it has succeeded or failed) or once the timer value specified in the "gracefulTerminationTimeout" attribute expires, the VNFM will shut down the VNF and release the resources.
gracefulTermination Timeout	Integer	0..1	This attribute is only applicable in case of graceful termination. It defines the time to wait for the VNF to be taken out of service before shutting down the VNF and releasing the resources. The unit is seconds. If not given and the "terminationType" attribute is set to "GRACEFUL", it is expected that the VNFM waits for the successful taking out of service of the VNF, no matter how long it takes, before shutting down the VNF and releasing the resources.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the NFVO as input to the termination process, specific to the VNF being terminated, as declared in the VNFD as part of "TerminateVnfOpConfig" defined in ETSI GS NFV-IFA 011 [10].
NOTE: If the VNF is still in service, requesting forceful termination can adversely impact network service.			

5.5.2.9 Type: HealVnfRequest

This type represents request parameters for the "Heal VNF" operation. It shall comply with the provisions defined in table 5.5.2.9-1.

Table 5.5.2.9-1: Definition of the HealVnfRequest data type

Attribute name	Data type	Cardinality	Description
cause	String	0..1	Indicates the reason why a healing procedure is required.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the NFVO as input to the healing process, specific to the VNF being healed, as declared in the VNFD as part of "HealVnfOpConfig" defined in ETSI GS NFV-IFA 011 [10].

5.5.2.10 Type: OperateVnfRequest

This type represents request parameters for the "Operate VNF" operation. It shall comply with the provisions defined in table 5.5.2.10-1.

Table 5.5.2.10-1: Definition of the OperateVnfRequest data type

Attribute name	Data type	Cardinality	Description
changeStateTo	VnfOperationalStateType	1	The desired operational state (i.e. started or stopped) to change the VNF to.
stopType	StopType	0..1	It signals whether forceful or graceful stop is requested. See note.
gracefulStopTimeout	Integer	0..1	The time interval (in seconds) to wait for the VNF to be taken out of service during graceful stop, before stopping the VNF. See note.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the NFVO as input to the process, specific to the VNF of which the operation status is changed, as declared in the VNFD as part of "OperateVnfOpConfig" defined in ETSI GS NFV-IFA 011 [10].
NOTE: The "stopType" and "gracefulStopTimeout" attributes shall be absent, when the "changeStateTo" attribute is equal to "STARTED". The "gracefulStopTimeout" attribute shall be present, when the "changeStateTo" is equal to "STOPPED" and the "stopType" attribute is equal to "GRACEFUL". The "gracefulStopTimeout" attribute shall be absent, when the "changeStateTo" attribute is equal to "STOPPED" and the "stopType" attribute is equal to "FORCEFUL". The request shall be treated as if the "stopType" attribute has been set to "FORCEFUL", when the "changeStateTo" attribute is equal to "STOPPED" and the "stopType" attribute is absent.			

5.5.2.11 Type: ChangeExtVnfConnectivityRequest

This type represents request parameters for the "Change external VNF connectivity" operation to modify the external connectivity of a VNF instance. It shall comply with the provisions defined in table 5.5.2.11-1.

Table 5.5.2.11-1: Definition of the ChangeExtVnfConnectivityRequest data type

Attribute name	Data type	Cardinality	Description
extVirtualLinks	ExtVirtualLinkData	1..N	Information about external VLs to change (e.g. connect the VNF to) including configuration information for the CPs via which the VNF instance can attach to this VL. Entries in the list of external VLs that are unchanged need not be supplied as part of this request. The following applies to the "ExtVirtualLinkData" information provided in this request, together with the related "ExtVirtualLinkInfo" information known to the VNFM represented in the "VnfInstance" structure (see clause 5.5.2.2) and the related overriding information provided in the "Grant" structure (see clause 9.5.2.3): Even if the VNF is not in fully scaled-out state, the API consumer shall provide enough CP configuration records to allow connecting the VNF instance, fully scaled out in all scaling aspects, to the external VLs.
vimConnectionInfo	map(VimConnectionInfo)	0..N	Information about VIM or CISM connections to be used for managing the resources for the VNF instance, or refer to external virtual links. This attribute shall only be supported and may be present if: - the resources for at least one of the VNFCs shall be managed by a VIM and VNF-related resource management in direct mode is applicable. - the resources for at least one of the VNFCs shall be managed by a CISM. The VNFM shall apply the content of this attribute to the "vimConnectionInfo" attribute of "VnfInstance" according to the rules of JSON Merge Patch (see IETF RFC 7396 [5]).
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the NFVO as input to the process, specific to the VNF of which the external connectivity is changed, as declared in the VNFD as part of "ChangeExtVnfConnectivityOpConfig" defined in ETSI GS NFV-IFA 011 [10].

The following behaviour applies for the changes that can be performed with this operation:

- To change the connection of external CP instances based on certain external CPDs from a "source" external VL to a different "target" external VL, the identifier of the "target" external VL shall be sent in the "extVirtualLinkId" attribute of the "extVirtualLinks" parameter, and the "extCps" attributes of that parameter shall refer via the "cpdId" attribute to the external CPDs of the corresponding external connection point instances that are to be reconnected to the target external VL.

NOTE: For CP instances that are not part of a trunk, this means that all CP instances based on a given external CPD will be reconnected. See clause B.3.3 for an illustration. Likewise, for CP instances that are part of a trunk and have the same segmentationId, all CP instances (subports) based on a given external CPD will be connected, disconnected or reconnected.

- To change the connectivity parameters of the external CPs connected to a particular external VL, including changing addresses, the identifier of that external VL shall be sent in the "extVirtualLinkId" attribute of the "extVirtualLinks" parameter, and the "extCps" attribute of that parameter shall contain at least those entries with modified parameters.

5.5.2.11a Type: ChangeCurrentVnfPkgRequest

This type represents request parameters for the "Change current VNF package" operation to replace the VNF package on which a VNF instance is based. It shall comply with the provisions defined in table 5.5.2.11a-1.

Table 5.5.2.11a-1: Definition of the ChangeCurrentVnfPkgRequest data type

Attribute name	Data type	Cardinality	Description
vnfdId	Identifier	1	Identifier of the VNFD which defines the destination VNF Package for the change.
extVirtualLinks	ExtVirtualLinkData	0..N	Information about external VLs to connect the VNF to, including configuration information for the CPs via which the VNF instance can attach to this VL. Entries in the list that are unchanged need not be supplied as part of this request. The following applies to the "ExtVirtualLinkData" information provided in this request, together with the related "ExtVirtualLinkInfo" information known to the VNFM represented in the "VnfInstance" structure (see clause 5.5.2.2) and the related overriding information provided in the "Grant" structure (see clause 9.5.2.3): Even if the VNF is not in fully scaled-out state after the change of the VNF package, the API consumer shall provide enough CP configuration records to allow connecting the VNF instance, fully scaled out in all scaling aspects, to the external VLs.
extManagedVirtualLinks	ExtManagedVirtualLinkData	0..N	Information about internal VLs that are managed by the NFVO. See notes 1 and 2.
vimConnectionInfo	map(VimConnectionInfo)	0..N	Information about VIM or CISM connections to be used for managing the resources for the VNF instance, or refer to external virtual links. This attribute shall only be supported and may be present if - the resources for at least one of the VNFCs shall be managed by a VIM and VNF-related resource management in direct mode is applicable. - the resources for at least one of the VNFCs shall be managed by a CISM. The VNFM shall apply the content of this attribute to the "vimConnectionInfo" attribute of "VnfInstance" according to the rules of JSON Merge Patch (see IETF RFC 7396 [5]).
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the NFVO as input to the process, specific to the VNF of which the underlying VNF package is changed, as declared in the VNFD as part of "ChangeCurrentVnfPkgOpConfig" defined in ETSI GS NFV-IFA 011 [10].
extensions	KeyValuePairs	0..1	If present, this attribute provides modifications to the values of the "extensions" attribute in "VnfInstance", as defined in clause 5.5.2.2. Provisions for handling extensions during the operation, and needed passed parameter values in case of conflicts, are defined in clause 5.4.11a.3.1.

Attribute name	Data type	Cardinality	Description
vnfConfigurableProperties	KeyValuePairs	0..1	If present, this attribute provides modifications to the values of the "vnfConfigurableProperties" attribute in "VnfInstance", as defined in clause 5.5.2.2. Provisions for handling VNF configurable properties during the operation, and needed passed parameter values in case of conflicts, are defined in clause 5.4.11a.3.1.
NOTE 1: The indication of externally-managed internal VLs is needed in case networks have been pre-configured for use with certain VNFs, for instance to ensure that these networks have certain properties such as security or acceleration features, or to address particular network topologies. The present document assumes that externally-managed internal VLs are managed by the NFVO and created towards the VIM. NOTE 2: It is possible to have several ExtManagedVirtualLinkData for the same VNF internal VL in case of a multi-site VNF spanning several VIMs. The set of ExtManagedVirtualLinkData corresponding to the same VNF internal VL shall indicate so by referencing to the same VnfVirtualLinkDesc and externally-managed multi-site VL instance (refer to clause 4.4.1.12).			

5.5.2.12 Type: VnfInfoModificationRequest

This type represents attribute modifications for an "Individual VNF instance" resource, i.e. modifications to a resource representation based on the "VnfInstance" data type. The attributes of "VnfInstance" that can be modified according to the provisions in clause 5.5.2.2 are included in the "VnfInfoModificationRequest" data type.

The "VnfInfoModificationRequest" data type shall comply with the provisions defined in table 5.5.2.12-1.

Table 5.5.2.12-1: Definition of the VnfInfoModificationRequest data type

Attribute name	Data type	Cardinality	Description
vnfInstanceName	String	0..1	New value of the "vnfInstanceName" attribute in "VnfInstance", or "null" to remove the attribute.
vnfInstanceDescription	String	0..1	New value of the "vnfInstanceDescription" attribute in "VnfInstance", or "null" to remove the attribute.
vnfdId	Identifier	0..1	New value of the "vnfdId" attribute in "VnfInstance". The value "null" is not permitted.
vnfConfigurableProperties	KeyValuePairs	0..1	Modifications of the "vnfConfigurableProperties" attribute in "VnfInstance". If present, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [5]).
metadata	KeyValuePairs	0..1	Modifications of the "metadata" attribute in "VnfInstance". If present, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [5]).
extensions	KeyValuePairs	0..1	Modifications of the "extensions" attribute in "VnfInstance". If present, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [5]).
vimConnectionInfo	map(VimConnectionInfo)	0..N	Modifications of the "vimConnectionInfo" attribute. If present, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [5]).

5.5.2.12a Type: VnfInfoModifications

This type represents attribute modifications that were performed on an "Individual VNF instance" resource. The attributes that can be included consist of those requested to be modified explicitly in the "VnfInfoModificationRequest" data structure, and additional attributes of the "VnfInstance" data structure that were modified implicitly e.g. when modifying the referenced VNF package.

The "VnfInfoModifications" data type shall comply with the provisions defined in table 5.5.2.12a-1.

Table 5.5.2.12a-1: Definition of the VnfInfoModifications data type

Attribute name	Data type	Cardinality	Description
vnfInstanceName	String	0..1	If present, this attribute signals modifications of the "vnfInstanceName" attribute in "VnfInstance" as defined in clause 5.5.2.12.
vnfInstanceDescription	String	0..1	If present, this attribute signals modifications of the "vnfInstanceDescription" attribute in "VnfInstance", as defined in clause 5.5.2.12.
vnfConfigurableProperties	KeyValuePairs	0..1	If present, this attribute signals modifications of the "vnfConfigurableProperties" attribute in "VnfInstance", as defined in clause 5.5.2.12. In addition, the provisions in clause 5.7 shall apply.
metadata	KeyValuePairs	0..1	If present, this attribute signals modifications of the "metadata" attribute in "VnfInstance", as defined in clause 5.5.2.12.
extensions	KeyValuePairs	0..1	If present, this attribute signals modifications of the "extensions" attribute in "VnfInstance", as defined in clause 5.5.2.12. In addition, the provisions in clause 5.7 shall apply.
vimConnectionInfo	map(VimConnectionInfo)	0..N	If present, this attribute signals modifications of the "vimConnectionInfo" attribute array in "VnfInstance", as defined in clause 5.5.2.12.
vnfdId	Identifier	0..1	If present, this attribute signals modifications of the "vnfdId" attribute in "VnfInstance", as defined in clause 5.5.2.12.
vnfProvider	String	0..1	If present, this attribute signals modifications of the "vnfProvider" attribute in "VnfInstance". See note.
vnfProductName	String	0..1	If present, this attribute signals modifications of the "vnfProductName" attribute in "VnfInstance". See note.
vnfSoftwareVersion	Version	0..1	If present, this attribute signals modifications of the "vnfSoftwareVersion" attribute in "VnfInstance". See note.
vnfdVersion	Version	0..1	If present, this attribute signals modifications of the "vnfdVersion" attribute in "VnfInstance". See note.
NOTE: If present, this attribute (which depends on the value of the "vnfdId" attribute) was modified implicitly following a request to modify the "vnfdId" attribute, by copying the value of this attribute from the VNFD in the VNF Package identified by the "vnfdId" attribute.			

5.5.2.13 Type: VnfLcmOpOcc

This type represents a VNF lifecycle management operation occurrence. It shall comply with the provisions defined in table 5.5.2.13-1.

Table 5.5.2.13-1: Definition of the VnfLcmOpOcc data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this VNF lifecycle management operation occurrence.
operationState	LcmOperationStateType	1	The state of the LCM operation.
stateEnteredTime	DateTime	1	Date-time when the current state has been entered.
startTime	DateTime	1	Date-time of the start of the operation.
vnfInstanceId	Identifier	1	Identifier of the VNF instance to which the operation applies.
grantId	Identifier	0..1	Identifier of the grant related to this VNF LCM operation occurrence. Shall be set to the value of the "id" attribute in the "Grant" representing the associated "Individual Grant", if such grant exists.
operation	LcmOperationType	1	Type of the actual LCM operation represented by this VNF LCM operation occurrence.

Attribute name	Data type	Cardinality	Description
isAutomaticInvocation	Boolean	1	Set to true if this VNF LCM operation occurrence has been triggered by an automated procedure inside the VNFM (i.e. ScaleVnf/ScaleVnfToLevel triggered by auto-scale, or HealVnf triggered by auto-heal). Set to false otherwise.
operationParams	Object	0..1	Input parameters of the LCM operation. This attribute shall be formatted according to the request data type of the related LCM operation. In addition, the provisions in clause 5.7 shall apply. The following mapping between operationType and the data type of this attribute shall apply: • INSTANTIATE: InstantiateVnfRequest • SCALE: ScaleVnfRequest • SCALE_TO_LEVEL: ScaleVnfToLevelRequest • CHANGE_FLAVOUR: ChangeVnfFlavourRequest • OPERATE: OperateVnfRequest • HEAL: HealVnfRequest • CHANGE_EXT_CONN: ChangeExtVnfConnectivityRequest • TERMINATE: TerminateVnfRequest • MODIFY_INFO: VnfInfoModificationRequest • CREATE_SNAPSHOT: CreateVnfSnapshotRequest • REVERT_TO_SNAPSHOT: RevertToVnfSnapshotRequest • CHANGE_VNFPKG: ChangeCurrentVnfPkgRequest This attribute shall be present if this data type is returned in a response to reading an individual resource and may be present according to the chosen attribute selector parameter if this data type is returned in a response to a query of a container resource.
isCancelPending	Boolean	1	If the VNF LCM operation occurrence is in "STARTING", "PROCESSING" or "ROLLING_BACK" state and the operation is being cancelled, this attribute shall be set to true. Otherwise, it shall be set to false.
cancelMode	CancelModeType	0..1	The mode of an ongoing cancellation. Shall be present when isCancelPending=true and shall be absent otherwise.
error	ProblemDetails	0..1	If "operationState" is "FAILED_TEMP" or "FAILED" or "operationState" is "PROCESSING" or "ROLLING_BACK" and previous value of "operationState" was "FAILED_TEMP", this attribute shall be present and contain error information, unless it has been requested to be excluded via an attribute selector.
resourceChanges	Structure (inlined)	0..1	This attribute contains information about the cumulative changes to virtualised resources that were performed so far by the LCM operation since its start, if applicable.
>affectedVnfcs	AffectedVnfc	0..N	Information about VNFC instances that were affected during the lifecycle operation. See note 1.
>affectedVirtualLinks	AffectedVirtualLink	0..N	Information about VL instances that were affected during the lifecycle operation. See notes 1 and 3.
>affectedExtLinkPorts	AffectedExtLinkPort	0..N	Information about external VNF link ports that were affected during the lifecycle operation. See note 1.
>affectedVirtualStorages	AffectedVirtualStorage	0..N	Information about virtualised storage instances that were affected during the lifecycle operation. See note 1.
changedInfo	VnfInfoModifications	0..1	Information about the changed VNF instance information, including VNF configurable properties, if applicable. See notes 1 and 2.
affectedVipCps	AffectedVipCp	0..N	Information about virtual IP CP instances that were affected during the execution of the lifecycle management operation.

Attribute name	Data type	Cardinality	Description
changedExtConnectivity	ExtVirtualLinkInfo	0..N	Information about changed external connectivity, if applicable. See note 1.
modificationsTriggeredByVnfPkgChange	ModificationsTriggeredByVnfPkgChange	0..1	Information about performed changes of "VnfInstance" attributes triggered by changing the current VNF package, if applicable. Shall be absent if the "operation" attribute is different from "CHANGE_VNFPKG". See notes 1 and 2.
vnfSnapshotInfoId	Identifier	0..1	Identifier of the "Individual VNF snapshot" resource. Shall be present if applicable to the type of LCM operation, i.e. if the value of the "operation" attribute is either "CREATE_SNAPSHOT" or "REVERT_TO_SNAPSHOT".
lcmCoordinations	Structure (inlined)	0..N	Information about LCM coordination actions (see clause 10 in ETSI GS NFV-SOL 002 [i.2]) related to this LCM operation occurrence.
>id	Identifier	1	Identifier of this coordination action
>coordinationActionName	Identifier	1	Indicator of the actual coordination action.
>coordinationResult	LcmCoordResultType	0..1	The result of executing the coordination action which also implies the action to be performed by the VNFM as the result of this coordination. Shall be present if the coordination has been finished. Shall be absent if the coordination is ongoing or has timed out (see note 4).
>startTime	DateTime	1	The time when the VNFM has received the confirmation that the coordination action has been started.
>endTime	DateTime	0..1	The time when the VNFM has received the confirmation that the coordination action has finished or has been cancelled, or the time when a coordination action has timed out. Shall be present for a coordination action that has finished or timed out (see note 4) and shall be absent if the coordination is ongoing.
>delay	DateTime	0..1	The end of the delay period. This attribute shall be present if the last known HTTP response related to this coordination has contained a "Retry-After" header, and shall be absent otherwise.
rejectedLcmCoordinations	Structure (inlined)	0..N	Information about LCM coordination actions (see clause 10 in ETSI GS NFV-SOL 002 [i.2]) that were rejected by 503 error which means they can be tried again after a delay. See note 5.
>coordinationActionName	Identifier	1	Indicator of the actual coordination action.
>rejectionTime	DateTime	1	The time when the VNFM has received the 503 response that rejects the actual coordination.
>endpointType	Enum (inlined)	1	The endpoint type used by this coordination action. Valid values: • MGMT: coordination with other operation supporting management systems (e.g. EM) • VNF: coordination with the VNF instance
>delay	DateTime	1	The end of the delay period, as calculated from the startTime and "Retry-After" header.
>endpointType	Enum (inlined)	1	The endpoint type used by this coordination action. Valid values: • MGMT: coordination with other operation supporting management systems (e.g. EM) • VNF: coordination with the VNF instance
warnings	String	0..N	Warning messages that were generated while the operation was executing. If the operation has included LCM coordination actions and these have resulted in warnings, such warnings should be added to this attribute.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>vnfInstance	Link	1	Link to the VNF instance that the operation applies to.
>grant	Link	0..1	Link to the grant for this operation, if one exists.

Attribute name	Data type	Cardinality	Description
>cancel	Link	0..1	Link to the task resource that represents the "cancel" operation for this VNF LCM operation occurrence, if cancelling is currently allowed.
>retry	Link	0..1	Link to the task resource that represents the "retry" operation for this VNF LCM operation occurrence, if retrying is currently allowed.
>rollback	Link	0..1	Link to the task resource that represents the "rollback" operation for this VNF LCM operation occurrence, if rolling back is currently allowed.
>fail	Link	0..1	Link to the task resource that represents the "fail" operation for this VNF LCM operation occurrence, if declaring as failed is currently allowed.
>vnfSnapshot	Link	0..1	Link to the VNF snapshot resource, if the VNF LCM operation occurrence is related to a VNF snapshot. Shall be present if operation="CREATE_SNAPSHOT" or operation="REVERT_TO_SNAPSHOT".
NOTE 1: This allows the NFVO to obtain the information contained in the latest "result" notification if it has not received it due to an error or a wrongly configured subscription filter. NOTE 2: Not more than one of changedInfo and modificationsTriggeredByVnfPkgChange shall be present. NOTE 3: For a particular affected VL, there shall be as many "AffectedVirtualLink" entries as needed for signalling the different types of changes, i.e. one per virtual link and change type. For instance, in the case of signaling affected VL instances involving the addition of a particular VL instance with links ports, one "AffectedVirtualLink" entry signals the addition of the VL by using the "changeType" attribute of "AffectedVirtualLink" structure equal to "ADDED", and another "AffectedVirtualLink" entry signals the addition of externally visible VNF link ports of the VL by using the "changeType" equal to "LINK_PORT_ADDED". NOTE 4: A coordination action has timed out if the VNFM has not been able to read the "Individual coordination action" resource within a timeout interval after requesting the coordination to be started or to be cancelled. The length of the timeout interval is defined by means outside the scope of the present document. NOTE 5: The list of rejected coordinations may be garbage collected if the LCM operation occurrence has reached a terminal state, i.e. one of "COMPLETED", "FAILED" and "ROLLED_BACK".			

5.5.2.14 Type: CancelMode

This type represents a parameter to select the mode of cancelling an ongoing VNF LCM operation occurrence. It shall comply with the provisions defined in table 5.5.2.14-1.

Table 5.5.2.14-1: Definition of the CancelMode data type

Attribute name	Data type	Cardinality	Description
cancelMode	CancelModeType	1	Cancellation mode to apply.

5.5.2.15 Type: LccnSubscriptionRequest

This type represents a subscription request related to notifications about VNF lifecycle changes. It shall comply with the provisions defined in table 5.5.2.15-1.

Table 5.5.2.15-1: Definition of the LccnSubscriptionRequest data type

Attribute name	Data type	Cardinality	Description
filter	LifecycleChangeNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this subscription, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [8]. This attribute shall only be present if the subscriber requires authorization of notifications.
verbosity	LcmOpOccNotificationVerbosityType	0..1	This attribute signals the requested verbosity of LCM operation occurrence notifications. If it is not present, it shall default to the value "FULL".

5.5.2.16 Type: LccnSubscription

This type represents a subscription related to notifications about VNF lifecycle changes. It shall comply with the provisions defined in table 5.5.2.16-1.

Table 5.5.2.16-1: Definition of the LccnSubscription data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this subscription resource.
filter	LifecycleChangeNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
verbosity	LcmOpOccNotificationVerbosityType	1	This attribute signals the verbosity of LCM operation occurrence notifications.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.

5.5.2.17 Type: VnfLcmOperationOccurrenceNotification

This type represents a VNF lifecycle management operation occurrence notification, which informs the receiver of changes in the VNF lifecycle caused by a VNF LCM operation occurrence. It shall comply with the provisions defined in table 5.5.2.17-1. The support of the notification is mandatory.

This notification shall be triggered by the VNFM when there is a change in the state of a VNF LCM operation occurrence that changes the VNF lifecycle, which represents an occurrence of one the following LCM operations:

- Instantiation of the VNF
- Scaling of the VNF instance (including auto-scaling)
- Healing of the VNF instance (including auto-healing)
- Change of the state of the VNF instance (i.e. Operate VNF)
- Change of the deployment flavour of the VNF instance
- Change of the external connectivity of the VNF instance
- Change of the current VNF package
- Termination of the VNF instance

- Modification of VNF instance information and/or VNF configurable properties through the "PATCH" method on the "Individual VNF instance" resource
- Creation of a VNF snapshot
- Reversion of the VNF instance to a VNF snapshot

Clause 5.6.2 defines the states and state transition of a VNF LCM operation occurrence, and also specifies details of the notifications to be emitted at each state transition.

If this is the initial notification about the start of a VNF LCM operation occurrence, it is assumed that the notification is sent by the VNFM before any action (including sending the grant request) is taken as part of the LCM operation. Due to possible race conditions, the "start" notification, the grant request and the LCM operation acknowledgment (i.e. the "202 Accepted" response) can arrive in any order at the NFVO, and the NFVO shall be able to handle such a situation.

If this is a notification about a final or intermediate result state of a VNF LCM operation occurrence, the notification shall be sent after all related actions of the LCM operation that led to this state have been executed.

The new state shall be set in the "Individual VNF LCM operation occurrence" resource before the notification about the state change is sent.

The amount of information provided in the LCM operation occurrence notifications to be issued by the VNFM when a particular subscription matches can be controlled by the API consumer using the "verbosity" attribute in the subscription request (see clause 5.5.2.15). The "verbosity" setting in a particular individual subscription shall only apply to the LCM operation occurrence notifications triggered by that subscription. However, it shall not affect the amount of information in the "VnfLcmOpOcc" structure (see clause 5.5.2.13) which represents the "Individual LCM operation occurrence" resource associated with each of the notifications.

See clause 5.6.2.2 for further provisions regarding sending this notification, including in cases of handling LCM operation errors.

Table 5.5.2.17-1: Definition of the VnfLcmOperationOccurrenceNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "VnfLcmOperationOccurrenceNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to. Shall be set to the value of the "id" attribute of the "LccnSubscription" representing the associated "Individual subscription" resource.
timeStamp	DateTime	1	Date-time of the generation of the notification.
notificationStatus	Enum (inlined)	1	Indicates whether this notification reports about the start of a lifecycle operation or the result of a lifecycle operation. Permitted values: - START: Informs about the start of the VNF LCM operation occurrence. - RESULT: Informs about the final or intermediate result of the VNF LCM operation occurrence.
operationState	LcmOperationStateType	1	The state of the VNF LCM operation occurrence.
vnfInstanceId	Identifier	1	The identifier of the VNF instance affected.
operation	LcmOperationType	1	The lifecycle management operation.

Attribute name	Data type	Cardinality	Description
isAutomaticInvocation	Boolean	1	Set to true if this VNF LCM operation occurrence has been triggered by an automated procedure inside the VNFM (i.e. ScaleVnf/ScaleVnfToLevel triggered by auto-scale, or HealVnf triggered by auto-heal). Set to false otherwise.
verbosity	LcmOpOccNotification VerbosityType	0..1	This attribute signals the verbosity of the notification. If it is not present, it shall default to the value "FULL". If the value is "SHORT", full change details can be obtained by performing a GET request on the "Individual LCM operation occurrence" resource that is signalled by the "vnfLcmOpOcc" child attribute of the "_links" attribute.
vnfLcmOpOccId	Identifier	1	The identifier of the VNF lifecycle management operation occurrence associated to the notification. Shall be set to the value of the "id" attribute of the "VnfLcmOpOcc" representing the associated "Individual VNF lifecycle management operation occurrence" resource.
affectedVnfcs	AffectedVnfc	0..N	Information about VNFC instances that were affected during the lifecycle operation. See note 1.
affectedVirtualLinks	AffectedVirtualLink	0..N	Information about VL instances that were affected during the lifecycle operation. See note 1 and note 2.
affectedExtLinkPorts	AffectedExtLinkPort	0..N	Information about external VNF link ports that were affected during the lifecycle operation. See note 1.
affectedVirtualStorages	AffectedVirtualStorage	0..N	Information about virtualised storage instances that were affected during the lifecycle operation. See note 1.
changedInfo	VnfInfoModifications	0..1	Information about the changed VNF instance information, including changed VNF configurable properties. Shall be present if the "notificationStatus" is set to "RESULT", the "operation" attribute is not equal to "CHANGE_VNFPKG", the "verbosity" attribute is set to "FULL" and the operation has performed any changes to VNF instance information, including VNF configurable properties. Shall be absent otherwise. See note 3.
affectedVipCps	AffectedVipCp	0..N	Information about virtual IP CP instances that were affected during the execution of the lifecycle management operation, if this notification represents the result of a lifecycle management operation occurrence. Shall be present if the "notificationStatus" is set to "RESULT", the "verbosity" attribute is set to "FULL" and the operation has made any changes to the VIP CP instances of the VNF instance. Shall be absent otherwise. Only information about VIP CP instances that have been added, deleted or modified shall be provided.
affectedVirtualCps	AffectedVirtualCp	0..N	Information about virtual CP instances that were affected during the execution of the lifecycle management operation, if this notification represents the result of a lifecycle management operation occurrence. Shall be present if the "notificationStatus" is set to "RESULT", the "verbosity" attribute is set to "FULL" and the operation has made any changes to the virtual CP instances of the VNF instance. Shall be absent otherwise. Only information about virtual CP instances that have been added, deleted or modified shall be provided.

Attribute name	Data type	Cardinality	Description
changedExtConnectivity	ExtVirtualLinkInfo	0..N	Information about changed external connectivity, if this notification represents the result of a lifecycle operation occurrence. Shall be present if the "notificationStatus" is set to "RESULT", the "verbosity" attribute is set to "FULL" and the operation has made any changes to the external connectivity of the VNF instance. Shall be absent otherwise. Only information about external VL instances that have been added or modified shall be provided.
modificationsTriggeredByVnfPkgChange	ModificationsTriggeredByVnfPkgChange	0..1	Information about performed changes of "VnfInstance" attributes triggered by changing the current VNF package. Shall be present if the "notificationStatus" is set to "RESULT", the "operation" attribute is equal to " CHANGE_VNFPKG", the "verbosity" attribute is set to "FULL" and the operation has performed any changes to "VnfInstance" attributes. Shall be absent otherwise. See note 3.
error	ProblemDetails	0..1	Details of the latest error, if one has occurred during executing the LCM operation (see clause 6.3 of ETSI GS NFV-SOL 013 [8]). Shall be present if the "operationState" attribute is "FAILED_TEMP", "FAILED" or "ROLLED_BACK" and shall be absent otherwise.
_links	LccnLinks	1	Links to resources related to this notification. The link URIs in this structure shall be set to point to the resources identified by the corresponding identifier attributes in this notification.

NOTE 1: Shall be present if the "notificationStatus" is set to "RESULT", the "verbosity" attribute is set to "FULL" and the operation has performed any resource modification. Shall be absent otherwise. This attribute contains information about the cumulative changes to virtualised resources that were performed so far by the VNF LCM operation occurrence and by any of the error handling procedures for that operation occurrence.

NOTE 2: For a particular affected VL, there shall be as many "AffectedVirtualLink" entries as needed for signalling the different types of changes, i.e. one per virtual link and change type. For instance, in the case of signaling affected VL instances involving the addition of a particular VL instance with links ports, one "AffectedVirtualLink" entry signals the addition of the VL by using the "changeType" attribute of "AffectedVirtualLink" structure equal to "ADDED", and another "AffectedVirtualLink" entry signals the addition of externally visible VNF link ports of the VL by using the "changeType" equal to "LINK_PORT_ADDED".

NOTE 3: Not more than one of changedInfo and modificationsTriggeredByVnfPkgChange shall be present.

5.5.2.18 Type: VnfIdentifierCreationNotification

This type represents a VNF identifier creation notification, which informs the receiver of the creation of a new "Individual VNF instance" resource and the associated VNF instance identifier. It shall comply with the provisions defined in table 5.5.2.18-1. The support of the notification is mandatory.

This notification shall be triggered by the VNFM when it has created an "Individual VNF instance" resource and the associated VNF instance identifier.

Table 5.5.2.18-1: Definition of the VnfIdentifierCreationNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "VnfIdentifierCreationNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
vnfInstanceId	Identifier	1	The created VNF instance identifier.
_links	LccnLinks	1	Links to resources related to this notification.

5.5.2.19 Type: VnfIdentifierDeletionNotification

This type represents a VNF identifier deletion notification, which informs the receiver of the deletion of a new "Individual VNF instance" resource and the associated VNF instance identifier. It shall comply with the provisions defined in table 5.5.2.19-1. The support of the notification is mandatory.

This notification shall be triggered by the VNFM when it has deleted an "Individual VNF instance" resource and the associated VNF instance identifier.

Table 5.5.2.19-1: Definition of the VnfIdentifierDeletionNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "VnfIdentifierDeletionNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
vnfInstanceId	Identifier	1	The deleted VNF instance identifier.
_links	LccnLinks	1	Links to resources related to this notification.

5.5.2.20 Type: CreateVnfSnapshotInfoRequest

This type represents request parameters for the creation of an "Individual VNF snapshot" resource which can be populated with content obtained by invoking the "Create VNF snapshot" LCM operation or extracted from a VNF snapshot package. It shall comply with the provisions defined in table 5.5.2.20-1.

Table 5.5.2.20-1: Definition of the CreateVnfSnapshotInfoRequest data type

Attribute name	Data type	Cardinality	Description
vnfSnapshotPkgId	Identifier	0..1	Identifier of the VNF snapshot package information held by the NFVO. See note.
vnfSnapshot	VnfSnapshot	0..1	Information about the VNF snapshot, content and/or reference to its content.
NOTE: The present attribute shall be provided if the "Individual VNF snapshot" resource is requested to be created as part of a VNF snapshot package extraction.			

5.5.2.21 Type: CreateVnfSnapshotRequest

This type represents request parameters for the "Create VNF Snapshot" LCM operation which takes a snapshot of a VNF instance and populates a previously-created "Individual VNF snapshot" resource with the content of the snapshot. It shall comply with the provisions defined in table 5.5.2.21-1.

Table 5.5.2.21-1: Definition of the CreateVnfSnapshotRequest data type

Attribute name	Data type	Cardinality	Description
vnfSnapshotInfoId	Identifier	1	Identifier of the "Individual VNF snapshot" resource to which the VNF snapshot is to be associated.
additionalParams	KeyValuePairs	0..1	Additional input parameters for the snapshot creation process, specific for the VNF being "snapshotted", as declared in the VNFD as part of "CreateSnapshotVnfOpConfig" defined in ETSI GS NFV-IFA 011 [10].
userDefinedData	KeyValuePairs	0..1	User defined data for the VNF snapshot.

5.5.2.22 Type: VnfSnapshotInfo

This type represents an "Individual VNF snapshot" resource. It shall comply with the provisions defined in table 5.5.2.22-1.

Table 5.5.2.22-1: Definition of the VnfSnapshotInfo data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the "Individual VNF snapshot" resource. This identifier is allocated by the VNFM.
vnfSnapshotPkgId	Identifier	0..1	Identifier of the VNF snapshot package information held by the NFVO. Shall be present when the "Individual VNF snapshot" resource is created from a VNF snapshot package extraction.
vnfSnapshot	VnfSnapshot	0..1	Information about the VNF snapshot, content and/or references to its content. Shall be present when the "Individual VNF snapshot" resource is associated to a VNF snapshot created via the corresponding "Create VNF Snapshot" task resource or extracted from a VNF snapshot package.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>takenFrom	Link	0..1	Link to the VNF instance from which this snapshot was taken. Shall be present when the "Individual VNF snapshot" resource is associated to a VNF snapshot created via the corresponding "Create VNF snapshot" task resource.

5.5.2.23 Type: VnfSnapshot

This type represents a VNF snapshot. It shall comply with the provisions defined in table 5.5.2.23-1.

Table 5.5.2.23-1: Definition of the VnfSnapshot data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the VNF snapshot. This identifier is allocated by the VNFM.
vnfInstanceId	Identifier	1	Identifier of the snapshotted VNF instance.
creationStartedAt	DateTime	1	Timestamp indicating when the VNF Snapshot creation has been started by the VNFM.
creationFinishedAt	DateTime	0..1	Timestamp indicating when the VNF snapshot has been completed by the VNFM. Shall be present once the VNF Snapshot creation has been completed.

Attribute name	Data type	Cardinality	Description
vnfdId	Identifier	1	Identifier of the VNFD in use at the time the snapshot of the VNF instance has been created.
vnfInstance	VnfInstance	1	VNF instance information of the snapshotted VNF instance. This is a copy of the "Individual VNF instance" resource.
vnfcSnapshots	VnfcSnapshotInfo	1..N	Information about VNFC snapshots constituting this VNF snapshot.
vnfStateSnapshotInfo	VnfStateSnapshotInfo	0..1	Information about VNF-specific state snapshot data. This attribute shall not be present before the VNF snapshot has been completed. Otherwise, this attribute shall be present if the VNF snapshot has associated additional VNF-specific state data.
userDefinedData	KeyValuePairs	0..1	User defined data for the VNF snapshot.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>vnfStateSnapshot	Link	0..1	Link to the "VNF state snapshot" resource. This attribute shall not be present before the VNF snapshot has been completed. Otherwise, this attribute shall be present if the VNF snapshot has associated additional VNF-specific state data.

5.5.2.24 Type: VnfSnapshotInfoModificationRequest

This type represents attribute modifications for an "Individual VNF snapshot" resource, i.e. modifications to a resource representation based on the "VnfSnapshotInfo" data type. The attributes of "VnfSnapshotInfo" that can be modified according to the provisions in clause 5.5.2.22 are included in the "VnfSnapshotInfoModificationRequest" data type.

The "VnfSnapshotInfoModificationRequest" data type shall comply with the provisions defined in table 5.5.2.24-1.

Table 5.5.2.24-1: Definition of the VnfSnapshotInfoModificationRequest data type

Attribute name	Data type	Cardinality	Description
vnfSnapshotPkgId	Identifier	0..1	New value of the "vnfSnapshotPkgId" attribute in "VnfSnapshotInfo". The value "null" is not permitted.
vnfSnapshot	VnfSnapshot	0..1	New value of the "vnfSnapshot" attribute in "VnfSnapshotInfo". The value "null" is not permitted.

5.5.2.25 Type: VnfSnapshotInfoModifications

This type represents attribute modifications that were performed on an "Individual VNF snapshot" resource. The attributes that can be included consist of those requested to be modified explicitly in the "VnfSnapshotInfoModificationRequest" data structure, and additional attributes of the "VnfSnapshotInfo" data structure that were modified implicitly.

The "VnfSnapshotInfoModifications" data type shall comply with the provisions defined in table 5.5.2.25-1.

Table 5.5.2.25-1: Definition of the VnfSnapshotInfoModifications data type

Attribute name	Data type	Cardinality	Description
vnfSnapshotPkgId	Identifier	0..1	If present, this attribute signals modifications of the "vnfSnapshotPkgId" attribute in "VnfSnapshotInfo" as defined in clause 5.5.2.22.
vnfSnapshot	VnfSnapshot	0..1	If present, this attribute signals modifications of the "vnfSnapshot" attribute in "VnfSnapshotInfo" as defined in clause 5.5.2.22.

5.5.2.26 Type: RevertToVnfSnapshotRequest

This type represents request parameters for the "Revert to VNF Snapshot" operation. It shall comply with the provisions defined in table 5.5.2.26-1.

Table 5.5.2.26-1: Definition of the RevertToVnfSnapshotRequest data type

Attribute name	Data type	Cardinality	Description
vnfSnapshotInfoId	Identifier	1	Identifier of the "Individual VNF snapshot" resource with the information of the VNF snapshot to be reverted to.
additionalParams	KeyValuePairs	0..1	Additional input parameters for the revert to VNF snapshot process, specific for the VNF being "reverted", as declared in the VNFD as part of "RevertToSnapshotVnfOpConfig" defined in ETSI GS NFV-IFA 011 [10].

5.5.3 Referenced structured data types

5.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but can neither be resource representations nor bound to any subscribe/notify mechanism.

5.5.3.2 Type: ExtVirtualLinkInfo

This type represents information about an external VL. It shall comply with the provisions defined in table 5.5.3.2-1.

Table 5.5.3.2-1: Definition of the ExtVirtualLinkInfo data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the external VL and the related external VL information instance. The identifier is assigned by the NFV-MANO entity that manages this VL instance.
resourceHandle	ResourceHandle	1	Reference to the resource realizing this VL.
extLinkPorts	ExtLinkPortInfo	0..N	Link ports of this VL.
currentVnfExtCpData	VnfExtCpData	1..N	Allows the API consumer to read the current CP configuration information for the connection of external CPs to the external virtual link. See note.
extNetAttDefResource	NetAttDefResourceInfo	0..N	Network attachment definition resources that provide the specification of the interface to attach connection points to this VL.
NOTE: This attribute reflects the current configuration information that has resulted from merging into this attribute the "VnfExtCpData" information which was passed as part of the "ExtVirtualLinkData" structure in the input of the most recent VNF LCM operation such as "InstantiateVnfRequest", "ChangeExtVnfConnectivityRequest", "ChangeVnfFlavourRequest" or "ChangeCurrentVnfPkgRequest", or in the Grant response. If applying such change results in an empty list of "currentVnfExtCpData" structure instances, the affected instance of "ExtVirtualLinkInfo" shall be removed from its parent data structure.			

5.5.3.3 Type: ExtManagedVirtualLinkInfo

This type provides information about an externally-managed virtual link. It shall comply with the provisions defined in table 5.5.3.3-1.

Table 5.5.3.3-1: Definition of the ExtManagedVirtualLinkInfo data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the externally-managed internal VL and the related externally-managed VL information instance. The identifier is assigned by the NFV-MANO entity that manages this VL instance.
vnfVirtualLinkDescId	IdentifierInVnfd	1	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case the value differs from the vnfdId attribute of the VnfInstance (e.g. during a "Change current VNF package" operation or due to its final failure).
networkResource	ResourceHandle	1	Reference to the VirtualNetwork resource providing this VL.
vnfLinkPorts	VnfLinkPortInfo	0..N	Link ports of this VL.
vnfNetAttDefResource	NetAttDefResourceInfo	0..N	Network attachment definition resources that provide the specification of the interface to attach connection points to this VL.
extManagedMultisiteVirtualLinkId	Identifier	0..1	Identifier of the externally-managed multi-site VL instance. The identifier is assigned by the NFV-MANO entity that manages the externally managed multi-site VL instance. It shall be present when the externally-managed internal VL is part of a multi-site VL, e.g. in support of multi-site VNF spanning several VIMs. All externally-managed internal VL instances corresponding to an internal VL created based on the same virtualLinkDescId shall refer to the same extManagedMultisiteVirtualLinkId.

5.5.3.4 Void

5.5.3.5 Type: VnfcResourceInfo

This type represents the information on virtualised compute and storage resources used by a VNFC in a VNF instance.

Depending on the form of virtualisation container of the VNFC:

- for a VNFC based on VM, a reference to the corresponding VirtualCompute shall be provided; and
- for a VNFC based on OS container(s), a reference to the Compute MCIO shall be provided. Hence, exposure of information by the VNFM to the NFVO is at the MCIO level.

In addition, the references to the storage resources depend on the form of the VNFC:

a) for a VNFC based on VM, storage resource identifiers shall refer to VirtualStorage resources; and

b) for a VNFC based on OS container(s), storage resource identifiers shall refer to Storage MCIOs.

It shall comply with the provisions defined in table 5.5.3.5-1.

Table 5.5.3.5-1: Definition of the VnfcResourceInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of this VnfcResourceInfo instance.
vduId	IdentifierInVnfd	1	Reference to the applicable VDU in the VNFD. See note 1.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case the value differs from the vnfdId attribute of the VnfInstance (e.g. during a "Change current VNF package" operation or due to its final failure). See note 4.
computeResource	ResourceHandle	1	Reference to the VirtualCompute resource or reference to a Compute MCIO.

Attribute name	Data type	Cardinality	Description
zoneId	Identifier	0..1	The identifier of the resource zone, as managed by the resource management layer (typically, the VIM), where the referenced VirtualCompute resource is placed. Shall be provided if this information is available from the VIM.
storageResourceIds	IdentifierInVnf	0..N	References to the VirtualStorage resources or references to Storage MCIOs. The value refers to a VirtualStorageResourceInfo item in the VnfInstance.
reservationId	Identifier	0..1	The reservation identifier applicable to the resource. It shall be present when an applicable reservation exists.
vnfcCpInfo	Structure (inlined)	0..N	CPs of the VNFC instance. Shall be present when that particular CP of the VNFC instance is exposed as an external CP of the VNF instance or is connected to an external CP of the VNF instance. See note 2. May be present otherwise.
>id	IdentifierInVnf	1	Identifier of this VNFC CP instance and the associated array entry.
>cpdId	IdentifierInVnfd	1	Identifier of the VDU CPD, cpdId, in the VNFD. See note 1.
>vnfExtCpId	IdentifierInVnf	0..1	Identifier of the related external CP. Shall be present when the VNFC CP is exposed as an external CP of the VNF instance or connected to an external CP of the VNF instance (see note 2) and shall be absent otherwise.
>cpProtocolInfo	CpProtocolInfo	0..N	Network protocol information for this CP. May be omitted if the VNFC CP is exposed as an external CP. See note 3.
>vnfLinkPortId	IdentifierInVnf	0..1	Identifier of the "VnfLinkPortInfo" structure in the "VnfVirtualLinkResourceInfo" or "ExtManagedVirtualLinkInfo" structure. Shall be present if the CP is associated to a link port on an internal VL (including externally-managed internal VL) of the VNF instance and shall be absent otherwise.
>parentCpId	IdentifierInVnf	0..1	Identifier of another VNFC CP instance that corresponds to the parent port of a trunk that the present VNFC CP instance participates in. Shall be provided if the present CP instance participates in a trunk as subport, and the referred VNFC CP instances are also present in the vnfcCpInfo attribute.
>netAttDefResourceId	Identifier	0..N	Identifier of the "NetAttDefResourceInfo" structure that provides the specification of the interface to attach the connection point to a secondary container cluster network. See notes 5 and 6. It shall be present if the internal CP is associated to a VNFC realized by one or a set of OS containers and is connected to a secondary container cluster network. It shall not be present otherwise.
>metadata	KeyValuePairs	0..1	Metadata about this CP.
metadata	KeyValuePairs	0..1	Metadata about this resource.

NOTE 1: ETSI GS NFV-SOL 001 [i.4] specifies the structure and format of the VNFD based on TOSCA specifications.
NOTE 2: A VNFC CP is "connected to" an external CP if the VNFC CP is connected to an internal VL that exposes an external CP. A VNFC CP is "exposed as" an external CP if it is connected directly to an external VL.
NOTE 3: The information can be omitted because it is already available as part of the external CP information.
NOTE 4: If only the value or the presence of this attribute is changed in the "VnfcResourceInfo" structure by an LCM operation occurrence, this does not represent a change that requires including a related "AffectedVnfc" structure in the VNF LCM operation occurrence notifications or the "VnfLcmOpOcc" structure related to this LCM operation occurrence.
NOTE 5: Cardinality greater than 1 is only applicable for specific cases where more than one network attachment definition resource is needed to fulfil the connectivity requirements of the internal CP, e.g. to build a link redundant mated pair in SR-IOV cases.
NOTE 6: When more than one netAttDefResourceId is indicated, all shall belong to the same namespace.

5.5.3.6 Type: VnfVirtualLinkResourceInfo

This type represents the information that allows addressing a virtualised resource that is used by an internal VL instance in a VNF instance. It shall comply with the provisions defined in table 5.5.3.6-1.

Table 5.5.3.6-1: Definition of the VnfVirtualLinkResourceInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of this VnfVirtualLinkResourceInfo instance.
vnfVirtualLinkDescId	IdentifierInVnfd	1	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case the value differs from the vnfdId attribute of the VnfInstance (e.g. during a "Change current VNF package" operation or due to its final failure). See note.
networkResource	ResourceHandle	1	Reference to the VirtualNetwork resource or reference to a Network MCIO.
zoneId	Identifier	0..1	The identifier of the resource zone, as managed by the resource management layer (typically, the VIM), where the referenced VirtualNetwork resource is placed. Shall be provided if this information is available from the VIM.
reservationId	Identifier	0..1	The reservation identifier applicable to the resource. It shall be present when an applicable reservation exists.
vnfLinkPorts	VnfLinkPortInfo	0..N	Links ports of this VL. Shall be present when the linkPort is used for external connectivity by the VNF (refer to VnfLinkPortInfo). May be present otherwise.
metadata	KeyValuePairs	0..1	Metadata about this resource.
NOTE: If only the value or the presence of this attribute is changed in the "VnfVirtualLinkResourceInfo" structure by an LCM operation occurrence, this does not represent a change that requires including a related "AffectedVirtualLink" structure in the VNF LCM operation occurrence notifications or the "VnfLcmOpOcc" structure related to this LCM operation occurrence.			

5.5.3.7 Type: VirtualStorageResourceInfo

This type represents the information that allows addressing a virtualised resource that is used by a VNF instance. It shall comply with the provisions defined in table 5.5.3.7-1.

Table 5.5.3.7-1: Definition of the VirtualStorageResourceInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of this VirtualStorageResourceInfo instance.
virtualStorageDescId	IdentifierInVnfd	1	Identifier of the VirtualStorageDesc in the VNFD.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case the value differs from the vnfdId attribute of the VnfInstance (e.g. during a "Change current VNF package" operation or due to its final failure). See note.
storageResource	ResourceHandle	1	Reference to the VirtualStorage resource or reference to a Storage MCIO.
zoneId	Identifier	0..1	The identifier of the resource zone, as managed by the resource management layer (typically, the VIM), where the referenced VirtualStorage resource is placed. Shall be provided if this information is available from the VIM.
reservationId	Identifier	0..1	The reservation identifier applicable to the resource. It shall be present when an applicable reservation exists.
metadata	KeyValuePairs	0..1	Metadata about this resource.
NOTE: If only the value or the presence of this attribute is changed in the "VirtualStorageResourceInfo" structure by an LCM operation occurrence, this does not represent a change that requires including a related "AffectedVirtualStorage" structure in the VNF LCM operation occurrence notifications or the "VnfLcmOpOcc" structure related to this LCM operation occurrence.			

5.5.3.8 Type: VnfLinkPortInfo

This type represents a link port of an internal VL of a VNF. It shall comply with the provisions defined in table 5.5.3.8-1.

Table 5.5.3.8-1: Definition of the VnfLinkPortInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of this link port as provided by the entity that has created the link port.
resourceHandle	ResourceHandle	1	Reference to the virtualised resource realizing this link port.
cpInstanceId	IdentifierInVnf	0..1	When the link port is used for external connectivity by the VNF, this attribute represents the identifier of the external CP associated with this link port. When the link port is used for internal connectivity in the VNF, this attribute represents the identifier of the VNFC CP to be connected to this link port. Shall be present when the link port is used for external connectivity by the VNF. May be present if used to reference a VNFC CP instance. There shall be at most one link port associated with any external connection point instance or internal connection point (i.e. VNFC CP) instance. The value refers to an "extCpInfo" item in the VnfInstance or a "vnfcCpInfo" item of a "vnfcResourceInfo" item in the VnfInstance. See note 1.
cpInstanceType	Enum (inlined)	0..1	Type of the CP instance that is identified by cpInstanceId. Shall be present if "cpInstanceId" is present and shall be absent otherwise. Permitted values: - VNFC_CP: The link port is connected to a VNFC CP. - EXT_CP: The link port is associated to an external CP. See note 1.
vipCpInstanceId	IdentifierInVnf	0..1	VIP CP instance of the VNF connected to this link port. May be present. See notes 1 and 2.
trunkResourceId	IdentifierInVim	0..1	Identifier of the trunk resource in the VIM. Shall be present if the present link port corresponds to the parent port that the trunk resource is associated with. See note 3.

NOTE 1: Either cpInstanceId with cpInstanceType set to "EXT_CP" or any combination of cpInstanceId with cpInstanceType set to "VNFC_CP" and vipCpInstanceId (i.e. one or both of them) shall be present for a VnfLinkPortInfo. In case both cpInstanceId with cpInstanceType set to "VNFC_CP" and vipCpInstanceId are present, the two different CP instances share the linkport.

NOTE 2: Clause A.4 of ETSI GS NFV-IFA 007 [1] provides examples for configurations where both vipCpInstanceId and vnfcCpInstanceId are present (UC#5 and UC#5-b), only vnfcCpInstanceId is present (UC#2), or only vipCpInstanceId is present (UC6 and UC#6-b).

NOTE 3: The value of "trunkResourceId" is scoped by the value of "vimConnectionId" in the "resourceHandle" attribute.

5.5.3.9 Type: ExtLinkPortInfo

This type represents information about a link port of an external VL, i.e. a port providing connectivity for the VNF to an NS VL. It shall comply with the provisions defined in table 5.5.3.9-1.

Table 5.5.3.9-1: Definition of the ExtLinkPortInfo data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this link port as provided by the entity that has created the link port.
resourceHandle	ResourceHandle	1	Reference to the virtualised resource realizing this link port.
cpInstanceId	IdentifierInVnf	0..1	Identifier of the external CP of the VNF connected to this link port. There shall be at most one link port associated with any external connection point instance. The value refers to an "extCpInfo" item in the VnfInstance.
secondaryCpInstanceId	IdentifierInVnf	0..1	Additional external CP of the VNF connected to this link port. If present, this attribute shall refer to a "secondary" ExtCpInfo item in the VNF instance that exposes a virtual IP CP instance which shares this linkport with the external CP instance referenced by the "cpInstanceId" attribute. See note 1.
trunkResourceId	IdentifierInVim	0..1	Identifier of the trunk resource in the VIM. Shall be present if the present link port corresponds to the parent port that the trunk resource is associated with. See note 2.
NOTE 1: The use cases UC#4 and UC#5 in clause A.4 of ETSI GS NFV-IFA 007 [1] provide examples for such a configuration. NOTE 2: The value of "trunkResourceId" is scoped by the value of "vimConnectionId" in the "resourceHandle" attribute.			

5.5.3.9a Void

5.5.3.9b Type: CpProtocolInfo

This type describes the protocol layer(s) that a CP uses together with protocol-related information, like addresses. It shall comply with the provisions defined in table 5.5.3.9b-1.

Table 5.5.3.9b-1: Definition of the CpProtocolInfo data type

Attribute name	Data type	Cardinality	Description
layerProtocol	Enum (inlined)	1	The identifier of layer(s) and protocol(s) associated to the network address information. Permitted values: 1. IP_OVER_ETHERNET 2. IP_FOR_VIRTUAL_CP See note.
ipOverEthernet	IpOverEthernetAddressInfo	0..1	IP addresses over Ethernet to assign to the extCP instance. Shall be present if layerProtocol is equal to "IP_OVER_ETHERNET" and shall be absent otherwise.
virtualCpAddress	VirtualCpAddressInfo	0..1	IP address data assigned to an external CP instance exposing a virtual CP. It shall be present if layerProtocol is equal to "IP_FOR_VIRTUAL_CP" and the external CP instance exposes a virtual CP and shall not be present otherwise.
NOTE: This attribute allows to signal the addition of further types of layer and protocol in future versions of the present document in a backwards-compatible way. In the current version of the present document, only IP over Ethernet is supported.			

5.5.3.10 Type: IpOverEthernetAddressInfo

This type represents information about a network address that has been assigned. It shall comply with the provisions defined in table 5.5.3.10-1.

Table 5.5.3.10-1: Definition of the IpOverEthernetAddressInfo data type

Attribute name	Data type	Cardinality	Description
macAddress	MacAddress	0..1	MAC address, if assigned. See note 1.
segmentationId	String	0..1	Identification of the network segment to which the CP instance connects to. See notes 3 and 4.
ipAddresses	Structure (inlined)	0..N	Addresses assigned to the CP instance. Each entry represents IP addresses assigned by fixed or dynamic IP address assignment per subnet. See note 1.
>type	Enum (inlined)	1	The type of the IP addresses. Permitted values: IPV4, IPV6.
>addresses	IpAddress	0..N	Fixed addresses assigned (from the subnet defined by "subnetId" if provided). See note 2.
>isDynamic	Boolean	0..1	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.
>addressRange	Structure (inlined)	0..1	An IP address range used, e.g. in case of egress connections. See note 2.
>>minAddress	IpAddress	1	Lowest IP address belonging to the range.
>>maxAddress	IpAddress	1	Highest IP address belonging to the range.

Attribute name	Data type	Cardinality	Description
>subnetId	IdentifierInVim	0..1	Subnet defined by the identifier of the subnet resource in the VIM. In case this attribute is present, IP addresses are bound to that subnet.
NOTE 1: At least one of "macAddress" or "ipAddresses" shall be present. NOTE 2: Exactly one of "addresses" or "addressRange" shall be present. NOTE 3: If the CP instance represents a subport in a trunk, segmentationId shall be present. Otherwise it shall not be present. NOTE 4: Depending on the NFVI networking infrastructure, the segmentationId may indicate the actual network segment value (e.g. vlan Id, Vxlan segmentation id, etc.) used in the transport header of the packets or it may be an identifier used between the application and the NFVI networking infrastructure to identify the network sub-interface of the trunk port in question. In the latter case the NFVI infrastructure will map this local segmentationId to whatever segmentationId is actually used by the NFVI's transport technology.			

5.5.3.10a Type: VirtualCpAddressInfo

This type represents information about a network address that has been assigned to a virtual CP. It shall comply with the provisions defined in table 5.5.3.10a-1.

Table 5.5.3.10a-1: Definition of the VirtualCpAddressInfo data type

Attribute name	Data type	Cardinality	Description
type	Enum (inlined)	1	The type of the IP addresses. Permitted values: IPV4, IPV6.
loadBalancerIp	IpAddress	0..1	Fixed addresses assigned to an external load balancer.

5.5.3.11 Type: MonitoringParameter

This type represents a monitoring parameter that is tracked by the VNFM, e.g. for auto-scaling purposes. It shall comply with the provisions defined in table 5.5.3.11-1.

Valid monitoring parameters of a VNF are defined in the VNFD.

NOTE: ETSI GS NFV-SOL 001 [i.4] specifies the structure and format of the VNFD based on TOSCA specifications.

Table 5.5.3.11-1: Definition of the MonitoringParameter data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnfd	1	Identifier of the monitoring parameter defined in the VNFD.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case the value differs from the vnfdId attribute of the VnfInstance (e.g. during a "Change current VNF package" operation or due to its final failure).
name	String	0..1	Human readable name of the monitoring parameter, as defined in the VNFD.
performanceMetric	String	1	Performance metric that is monitored. This attribute shall contain the related "Measurement Name" value as defined in clause 7.2 of ETSI GS NFV-IFA 027 [6].

5.5.3.12 Type: LifecycleChangeNotificationsFilter

This type represents a subscription filter related to notifications about VNF lifecycle changes. It shall comply with the provisions defined in table 5.5.3.12-1.

At a particular nesting level in the filter structure, the following applies: All attributes shall match in order for the filter to match (logical "and" between different filter attributes). If an attribute is an array, the attribute shall match if at least one of the values in the array matches (logical "or" between the values of one filter attribute).

Table 5.5.3.12-1: Definition of the LifecycleChangeNotificationsFilter data type

Attribute name	Data type	Cardinality	Description
vnfInstanceSubscriptionFilter	VnfInstanceSubscriptionFilter	0..1	Filter criteria to select VNF instances about which to notify.
notificationTypes	Enum (inlined)	0..N	Match particular notification types. Permitted values: - VnfLcmOperationOccurrenceNotification - VnfIdentifierCreationNotification - VnfIdentifierDeletionNotification See note.
operationTypes	LcmOperationType	0..N	Match particular VNF lifecycle operation types for the notification of type VnfLcmOperationOccurrenceNotification. May be present if the "notificationTypes" attribute contains the value "VnfLcmOperationOccurrenceNotification" and shall be absent otherwise.
operationStates	LcmOperationStateType	0..N	Match particular LCM operation state values as reported in notifications of type VnfLcmOperationOccurrenceNotification. May be present if the "notificationTypes" attribute contains the value "VnfLcmOperationOccurrenceNotification" and shall be absent otherwise.
NOTE: The permitted values of the "notificationTypes" attribute are spelled exactly as the names of the notification types to facilitate automated code generation systems.			

5.5.3.13 Type: AffectedVnfc

This type provides information about added, deleted, modified and temporary VNFCs. It shall comply with the provisions in table 5.5.3.13-1.

Table 5.5.3.13-1: Definition of the AffectedVnfc data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of the Vnfc instance, identifying the applicable "vnfcResourceInfo" entry in the "VnfInstance" data type (see clause 5.5.2.2).
vduId	IdentifierInVnfd	1	Identifier of the related VDU in the VNFD.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case of a "change current VNF Package" to identify whether the affected VNFC instance is associated to a VDU which is referred from the source or destination VNFD.

Attribute name	Data type	Cardinality	Description
changeType	Enum (inlined)	1	Signals the type of change. Permitted values: - ADDED - REMOVED - MODIFIED - TEMPORARY For a temporary resource, an AffectedVnfc structure exists as long as the temporary resource exists.
computeResource	ResourceHandle	1	Reference to the VirtualCompute resource. Detailed information is (for new and modified resources) or has been (for removed resources) available from the VIM.
resourceDefinitionId	IdentifierLocal	0..1	The identifier of the "ResourceDefinition" in the granting exchange related to the LCM operation occurrence. It shall be present when an applicable GrantInfo for thegranted resource exists. See note.
zoneId	Identifier	0..1	The identifier of the resource zone, as managed by the resource management layer (typically, the VIM), where the referenced VirtualCompute resource is placed. Shall be provided if this information is available from the VIM.
metadata	KeyValuePairs	0..1	Metadata about this resource. The content of this attribute shall be a copy of the content of the "metadata" attribute of the VnfcResourceInfo structure.
affectedVnfcCpIds	IdentifierInVnf	0..N	Identifiers of CP(s) of the VNFC instance that were affected by the change. Shall be present for those affected CPs of the VNFC instance that are associated to an external CP of the VNF instance. May be present for further affected CPs of the VNFC instance.
addedStorageResourceIds	IdentifierInVnf	0..N	References to VirtualStorage resources that have been added. Each value refers to a VirtualStorageResourceInfo item in the VnfInstance that was added to the VNFC. It shall be provided if at least one storage resource was added to the VNFC.
removedStorageResourceIds	IdentifierInVnf	0..N	References to VirtualStorage resources that have been removed. The value contains the identifier of a VirtualStorageResourceInfo item that has been removed from the VNFC and might no longer exist in the VnfInstance. It shall be provided if at least one storage resource was removed from the VNFC.
NOTE: The "resourceDefinitionId" attribute provides information to the API consumer (i.e. the NFVO) to assist in correlating the resource changes performed during the LCM operation with the granted resources in a specific Grant exchange, which is identified by the "grantId" available in the "Individual VNF lifecycle management operation occurrence" and the "id" in the "Individual Grant".			

5.5.3.14 Type: AffectedVirtualLink

This type provides information about added, deleted, modified and temporary VLs, and added or removed VNF link ports. It shall comply with the provisions in table 5.5.3.14-1.

Table 5.5.3.14-1: Definition of the AffectedVirtualLink data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of the virtual link instance, identifying the applicable "vnfVirtualLinkResourceInfo" or "extManagedVirtualLinkInfo" entry in the "VnfInstance" data type (see clause 5.5.2.2).
vnfVirtualLinkDescId	IdentifierInVnfd	1	Identifier of the related VLD in the VNFD.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case of a "change current VNF Package" to identify whether the affected VL instance is associated to a VLD which is referred from the source or destination VNFD.
changeType	Enum (inlined)	1	Signals the type of change. Permitted values: - ADDED - REMOVED - MODIFIED - TEMPORARY - LINK_PORT_ADDED - LINK_PORT_REMOVED For a temporary resource, an AffectedVirtualLink structure exists as long as the temporary resource exists. See note 1.
networkResource	ResourceHandle	1	Reference to the VirtualNetwork resource. Detailed information is (for new and modified resources) or has been (for removed resources) available from the VIM. See note 1.
vnfLinkPortIds	IdentifierInVnf	0..N	Identifiers of the link ports of the affected VL related to the change. Each identifier references a "VnfLinkPortInfo" structure. Shall be set when changeType is equal to "LINK_PORT_ADDED" or "LINK_PORT_REMOVED", and the related "VnfLinkPortInfo" structures are present (case "added") or have been present (case "removed") in the "VnfVirtualLinkResourceInfo" or "ExtManagedVirtualLinkInfo" structures that are represented by the "vnfVirtualLinkResourceInfo" or "extManagedVirtualLinkInfo" attribute in the "VnfInstance" structure. See note 1.
resourceDefinitionId	IdentifierLocal	0..1	The identifier of the "ResourceDefinition" in the granting exchange related to the LCM operation occurrence. It shall be present when an applicable GrantInfo for the granted resource exists. See note 1 and note 2.
zoneId	Identifier	0..1	The identifier of the resource zone, as managed by the resource management layer (typically, the VIM), where the referenced VirtualNetwork resource is placed. Shall be provided if this information is available from the VIM.

Attribute name	Data type	Cardinality	Description
metadata	KeyValuePairs	0..1	Metadata about this resource. The content of this attribute shall be a copy of the content of the "metadata" attribute of the applicable "vnfVirtualLinkResourceInfo" structure if such structure is referenced by the "id" attribute and it has metadata.
NOTE 1: When signalling the addition (LINK_PORT_ADDED) or removal (LINK_PORT_REMOVED) of VNF link ports, the "networkResource" and "resourceDefinitionId" attributes refer to the affected virtual link instance, not the link port instance. The resource handles of the affected VNF link ports can be found by dereferencing the identifiers in the "vnfLinkPortIds" attribute. NOTE 2: The "resourceDefinitionId" attribute provides information to the API consumer (i.e. the NFVO) to assist in correlating the resource changes performed during the LCM operation with the granted resources in a specific Grant exchange, which is identified by the "grantId" available in the "Individual VNF lifecycle management operation occurrence" and the "id" in the "Individual Grant".			

5.5.3.14a Type: AffectedExtLinkPort

This type provides information about added and deleted external link ports (link ports attached to external virtual links). It shall comply with the provisions in table 5.5.3.14a-1.

Table 5.5.3.14a-1: Definition of the AffectedExtLinkPort data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of the link port, identifying the applicable "extLinkPorts" entry in the "ExtVirtualLinkInfo" data type (see clause 5.5.3.2).
changeType	Enum (inlined)	1	Signals the type of change. Permitted values: - ADDED - MODIFIED - REMOVED
extCpInstanceId	IdentifierInVnf	1	Identifier of the related external CP instance.
resourceHandle	ResourceHandle	1	Reference to the link port resource. Detailed information is (for added resources) or has been (for removed resources) available from the VIM.
resourceDefinitionId	IdentifierLocal	0..1	The identifier of the "ResourceDefinition" in the granting exchange related to the LCM operation occurrence. It shall be present when an applicable GrantInfo for the granted resource exists. See note.
NOTE: The "resourceDefinitionId" attribute provides information to the API consumer (i.e. the NFVO) to assist in correlating the resource changes performed during the LCM operation with the granted resources in a specific Grant exchange, which is identified by the "grantId" available in the "Individual VNF lifecycle management operation occurrence" and the "id" in the "Individual Grant".			

5.5.3.14b Type: AffectedVipCp

This type provides information about added, deleted and modified virtual IP CP instances. It shall comply with the provisions in table 5.5.3.14b-1.

Table 5.5.3.14b-1: Definition of the AffectedVipCp data type

Attribute name	Data type	Cardinality	Description
cpInstanceId	IdentifierInVnf	1	Identifier of the virtual IP CP instance and the related "VipCpInfo" structure in "VnfInstance".
cpdId	IdentifierInVnfd	1	Identifier of the VipCpd in the VNFD.
vnfdId	Identifier	0..1	Reference to the VNFD. Shall be present in case of a "change current VNF Package" to identify whether the affected virtual CP instance is associated to a VipCpd which is referred from the source or destination VNFD.
changeType	Enum	1	Signals the type of change. Permitted values: - ADDED - REMOVED - MODIFIED

5.5.3.15 Type: AffectedVirtualStorage

This type provides information about added, deleted, modified and temporary virtual storage resources. It shall comply with the provisions in table 5.5.3.15-1.

Table 5.5.3.15-1: Definition of the AffectedVirtualStorage data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of the storage instance, identifying the applicable "virtualStorageResourceInfo" entry in the "VnfInstance" data type (see clause 5.5.2.2).
virtualStorageDescId	IdentifierInVnfd	1	Identifier of the related VirtualStorage descriptor in the VNFD.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case of a "change current VNF Package" to identify whether the affected virtual storage instance is associated to a VirtualStorage descriptor which is referred from the source or destination VNFD.
changeType	Enum (inlined)	1	Signals the type of change. Permitted values: - ADDED - REMOVED - MODIFIED - TEMPORARY For a temporary resource, an AffectedVirtualStorage structure exists as long as the temporary resource exists.
storageResource	ResourceHandle	1	Reference to the VirtualStorage resource. Detailed information is (for new and modified resources) or has been (for removed resources) available from the VIM.

Attribute name	Data type	Cardinality	Description
resourceDefinitionId	IdentifierLocal	0..1	The identifier of the "ResourceDefinition" in the granting exchange related to the LCM operation occurrence. It shall be present when an applicable GrantInfo for the granted resource exists. See note.
zoneId	Identifier	0..1	The identifier of the resource zone, as managed by the resource management layer (typically, the VIM), where the referenced VirtualStorage resource is placed. Shall be provided if this information is available from the VIM.
metadata	KeyValuePairs	0..1	Metadata about this resource. The content of this attribute shall be a copy of the content of the "metadata" attribute of the VirtualStorageResourceInfo structure.
NOTE: The "resourceDefinitionId" attribute provides information to the API consumer (i.e. the NFVO) to assist in correlating the resource changes performed during the LCM operation with the granted resources in a specific Grant exchange, which is identified by the "grantId" available in the "Individual VNF lifecycle management operation occurrence" and the "id" in the "Individual Grant".			

5.5.3.16 Type: LccnLinks

This type represents the links to resources that a notification can contain. It shall comply with the provisions defined in table 5.5.3.16-1.

Table 5.5.3.16-1: Definition of the LccnLinks data type

Attribute name	Data type	Cardinality	Description
vnfInstance	NotificationLink	1	Link to the resource representing the VNF instance to which the notified change applies.
subscription	NotificationLink	1	Link to the related subscription.
vnfLcmOpOcc	NotificationLink	0..1	Link to the VNF lifecycle management operation occurrence that this notification is related to. Shall be present if there is a related lifecycle operation occurrence.

5.5.3.17 Type: VnfExtCpInfo

This type represents information about an external CP of a VNF. It shall comply with the provisions defined in table 5.5.3.17-1.

Table 5.5.3.17-1: Definition of the VnfExtCpInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of the external CP instance and the related information instance.
cpdId	IdentifierInVnfd	1	Identifier of the external CPD, VnfExtCpd, in the VNFD.
cpConfigId	IdentifierInVnf	1	Identifier that references the applied "VnfExtCpConfig" entry in the "cpConfig" map of the "currentVnfExtCpData" in the "ExtVirtualLinkInfo" structure.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case the value differs from the vnfdId attribute of the VnfInstance (e.g. during a "Change current VNF package" operation or due to its final failure).
cpProtocolInfo	CpProtocolInfo	1..N	Network protocol information for this CP.
extLinkPortId	Identifier	0..1	Identifier of the "ExtLinkPortInfo" structure inside the "ExtVirtualLinkInfo" structure. Shall be present if the CP is associated to a link port. See note 2.

Attribute name	Data type	Cardinality	Description
metadata	KeyValuePairs	0..1	Metadata about this external CP.
associatedVnfcCpId	IdentifierInVnf	0..1	Identifier of the "vnfcCpInfo" structure in "VnfcResourceInfo" structure that represents the VNFC CP which is exposed by this external CP instance, either directly or via a floating IP address. Shall be present in case this CP instance maps to a VNFC CP. See note 1.
associatedVipCpId	IdentifierInVnf	0..1	Identifier of the VIP CP instance that is exposed as this VnfExtCp instance, either directly or via a floating IP address, and the related "VipCpInfo" structure in "VnfInstance". Shall be present if the cpdId of this VnfExtCp has a vipCpd attribute. See note 1.
associatedVirtualCpId	IdentifierInVnf	0..1	Identifier of the "VirtualCpInfo" structure that represents the Virtual CP that is exposed by this external CP instance. Shall be present in case this CP instance maps to a Virtual CP. See note 1.
associatedVnfVirtualLinkId	IdentifierInVnf	0..1	Identifier of the "VnfVirtualLinkResourceInfo" structure that represents the internal VL or of the "ExtManagedVirtualLinkInfo" structure that represents the externally-managed internal VL which is exposed by this external CP instance. Shall be present in case this CP instance maps to an internal VL (including externally-managed internal VL). See note 1.
netAttDefResourceId	Identifier	0..N	Identifier of the "NetAttDefResourceInfo" structure that provides the specification of the interface to attach the connection point to a secondary container cluster network. See notes 3 and 4. It shall be present if the external CP is associated to a VNFC realized by one or a set of OS containers and is connected to a secondary container cluster network. It shall not be present otherwise.
NOTE 1: The attributes "associatedVnfcCpId", "associatedVipCpId", "associatedVirtualCpId" and "associatedVnfVirtualLinkId" are mutually exclusive. Exactly one shall be present. NOTE 2: An external CP instance is not associated to a link port in the cases indicated for the "extLinkPorts" attribute in clause 4.4.1.11. NOTE 3: Cardinality greater than 1 is only applicable for specific cases where more than one network attachment definition resource is needed to fulfil the connectivity requirements of the external CP, e.g. to build a link redundant mated pair in SR-IOV cases. NOTE 4: When more than one netAttDefResourceId is indicated, all shall belong to the same namespace.			

5.5.3.18 Type: VnfLinkPortData

This type represents an externally provided link port to be used to connect a VNFC connection point to an exernally-managed VL. It shall comply with the provisions defined in table 5.5.3.18-1.

Table 5.5.3.18-1: Definition of the VnfLinkPortData data type

Attribute name	Data type	Cardinality	Description
vnfLinkPortId	Identifier	1	Identifier of this link port as provided by the entity that has created the link port.
resourceHandle	ResourceHandle	1	Resource handle of the virtualised resource that realizes the link port.

5.5.3.19 Type: VnfcSnapshotInfo

This type represents a VNFC snapshot. It shall comply with the provisions defined in table 5.5.3.19-1.

Table 5.5.3.19-1: Definition of the VnfcSnapshotInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierLocal	1	Identifier of the information held by the VNFM about a specific VNFC snapshot. This identifier is allocated by the VNFM and is unique within the scope of a VNF snapshot. The attribute also identifies the compute snapshot image associated to this VNFC snapshot within the context of a referred VNF snapshot.
vnfcInstanceId	IdentifierInVnf	1	Identifier of the snapshotted VNFC instance.
creationStartedAt	DateTime	1	Timestamp indicating when the VNFC Snapshot creation has been started by the VNFM.
creationFinishedAt	DateTime	0..1	Timestamp indicating when the VNFC snapshot has been completed. Shall be present once the VNFC Snapshot creation has been completed by the VNFM.
vnfcResourceInfoId	IdentifierInVnf	1	Reference to the "VnfcResourceInfo" structure in the "VnfInstance" structure that represents the resources of the snapshotted VNFC instance. A snapshot of that structure is available in the "vnfInstance" attribute of the "VnfSnapshot" structure.
computeSnapshotResource	ResourceHandle	0..1	Reference to a compute snapshot resource. See note 1.
storageSnapshotResources	Structure (inlined)	0..N	Mapping of the storage resources associated to the VNFC with the storage snapshot resources.
>storageResourceId	IdentifierInVnf	1	Reference to the "VirtualStorageResourceInfo" structure in the "VnfInstance" structure that represents the virtual storage resource. The attribute also identifies the storage snapshot image associated to this VNFC snapshot within the context of a referred VNF snapshot.
>storageSnapshotResource	ResourceHandle	0..1	Reference to a storage snapshot resource. See note 2.
userDefinedData	KeyValuePairs	0..1	User defined data for the VNFC snapshot.

NOTE 1: The identifier of the compute snapshot resource is assigned during creation of a VNFC snapshot being returned from the VIM as output data in the response message of the individual resource operations. This attribute shall only be present for a VNFC snapshot that has been newly created by the VNFM as a result of the "Create VNF snapshot task".

NOTE 2: The identifier of the storage snapshot resource is assigned during creation of a VNFC snapshot being returned from the VIM as output data in the response message of the individual resource operations. This attribute shall only be present for a VNFC snapshot with an associated storage resource and that has been newly created by the VNFM as a result of the "Create VNF snapshot task".

5.5.3.20 Type: VnfStateSnapshotInfo

This type represents information about VNF-specific state snapshot data. It shall comply with the provisions defined in table 5.5.3.20-1.

Table 5.5.3.20-1: Definition of the VnfStateSnapshotInfo data type

Attribute name	Data type	Cardinality	Description
checksum	Checksum	1	Checksum of the VNF state snapshot file. Hash algorithms applicable to VNF snapshot package artifacts are defined in ETSI GS NFV-SOL 010 [i.14].
isEncrypted	Boolean	1	Reflects whether the VNF state snapshot content is encrypted (true) or not (false).
metadata	KeyValuePairs	0..1	The metadata with additional information such as content type, size, creation date, etc.

5.5.3.21 Type: ModificationsTriggeredByVnfPkgChange

This type represents attribute modifications that were performed on an "Individual VNF instance" resource when changing the current VNF package. The attributes that can be included consist of those requested to be modified explicitly in the "ChangeCurrentVnfPkgRequest" data structure, and additional attributes of the "VnfInstance" data structure that were modified implicitly during the operation.

The "ModificationsTriggeredByVnfPkgChange" data type shall comply with the provisions defined in table 5.5.3.21-1.

Table 5.5.3.21-1: Definition of the ModificationsTriggeredByVnfPkgChange data type

Attribute name	Data type	Cardinality	Description
vnfConfigurableProperties	KeyValuePairs	0..1	This attribute signals the modifications of the "vnfConfigurableProperties" attribute in "VnfInstance" performed by the operation and shall be present if that attribute was modified during the operation. See note 1. In addition, the provisions in clause 5.7 shall apply.
metadata	KeyValuePairs	0..1	This attribute signals the modifications of the "metadata" attribute in "VnfInstance" performed by the operation and shall be present if that attribute was modified during the operation. See note 1.
extensions	KeyValuePairs	0..1	This attribute signals the modifications of the "extensions" attribute in "VnfInstance" performed by the operation and shall be present if that attribute was modified during the operation. See note 1. In addition, the provisions in clause 5.7 shall apply.
vnfdId	Identifier	0..1	If present, this attribute signals the new value of the "vnfdId" attribute in "VnfInstance".
vnfProvider	String	0..1	If present, this attribute signals the new value of the "vnfProvider" attribute in "VnfInstance". See note 2.
vnfProductName	String	0..1	If present, this attribute signals the new value of the "vnfProductName" attribute in "VnfInstance". See note 2.
vnfSoftwareVersion	Version	0..1	If present, this attribute signals the new value of the "vnfSoftwareVersion" attribute in "VnfInstance". See note 2.
vnfdVersion	Version	0..1	If present, this attribute signals the new value of the "vnfdVersion" attribute in "VnfInstance". See note 2.
vimConnectionInfo	map(VimConnectionInfo)	0..N	If present, this attribute signals the changes to VIM connection info that were passed in the related "ChangeCurrentVnfPkgRequest" structure. The provisions for sensitive information defined in clause 4.4.1.6 apply.

NOTE 1: This attribute represents the delta (semantics as per IETF RFC 7396 [5], JSON Merge Patch) between the value of the attribute at the start of the "Change current VNF package" operation and the value of the attribute at its completion.

NOTE 2: If present, this attribute (which depends on the value of the "vnfdId" attribute) was modified implicitly during the related operation and contains a copy of the value of the related attribute from the VNFD in the VNF Package identified by the "vnfdId" attribute.

5.5.3.22 Type: VipCpInfo

This type provides information related to virtual IP (VIP) CP. It shall comply with the provisions defined in table 5.5.3.22-1.

Table 5.5.3.22-1: Definition of the VipCpInfo data type

Attribute name	Data type	Cardinality	Description
cpInstanceId	IdentifierInVnf	1	Identifier of this VIP CP instance and of this VipCpInfo information element.
cpdId	IdentifierInVnfd	1	Identifier of the VIP Connection Point Descriptor, VipCpd, in the VNFD.
vnfdId	Identifier	0..1	Identifier of the VNFD. Shall be present in case the value differs from the vnfdId attribute of the VnfInstance (e.g. during a "Change current VNF package" operation or due to its final failure). See note 2.
vnfExtCpId	IdentifierInVnf	0..1	When the VIP CP is exposed as external CP of the VNF, the identifier of this external VNF CP instance.
cpProtocolInfo	CpProtocolInfo	0..N	Protocol information for this CP. There shall be one cpProtocolInfo for layer 3. There may be one cpProtocolInfo for layer 2.
associatedVnfcCpIds	IdentifierInVnf	0..N	Identifiers of the VnfcCps that share the virtual IP addresse allocated to the VIP CP instance. See note 1.
vnfLinkPortId	IdentifierInVnf	0..1	Identifier of the "VnfLinkPortInfo" structure in the "VnfVirtualLinkResourceInfo" or "ExtManagedVirtualLinkInfo" structure. Shall be present if the CP is associated to a link port on an internal VL (including externally-managed internal VL).
metadata	KeyValuePairs	0..N	Metadata about this VIP CP.
NOTE 1: It is possible that there is no associated VnfcCp because the VIP CP is available but not associated yet. NOTE 2: If only the value or the presence of this attribute is changed in the "VipCpInfo" structure by an LCM operation occurrence, this does not represent a change that requires including a related "AffectedVipCp" structure in the VNF LCM operation occurrence notifications or the "VnfLcmOpOcc" structure related to this LCM operation occurrence.			

5.5.3.23 Type: AffectedVirtualCp

This type provides information about added, deleted and modified virtual CP instances. It shall comply with the provisions defined in table 5.5.3.23-1.

Table 5.5.3.23-1: Definition of the AffectedVirtualCp data type

Attribute name	Data type	Cardinality	Description
cpInstanceId	IdentifierInVnf	1	Identifier of the virtual CP instance and the related "VirtualCpInfo" structure in "VnfInstance".
cpdId	IdentifierInVnfd	1	Identifier of the VirtualCpd in the VNFD.
vnfdId	Identifier	0..1	Reference to the VNFD. Shall be present in case of a "change current VNF Package" to identify whether the affected virtual CP instance is associated to a VirtualCpd which is referred from the source or destination VNFD.
changeType	Enum	1	Signals the type of change. Permitted values: - ADDED - REMOVED - MODIFIED

5.5.3.24 Type: McioInfo

This type provides information about an MCIO representing the set of VNFC instances realized by one or a set of OS containers which have been created based on the same VDU.

Within the CISM, an MCIO controller monitors the actual state of an MCIO representing the set of VNFC instances realized by one or a set of OS containers and compare it to the desired state as specified in the respective declarative descriptor. It triggers actions toward the CIS to align the actual to the desired state. Monitoring the actual state includes monitoring the number of MCIO instances available at any specific point in time. In addition, an MCIO controller maintains properties and runtime information on the MCIO instances which have been created based on the same VDU. The McioInfo data structure provides the runtime information on the MCIOs obtained from the MCIO controller.

NOTE: There are different types of MCIOs. The set of VNFC instances based on the same VDU is represented by one MCIO, e.g. of type Deployment. Each individual VNFC instance is represented by another type of MCIO, e.g. a POD.

Runtime information of the set of OS containers realizing an individual VNFC instance is not part of the McioInfo data structure; such runtime information is provided in the ResourceHandle data structure referenced from the VnfcResourceInfo. The McioInfo does not provide runtime information of a constituent VNFC instance created based on a specific VDU.

It shall comply with the provisions defined in table 5.5.3.24-1.

Table 5.5.3.24-1: Definition of the McioInfo data type

Attribute name	Data type	Cardinality	Description
mcioId	Identifier	1	Identifier of this MCIO, created by the CISM. See note 3.
mcioName	String	1	Human readable name of this MCIO. See note 4.
mcioNamespace	String	1	Namespace of this MCIO.
vduId	IdentifierInVnfd	1	Reference to the related VDU in the VNFD applicable to this resource.
cismId	Identifier	1	Identifier of the CISM managing this MCIO.
mcioType	String	1	The type of MCIO. Specific values, their semantics and associated MCIO types are defined in clause 5.5.4.9. Additional values are also permitted. See note 1.
desiredInstances	Integer	1	Number of desired MCIO instances.
availableInstances	Integer	1	Number of available MCIO instances.
additionalInfo	KeyValuePairs	0..1	Additional information which is specific to the MCIO, its type, and which is available from the CISM. See note 2.

NOTE 1: The type of MCIO as specified in the declarative descriptor of the MCIO, and that can be read from the CISM. EXAMPLE: In case of MCIOs managed by Kubernetes®, the type of MCIO corresponds to the "kind" property of the declarative descriptor.
NOTE 2: If the attribute additionalInfo is present, it may contain runtime information on the actual and desired state of the MCIO(s).
NOTE 3: When the container infrastructure service is a Kubernetes® instance, the mcioId is the combined values from the kind and name fields of the Kubernetes® resource object, separated by a slash. Example: "Deployment/abcd".
NOTE 4: When the container infrastructure service is a Kubernetes® instance, the mcioName is the name field of the resource object.

5.5.3.25 Type: VirtualCpInfo

This type provides information related to a virtual CP instance of a VNF. It shall comply with the provisions defined in table 5.5.3.25-1.

Table 5.5.3.25-1: Definition of the VirtualCpInfo data type

Attribute name	Data type	Cardinality	Description
cpInstanceId	IdentifierInVnf	1	Identifier of this virtual CP instance.
cpdId	IdentifierInVnfd	1	Identifier of the VirtualCpd in the VNFD.
resourceHandle	ResourceHandle	1	Reference to the virtualised resource realizing this virtual CP.
vnfExtCpId	IdentifierInVnf	0..1	When the virtual CP is exposed as external CP of the VNF, the identifier of this external VNF CP instance.
cpProtocolInfo	CpProtocolInfo	0..N	Protocol information for this CP. There shall be one cpProtocolInfo for each layer protocol supported.
vduIds	IdentifierInVnfd	1..N	Reference to the VDU(s) which implement the service accessible via the virtual CP instance. See note.
additionalServiceInfo	AdditionalServiceInfo	0..N	Additional service identification information of the virtual CP instance.
metadata	KeyValuePairs	0..1	Metadata about this virtual CP instance.
NOTE: A consumer of the VNF LCM interface can learn the actual VNFC instances implementing the service accessible via the virtual CP instance by querying the "vnfcResourceInfo" from the "InstantiatedVnfInfo" and filtering by corresponding "vduIds" values.			

5.5.3.26 Type: AddtionalServiceInfo

This type provides additional service information of the virtual CP instance used to expose properties of the virtual CP to NFV-MANO. It shall comply with the provisions defined in table 5.5.3.26-1.

Table 5.5.3.26-1: Definition of the AddtionalServiceInfo data type

Attribute name	Data type	Cardinality	Description
portInfo	ServicePortInfo	1..N	Service port numbers exposed by the virtual CP instance.
serviceInfo	KeyValuePairs	0..1	Service matching information exposed by the virtual CP instance. See note.
NOTE: This attribute shall only be present if additional information is needed to identify the service termination within the VNF, such as for example a URL path information in an HTTP request required to allow a single virtual CP IP address to be used for several HTTP based services that use the same port number.			

5.5.3.27 Type: ServicePortInfo

This type describes the service identifying port properties exposed by the virtual CP instance. It shall comply with the provisions defined in table 5.5.3.27-1.

Table 5.5.3.27-1: Definition of the ServicePortInfo data type

Attribute name	Data type	Cardinality	Description
name	String	1	The name of the port exposed by the virtual CP instance.
protocol	Enum (inlined)	0..1	The L4 protocol for this port exposed by the virtual CP instance. Permitted values: • TCP • UDP • SCTP
port	Integer	1	The L4 port number exposed by the virtual CP instance.
portConfigurable	Boolean	1	Specifies whether the port attribute value is allowed to be configurable.

5.5.3.28 Type: NetAttDefResourceInfo

This type contains information related to a network attachment definition resource that provides the specification of the interface used to connect one or multiple connection points to a secondary container cluster network. It shall comply with the provisions defined in table 5.5.3.28-1.

Table 5.5.3.28-1: Definition of the NetAttDefResourceInfo data type

Attribute name	Data type	Cardinality	Description
netAttDefResourceInfoId	Identifier	1	Identifier of this network attachment definition resource as provided by the entity that has created it.
netAttDefResource	ResourceHandle	1	Resource handle of the resource in the scope of the CISM.
associatedExtCpId	IdentifierInVnf	0..N	Identifier of the external CP associated to this network attachment definition resource. Shall be present when the network attachment definition resource is used for external connectivity by the VNF.
associatedVnfcCpId	IdentifierInVnf	0..N	Identifier of the VNFC CP associated to this network attachment definition resource. May be present when the network attachment definition resource is used for internal connectivity by the VNF.

5.5.4 Referenced simple data types and enumerations

5.5.4.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

5.5.4.2 Simple data types

No particular simple data types are defined for this interface, in addition to those defined in clause 4.4.

5.5.4.3 Enumeration: VnfOperationalStateType

The enumeration VnfOperationalStateType shall comply with the provisions defined in table 5.5.4.3-1.

Table 5.5.4.3-1: Enumeration VnfOperationalStateType

Enumeration value	Description
STARTED	The VNF instance is up and running.
STOPPED	The VNF instance has been shut down.

5.5.4.4 Enumeration: StopType

The enumeration StopType shall comply with the provisions defined in table 5.5.4.4-1.

Table 5.5.4.4-1: Enumeration StopType

Enumeration value	Description
FORCEFUL	The VNFM will stop the VNF instance immediately after accepting the request.
GRACEFUL	The VNFM will first arrange to take the VNF instance out of service after accepting the request. Once that operation is successful or once the timer value specified in the "gracefulStopTimeout" attribute expires, the VNFM will stop the VNF instance.

5.5.4.5 Enumeration: LcmOperationStateType

The enumeration LcmOperationStateType shall comply with the provisions defined in table 5.5.4.5-1. More information of the meaning of the states can be found in clause 5.6.2.2.

Table 5.5.4.5-1: Enumeration LcmOperationStateType

Enumeration value	Description
STARTING	The LCM operation is starting.
PROCESSING	The LCM operation is currently in execution.
COMPLETED	The LCM operation has been completed successfully.
FAILED_TEMP	The LCM operation has failed and execution has stopped, but the execution of the operation is not considered to be closed.
FAILED	The LCM operation has failed and it cannot be retried or rolled back, as it is determined that such action will not succeed.
ROLLING_BACK	The LCM operation is currently being rolled back.
ROLLED_BACK	The LCM operation has been successfully rolled back, i.e. The state of the VNF prior to the original operation invocation has been restored as closely as possible.

5.5.4.6 Enumeration: CancelModeType

The enumeration CancelModeType defines the valid modes of cancelling a VNF LCM operation occurrence. It shall comply with the provisions defined in table 5.5.4.6-1.

Table 5.5.4.6-1: Enumeration CancelModeType

Enumeration value	Description
GRACEFUL	If the VNF LCM operation occurrence is in "PROCESSING" or "ROLLING_BACK" state, the VNFM shall not start any new resource management operation and shall wait for the ongoing resource management operations in the underlying system, typically the VIM, to finish execution or to time out. After that, the VNFM shall put the operation occurrence into the FAILED_TEMP state. If the VNF LCM operation occurrence is in "STARTING" state, the VNFM shall not start any resource management operation and shall wait for the granting request to finish execution or time out. After that, the VNFM shall put the operation occurrence into the ROLLED_BACK state.

Enumeration value	Description
FORCEFUL	If the VNF LCM operation occurrence is in "PROCESSING" or "ROLLING_BACK" state, the VNFM shall not start any new resource management operation, shall cancel the ongoing resource management operations in the underlying system, typically the VIM, and shall wait for the cancellation to finish or to time out. After that, the VNFM shall put the operation occurrence into the FAILED_TEMP state. If the VNF LCM operation occurrence is in "STARTING" state, the VNFM shall not start any resource management operation and put the operation occurrence into the ROLLED_BACK state.

5.5.4.7 Enumeration: LcmOperationType

The enumeration LcmOperationType defines the permitted values to represent VNF lifecycle operation types in VNF lifecycle management operation occurrence resources and VNF lifecycle management operation occurrence notifications. It shall comply with the provisions defined in table 5.5.4.7-1.

Table 5.5.4.7-1: Enumeration LcmOperationType

Enumeration value	Description
INSTANTIATE	Represents the "Instantiate VNF" LCM operation.
SCALE	Represents the "Scale VNF" LCM operation.
SCALE_TO_LEVEL	Represents the "Scale VNF to Level" LCM operation.
CHANGE_FLAVOUR	Represents the "Change VNF Flavour" LCM operation.
TERMINATE	Represents the "Terminate VNF" LCM operation.
HEAL	Represents the "Heal VNF" LCM operation.
OPERATE	Represents the "Operate VNF" LCM operation.
CHANGE_EXT_CONN	Represents the "Change external VNF connectivity" LCM operation.
MODIFY_INFO	Represents the "Modify VNF Information" LCM operation.
CREATE_SNAPSHOT	Represents the "Create VNF Snapshot" LCM operation.
REVERT_TO_SNAPSHOT	Represents the "Revert To VNF Snapshot" LCM operation.
CHANGE_VNFPKG	Represents the "Change current VNF package" LCM operation.

5.5.4.8 Enumeration: LcmOpOccNotificationVerbosityType

The enumeration LcmOpOccNotificationVerbosityType provides values to control the verbosity of LCM operation occurrence notifications. It shall comply with the provisions defined in table 5.5.4.8-1.

Table 5.5.4.8-1: Enumeration VnfOperationalStateType

Enumeration value	Description
FULL	This signals a full notification which contains all change details.
SHORT	This signals a short notification which omits large-volume change details to reduce the size of data to be sent via the notification mechanism.

5.5.4.9 Type: McioTypeName

The present type definition provides valid string values for the "mcioType" attribute of the "McioInfo" data type , and the valid values are defined in table 5.5.4.9-1.

Table 5.5.4.9-1: String values for MCIO types

String value	Description
"Deployment"	Represents the Deployment compute MCIO as defined in ETSI GS NFV-SOL 018 [i.17].
"Statefulset "	Represents the StatefulSet compute MCIO as defined in ETSI GS NFV-SOL 018 [i.17].

5.6 Success and error states of VNF lifecycle management operations

5.6.1 Basic concepts for error handling (informative)

5.6.1.1 Motivation

VNF lifecycle management operation occurrences can fail. Failure can be caused by multiple reasons, which generally fall into the following categories:

- Transient errors which do not require intervention from a human operator or a higher-layer management entity for resolution, e.g. momentary network outage.
- "Permanent" errors which require such intervention.

It is unreasonable to expect that all errors can be resolved automatically, therefore the possibility of intervention will usually be incorporated in the system design as acknowledged means of error resolution.

5.6.1.2 Failure resolution strategies: Retry and Rollback

Most transient errors are handled best with a retry mechanism. Retry might happen automatically at the point of failure within the same LCM workflow (where it makes sense to limit the number of automatic retries). It is important to strive for designing retry operations that have no unintended side effects from the original invocation of the operation. This is called *idempotent retry*. Idempotent retry can also be used as an on-demand error resolution mechanism (see below) if the original operation failed because of a condition that has been resolved manually by the human operator or by a higher-level management entity, so idempotent retry is suitable for general error resolution in most cases.

However, even if a system is designed with idempotent retry capabilities, eventual success of the operation cannot be guaranteed. In this case, the resolution of the inconsistent state can be attempted by requesting to roll back the changes made by the operation. Therefore, rollback as an error handling strategy is also desired to be allowed in the system design.

In many cases, idempotent retry can resolve transient errors and lead to success eventually. Depending on the situation, rollback followed by a repetition of the operation could take longer than a successful retry, as rollback first removes allocated resources and then the repetition of the operation allocates them again, which costs time.

Therefore, it often makes sense to perform first idempotent retry, which is followed by rollback if the retry has failed. Idempotent retry is meaningful and useful for all operation types, but for some operations rollback is better suited and has a better chance of success. In general, rollback is well-suited for additive operations such as InstantiateVnf or scale out, while ill-suited for subtractive ones such as scale in or TerminateVnf, or for HealVnf.

Both rollback and idempotent retry can fail. In that case, the system can be left in an inconsistent state after a failed operation, which requires resolution by a higher-level entity such as NFVO or human operator.

5.6.1.3 Error handling at VNFM and NFVO

If the VNFM executes an LCM workflow and encounters a problem, the following options are possible:

- Stop on first error:
  - Once the VNFM encounters an error, the normal execution of the LCM workflow is interrupted, and an error handling procedure is triggered (automatic retry, automatic rollback, automatic fail, escalate). See the paragraphs below for description of error handling procedures.
  - It is assumed that all VNFs and all VNFMs support "stop on first error".

EXAMPLE 1: NFVO is attempting to instantiate a VNF with 100 VNFCs. The first 97 VNFCs are instantiated successfully, however, an error occurs when attempting to instantiate VNFC #98. The VNFM stops execution and chooses which of the error handling options it invokes (note that it even could try multiple options after each other).

- Best Effort:
  - Each time the VNFM encounters an error, it is decided whether the execution of a part or all of the remaining steps of the LCM workflow is performed, or whether the execution is interrupted and an error handling procedure is triggered (automatic retry, automatic rollback, automatic fail, escalate). See the paragraphs below for description of error handling procedures.
  - Support of "best effort" requires a suitable workflow design.
  - It is therefore assumed that not all VNFs and not all VNFMs support "best effort".

EXAMPLE 2: Same example as above. After the error occurs attempting to instantiate VNFC #98, the VNFM continues by creating #99 and #100, and then chooses which error handling options it invokes.

The VNFM has the following error handling procedures to react to errors (see clause 5.6.1.2 for general elaboration regarding retry and rollback):

- Automatic Retry: The VNFM retries (once or more) to continue the execution of the workflow without involving an external entity. Automatic retry of failed parts of the workflow might even be built into the workflow itself. Retry can eventually succeed or fail. Successful retry leads to the LCM operation to be reported as successful. Failed retry is typically escalated.
- Automatic Rollback: The VNFM rolls back the VNF to the state prior to starting the LCM operation without involving an external entity. Rollback can eventually succeed or can fail, preventing the VNF from reaching that previous state. Successful rollback leads to the LCM operation to be reported as rolled back. Failed rollback is typically escalated.
- Escalate: After failed automatic retry/retries, automatic rollback is typically not the first option in most situations, but the error is preferably reported to the NFVO for further resolution. The same applies if no automatic error resolution was attempted by the VNFM, or if automated rollback has failed. This is done by sending a VNF LCM operation occurrence notification.
- Unresolvable Error: The VNFM determines that the operation has failed and definitely cannot be recovered (e.g. if no retry and no rollback is possible), and that escalating the error to the NFVO will have no chance to lead to a resolution either. In this case, the VNFM would report that the operation has terminally failed. After that, other means of resolution can be attempted, such as the invocation of HealVnf, or manual procedures using the GUI of the VNFM or VIM to release stranded resources.

The NFVO has the following error handling procedures to react to error reports from the VNFM:

- On-demand retry: After the VNFM has reported the error to the NFVO, the NFVO or the human operator takes steps to resolve the situation that has led to the occurrence of the error. Subsequently, the retry of the operation is triggered towards the VNFM by the NFVO via the VNF LCM interface.
- On-demand rollback: After the VNFM has reported the error to the NFVO, and after the NFVO or the human operator has decided to roll back the operation, the rollback of the operation is triggered towards the VNFM by the NFVO via the VNF LCM interface.
- Fail: After the VNFM has reported the error to the NFVO, and after the NFVO or the human operator has determined that neither on-demand retry nor on-demand rollback will fix the error, the LCM operation can be declared as terminally failed towards the VNFM by the NFVO via the VNF LCM interface. After that, other means of resolution can be attempted, such as the invocation of HealVnf, or manual procedures using the GUI of the VNFM or VIM to release stranded resources.

5.6.2 States and state transitions of a VNF lifecycle management operation occurrence

5.6.2.1 General

A VNF lifecycle management operation occurrence supports a number of states and error handling operations. The states and state transitions that shall be supported by the VNFM are shown in figure 5.6.2.1-1. Transitions labelled with underlined text represent error handling operations; other transitions represent conditions.

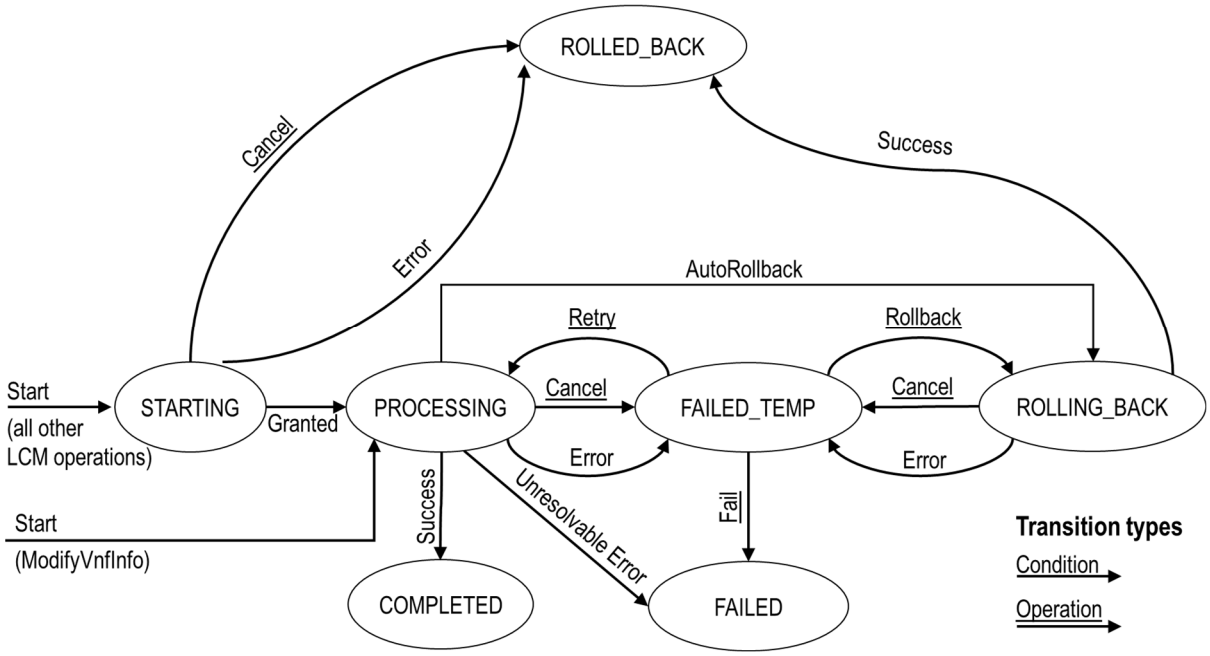

Figure 5.6.2.1-1: States of a VNF lifecycle management operation occurrence

5.6.2.2 States of a VNF lifecycle management operation occurrence

At each time, a VNF lifecycle management operation occurrence is in one of the following states. There are transient states (states from which a different state can be reached) and terminal states (states from which no other state can be reached; i.e. the state of a VNF lifecycle management operation occurrence in a terminal state cannot change anymore).

STARTING: The operation is starting. This state represents the preparation phase of the operation, including invoking Grant Lifecycle Operation. This state has the following characteristics:

- This is the initial state for any LCM operation except ModifyVnfInformation.
- This is a transient state.
- This state may block other LCM operations from being executed on the same VNF instance (up to VNF and VNFM implementation).
- In this state, the VNF lifecycle management operation occurrence does not perform any changes to the VNF instance or to resources.
- Once the VNF lifecycle operation has been granted, the VNF lifecycle management operation occurrence shall transit into the PROCESSING state.
- If the LCM operation is cancelled in the "STARTING" state, the VNF lifecycle management operation occurrence shall transit to the "ROLLED_BACK" state. The NFVO shall be prepared to receive the notification about the cancellation of the operation before and after having provided the grant. This is necessary to address possible race conditions.
- If an error occurs before the VNFM receives the grant response, or the grant is rejected, as no changes to the underlying VNF or resources were done, the VNF lifecycle management operation occurrence shall transit into the "ROLLED_BACK" state.

COMPLETED: The operation has completed successfully. This is a terminal state.

FAILED_TEMP: The operation has failed and execution has stopped, but the execution of the operation is not considered to be closed. This state has the following characteristics:

- This is a transient state.
- The grant received for the operation is still valid, and the granted resource changes are still foreseen for the VNF.
- This state may block other LCM operations from being executed on the same VNF instance (enforced by the VNFM, and up to VNF and VNFM capabilities).
- Retry or rollback or fail may be invoked for the operation.
- If the VNF LCM operation is retried, the VNF lifecycle management operation occurrence shall transit into the "PROCESSING" state.
- If the VNF LCM operation is rolled back, the VNF lifecycle management operation occurrence shall transit into the "ROLLING_BACK" state.
- If the VNF LCM operation is marked as "failed", the VNF lifecycle management operation occurrence shall transit into the "FAILED" state.
- Operation cancellation and failure to roll back should result in FAILED_TEMP.

FAILED: The operation has failed and it cannot be retried or rolled back, as it is determined that such action will not succeed. This state has the following characteristics:

- This is a terminal state.
- Such an operation state is typically the result of a decision of a higher layer management entity (NFVO) or its human operator that an operation in "FAILED_TEMP" state cannot be retried or rolled back ("Fail").
- Such an operation state can also be reached immediately in case of failure of an operation in "PROCESSING" state that can neither be retried nor rolled back ("Unresolvable Error").

NOTE 1: The direct transition from "PROCESSING" into "FAILED" state is deprecated and only provided for backward compatibility with legacy; implementations need to be aware that support can be removed in subsequent versions of the present document.

- The result of the LCM operation (the actual resource changes) can show an inconsistent state of the VNF and can reflect partial resource changes compared to the granted changes. Nevertheless, these resource changes, as known by the VNFM, shall be synchronized between the VNFM and NFVO (by reporting them in the LCCN, and by allowing the NFVO to obtain them on request) in order for other VNF LCM operations (e.g. Heal, Terminate) to be guaranteed to work on resources that are known to the NFVO.

NOTE 2: In certain error cases during a procedure that requires interactions with the VIM, the information about VIM resources known by the VNFM might not be accurate.

- The fact that an LCM operation is in "FAILED" state shall not block other operations from execution on the VNF instance by the VNFM. However, the VNF instance may itself be in a state that disallows certain operations.

ROLLED_BACK: The state of the VNF prior to the original operation invocation has been restored as closely as possible. This state has the following characteristics:

- This is a terminal state.
- This may involve recreating some resources that have been deleted by the operation, the recreated resources should be as similar as possible to the deleted ones. Differences between original resources and re-created ones may include a different resource identity, but also different dynamic attributes such as an IP address.

PROCESSING: The LCM operation is currently in execution. This state has the following characteristics:

- This is the initial state for the "ModifyVnfInformation" operation.

- This is a transient state.
- This state may block other LCM operations from being executed on the same VNF instance (up to VNF and VNFM implementation).
- The operations "Retry" and "Rollback" shall not be permitted to be invoked for an operation that is in this state.
- All failures of procedures executed by the VNFM as part of the LCM operation while in "PROCESSING" state shall result in transiting to "FAILED_TEMP", with the following two exceptions:
  - If a failure occurs in the "PROCESSING" state from which the VNFM knows that the VNF instance can be brought into a consistent state by immediately rolling back the operation, the VNF lifecycle management operation occurrence may transit directly into the "ROLLING_BACK" state ("Autorollback"). For the "ModifyVnfInformation" operation, Autorollback is the typical error handling method.
  - If a failure occurs in the "PROCESSING" state from which the VNFM knows that it can neither be fixed by retrying nor be rolled back, the VNF lifecycle management operation occurrence may transit directly into the "FAILED" state ("Unresolvable Error").

NOTE 3: The direct transition from "PROCESSING" into "FAILED" state is deprecated and only provided for backward compatibility with legacy; implementations need to be aware that support can be removed in subsequent versions of the present document.

- If a "cancel" request was issued during the operation is in "PROCESSING" state, processing will be cancelled but this might not be immediate. This is represented by a flag in the data model that indicates there is a pending "cancel" request for this state. Upon successful cancellation, the VNF lifecycle management operation occurrence shall transit into the "FAILED_TEMP" state.

ROLLING_BACK: The LCM operation is currently being rolled back. This state has the following characteristics:

- This is a transient state.
- This state may block other LCM operations from being executed on the same VNF instance (up to VNF and VNFM implementation).
- The operations "Retry" and "Rollback" shall not be permitted to be invoked for an operation that is in this state.
- If a "Cancel" request was issued during the operation is in "ROLLING_BACK" state, rolling back will be cancelled but this might not be immediate. This is represented by a flag in the data model that indicates there is a pending "Cancel" request for this state. Upon successful cancellation, the VNF lifecycle management operation occurrence shall transit into the "FAILED_TEMP" state.
- If a failure occurs during rolling back, the operation should transition to the "FAILED_TEMP" state.
- Upon successful rollback, the VNF lifecycle management operation occurrence shall transit into the "ROLLED_BACK" state.

The following provisions apply to the sending of VNF lifecycle management operation occurrence notifications by the VNFM:

- The "start" notification (i.e. notificationStatus="START") shall be sent each time when the operation enters one of states "STARTING", "PROCESSING" and "ROLLING_BACK" from another state, indicating the state entered in the "operationState" attribute.
- The "result" notification (i.e. notificationStatus="RESULT") shall be sent each time when the VNF LCM operation occurrence enters one of the error states "FAILED_TEMP", "FAILED", "ROLLED_BACK", indicating the state entered in the "operationState" attribute, as well as the error cause and the changes to the VNF's resources since the operation was initially started.
- The "result" notification (i.e. notificationStatus="RESULT") shall be sent when the operation enters the success state "COMPLETED", indicating the state entered in the "operationState" attribute, as well as the changes to the VNF's resources.

The following provisions apply to the sending of notifications related to VNF lifecycle changes (VNF LCM operation Occurrence Notifications, VNF identifier creation and VNF identifier deletion notifications):

- The processing of a VNF LCM operation occurrence shall not wait for the acknowledgement of the delivery of the triggered notifications.
- Invoking a subsequent LCM operation on the same VNF instance shall not be blocked while waiting for the acknowledgement of the delivery of all notifications triggered by a previous LCM operation occurrence on the same VNF instance.

Such a notification scheme allows the NFVO to keep in sync with changes to the VNF's resources by an ongoing LCM operation. If the notification relates to a transient state, further changes can be expected. If the notification relates to a terminal state, no further changes to the VNF's resources will be performed by the related VNF lifecycle management operation occurrence. In order to avoid inconsistent information about the state and result of the VNF lifecycle management operation by the NFVO, which can impact the error handling procedure, the state of the VNF lifecycle management operation shall be synchronized between the VNFM and NFVO. The NFVO can use the information in the notification to synchronize its internal state with the current state and result of the LCM operation. In case of loss of notifications, the NFVO can read the resource that represents the VNF lifecycle management operation occurrence to obtain the same information.

5.6.2.3 Error handling operations that change the state of a VNF lifecycle management operation occurrence

Retry: This operation retries a VNF lifecycle operation. It has the following characteristics:

- Execution of "Retry" for an actual LCM operation on a particular VNF may be supported, depending on characteristics of the VNF and the LCM operation.
- "Retry" shall operate within the bounds of the Grant for the LCM operation.
- The operation may be invoked via an interface, or the VNFM may invoke the operation per its own decision.

Rollback: This operation rolls back a VNF lifecycle operation. It has the following characteristics:

- Execution of "Rollback" for an actual LCM operation on a particular VNF may be supported, depending on characteristics of the VNF and the LCM operation.
- "Rollback" shall operate within the bounds of the Grant for the LCM operation, an additionally may execute the inverse of granted LCM operations (e.g. if a resource deletion was granted, rollback might re-create the deleted resource or a similar resource).
- The operation may be invoked via an interface, or the VNFM may invoke the operation per its own decision.

Fail: This operation transits the VNF lifecycle management operation occurrence into the terminal "FAILED" state. It has the following characteristics:

- Execution of "Fail" shall be supported for an LCM operation on a particular VNF if at least one of Retry, Rollback, Cancel is supported for this operation.
- The operation may be invoked via an interface, or the VNFM may invoke the operation per its own decision.

Cancel: This operation cancels an ongoing VNF lifecycle management operation, its Retry or Rollback. It has the following characteristics:

- Execution of Cancel for an actual LCM operation on a particular VNF may be supported, depending on characteristics of the VNF and the LCM operation.
- The "Cancel" operation need not have immediate effect, depending on the capabilities of the underlying systems, and the currently executed resource management operation.

- Two modes of cancellation are supported: graceful and forceful:
  - When executing the *graceful* "Cancel" operation, the VNFM will not initiate any new operation towards the underlying systems, will wait until the currently executed operations finish, fail or time out in the VNFM, and will then put the VNF lifecycle management operation occurrence into the "FAILED_TEMP" state.
  - When executing the *forceful* "Cancel" operation, the VNFM will cancel all ongoing operations in the underlying systems for which cancellation is supported, will not initiate any new operation towards the underlying systems, will wait for the requested cancellations to finish, fail or time out in the VNFM, and will then put the VNF lifecycle management operation occurrence into the "FAILED_TEMP" state.

NOTE: In both modes, the time-out is determined by means outside the scope of the present document.

  - In "STARTING" state, there is no difference between the graceful and the forceful cancellation mode.

- Executing "Cancel" can lead to inconsistencies between the information that the VNFM has about the state of the resources of the VNF, and their actual state. The probability of such inconsistencies is bigger when using the *forceful* cancellation mode.

5.6.3 Detailed flows for error handling

5.6.3.1 Immediate failure

If the VNF LCM operation fails immediately, i.e. it returns an HTTP error, then the operation has not started, and no "Individual VNF LCM operation occurrence" resource has been created. Also, neither a "start" VNF lifecycle management operation occurrence notification nor a Grant request has been sent. The operation cannot be retried, but the same operation may be invoked again from the API. The VNF instance is not changed by a synchronous failure, so no special error handling is required.

Figure 5.6.3.1-1 illustrates the flow.

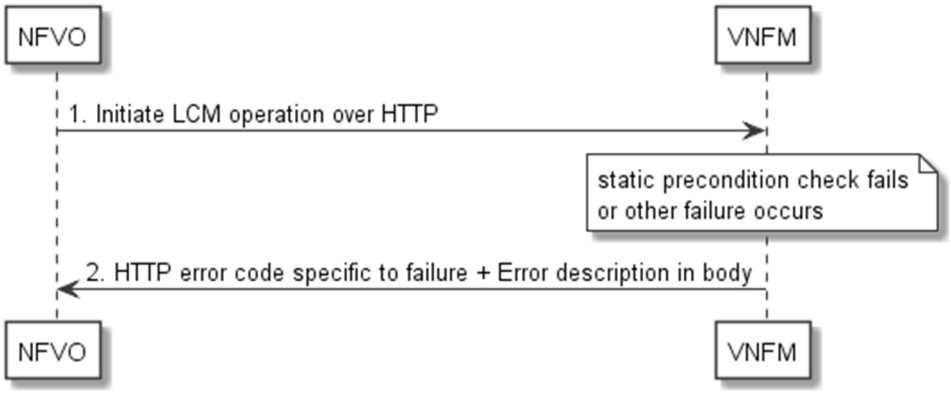

Figure 5.6.3.1-1: Immediate failure of a VNF LCM operation

5.6.3.2 Failure in "STARTING" state

This error scenario assumes that the "Individual VNF LCM operation occurrence" resource has been created and the "start" VNF lifecycle management operation occurrence notification has been sent.

If the operation fails before the VNFM receives the Grant response, or the Grant is rejected, persistent change to the state of the VNF cannot have happened. Therefore, it is assumed that this operation enters the ROLLED_BACK state immediately. Figure 5.6.3.2-1 illustrates the flow.

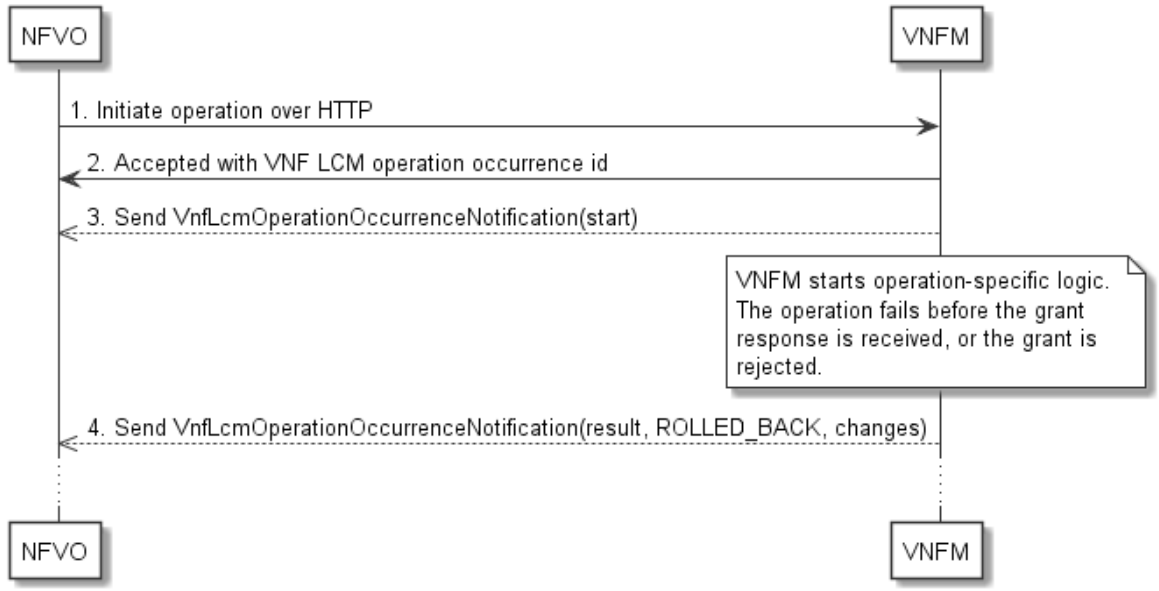

Figure 5.6.3.2-1: Failure of a VNF LCM operation before applying any change to the VNF instance

5.6.3.3 Failure during actual LCM operation execution

After a failed resource management operation, automatic retry can be invoked by the VNFM itself. These invocations are not visible outside of the VNFM, as the VNF LCM operation occurrence stays in "PROCESSING" state during these automatic retries. If these do not resolve the issue, intervention (typically by a human operator) is necessary. For that purpose, the LCM operation is set into a temporary failure state, and the NFVO is notified. The human operator performs a root cause analysis and eventually resolves the obstacle. Subsequently, and if supported, the operation can be retried, rolled-back or determined as permanently failed. Figure 5.6.3.3-1 illustrates the possible options.

NOTE 1: Excluding automated rollback which is seen as a rare option.

NOTE 2: Excluding "start" notifications (i.e. notificationStatus="START") for simplification purposes.

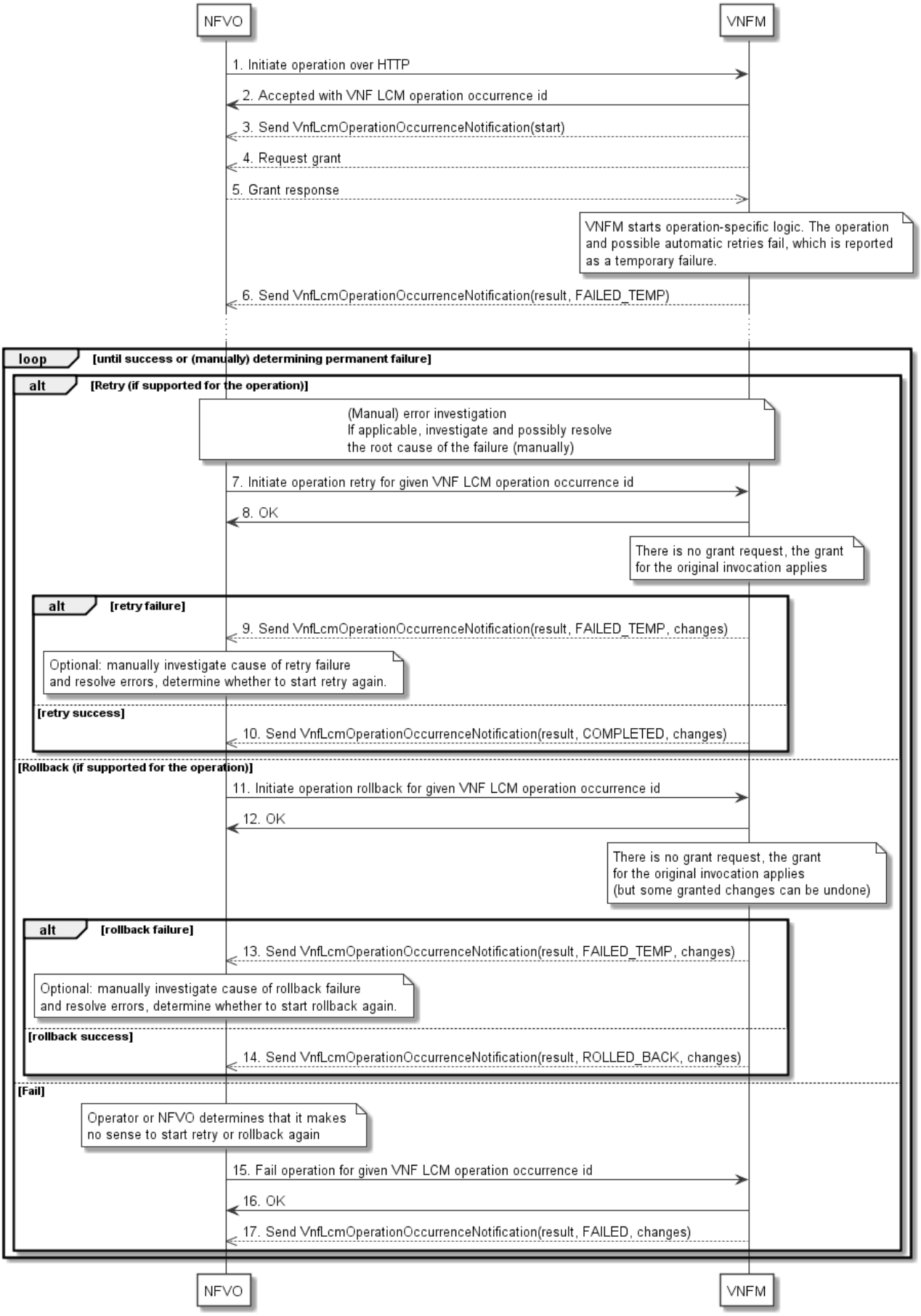

Figure 5.6.3.3-1: Handling failures during the actual execution of a VNF LCM operation

5.6.3.4 LCM operation cancellation

The cancellation of an LCM operation that is in PROCESSING or ROLLING_BACK state is handled like any other error that leads to stopping the execution of the VNF LCM workflow before it can be successfully completed. The VNF LCM operation transits into the FAILED_TEMP state which allows root cause analysis, possible fixing of the root cause, followed by retrying, rolling back, or finally failing of the operation.

The cancellation of an operation in STARTING state (i.e. until the Grant is received) transits the operation into the ROLLED_BACK state, as no changes to the resources or VNF instance have been performed.

5.7 Handling of security-sensitive attributes

The VNFD allows the VNF provider to declare certain VNF-specific attributes, such as additional parameters of VNF LCM operations or VNF configurable properties, as "sensitive" which means that their exposure can be a security risk. Attributes marked as "sensitive" shall be omitted in HTTP response bodies and in notifications in order to prevent their exposure. In case a change to a sensitive attribute is the only modification reported in a notification that notification shall still be sent, omitting the sensitive attribute.

6 VNF Performance Management interface

6.1 Description

This interface allows providing performance management (measurement results collection and notifications) related to VNFs. Performance information on a given VNF instance results from performance information of the virtualised resources that is collected from the VIM and mapped to this VNF instance. Collection and reporting of performance information is controlled by a PM job that groups details of performance collection and reporting information. Further, this interface allows API version information retrieval.

When new performance information is available, the API consumer is notified using the notification PerformanceInformationAvailableNotification.

The operations provided through this interface are:

- Create PM Job
- Query PM Job
- Delete PM Job
- Create Threshold
- Query Threshold
- Delete Threshold
- Notify

6.1a API version

For the VNF performance management interface version as specified in the present document, the MAJOR version field shall be 2, the MINOR version field shall be 10 and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [8] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v2".

6.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [8].

The string "vnfpm" shall be used to represent {apiName}. All resource URIs in the clauses below are defined relative to the above base URI.

Figure 6.2-1 shows the overall resource URI structure defined for the performance management API.

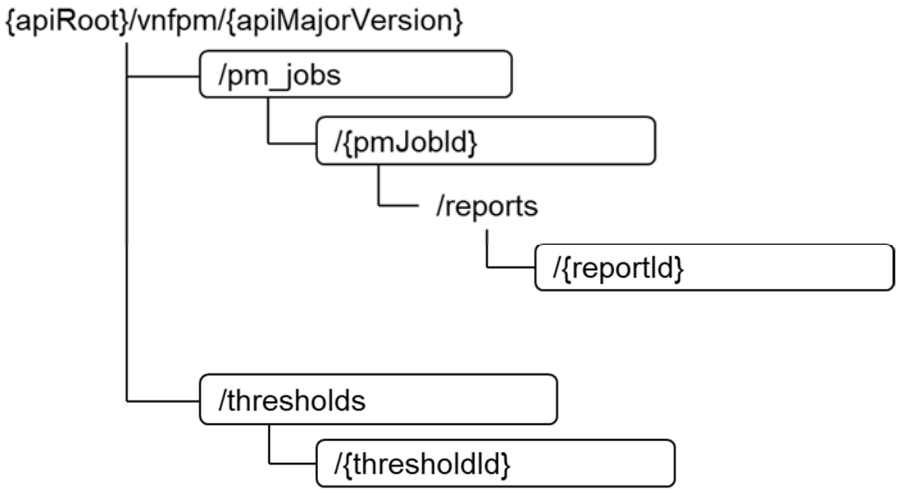

Figure 6.2-1: Resource URI structure of the VNF Performance Management interface

Table 6.2-1 lists the individual resources defined, and the applicable HTTP methods.

The VNFM shall support responding to requests for all HTTP methods on the resources in table 6.2-1 that are marked as "M" (mandatory) in the "Cat" column. The VNFM shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [8].

Table 6.2-1: Resources and methods overview of the VNF Performance Management interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
PM jobs	/pm_jobs	POST	M	Create a PM job
		GET	M	Query PM jobs
Individual PM job	/pm_jobs/{pmJobId}	GET	M	Read a single PM job
		PATCH	M	Update PM job callback
		DELETE	M	Delete a PM job
Individual performance report	/pm_jobs/{pmJobId}/reports/{reportId}	GET	M	Read an individual performance report
Thresholds	/thresholds	POST	M	Create a threshold
		GET	M	Query thresholds
Individual threshold	/thresholds/{thresholdId}	GET	M	Read a single threshold
		PATCH	M	Update threshold callback
		DELETE	M	Delete a threshold
Notification endpoint	(provided by API consumer)	POST	See note	Notify about PM related events
		GET	See note	Test the notification endpoint
NOTE: The VNFM shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the NFVO. If the NFVO supports invoking the POST method on the "PM jobs" or "Thresholds" resource towards the VNFM, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.				

6.3 Sequence diagrams (informative)

6.3.1 Flow of creating a PM job

This clause describes a sequence for creating a performance management job.

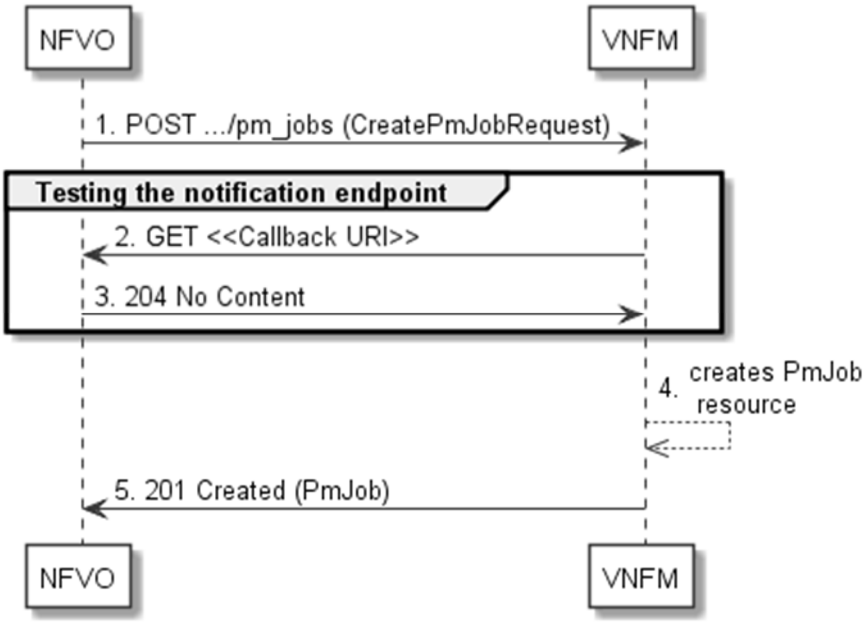

Figure 6.3.1-1: Flow of PM job creation

PM job creation, as illustrated in figure 6.3.1-1, consists of the following steps:

1) If the NFVO intends to create a PM job, it sends a POST request to the "PM jobs" resource, including one data structure of type "CreatePmJobRequest" in the payload body.

2) To test the notification endpoint that was registered by the NFVO during PM job creation, the VNFM sends a GET request to the notification endpoint URI.

3) The NFVO returns a "204 No Content" response to indicate success.

4) The VNFM creates a PM job instance.

5) The VNFM returns a "201 Created" response to the NFVO and includes in the payload body a representation of the PM job just created.

Error handling: In case of failure, including an invalid notification endpoint, appropriate error information is provided in the response.

6.3.1a Flow of updating the callback URI of a PM job

This clause describes a sequence for updating the callback URI in a PM job.

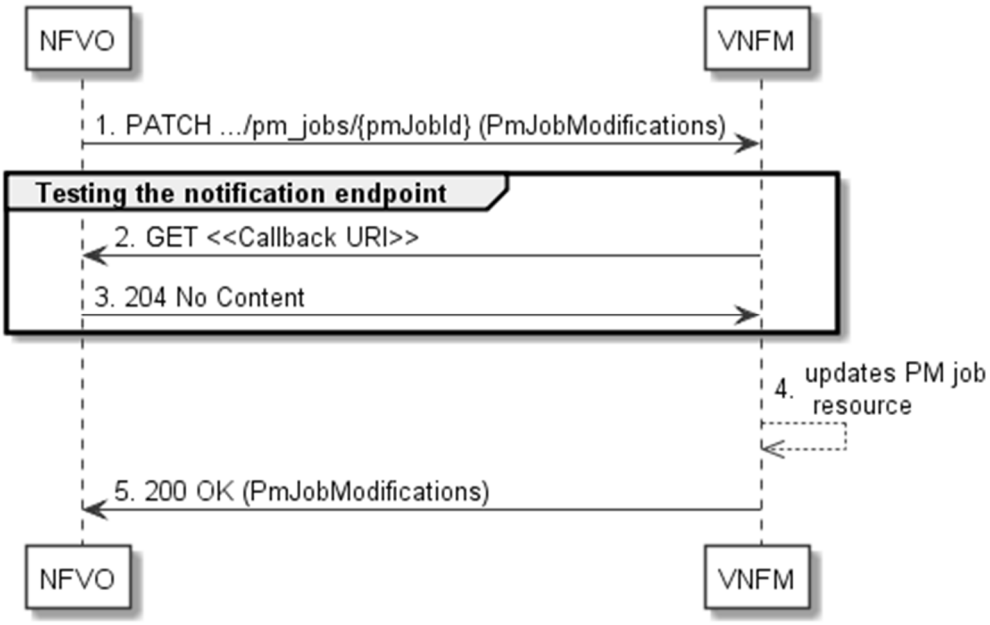

Figure 6.3.1a-1: Flow of PM job callback URI update

PM job callback URI update, as illustrated in figure 6.3.1a-1, consists of the following steps:

1) If the NFVO intends to update the callback URI in a PM job, it sends a PATCH request to the "Individual PM job" resource, including a data structure of type "PmJobModifications" in the payload body.

2) To test the notification endpoint that is addressed by the new callback URI, the VNFM sends a GET request to the notification endpoint URI.

3) The NFVO returns a "204 No Content" response to indicate success.

4) The VNFM updates the callback URI of the "Individual PM job" resource.

5) The VNFM returns a "200 OK " response to the NFVO and includes in the payload body a data structure of type "PmJobModifications" to indicate the performed modifications.

Error handling: In case of failure, including an invalid notification endpoint, appropriate error information is provided in the response.

6.3.2 Flow of querying/reading PM jobs

This clause describes a sequence for querying/reading performance management jobs.

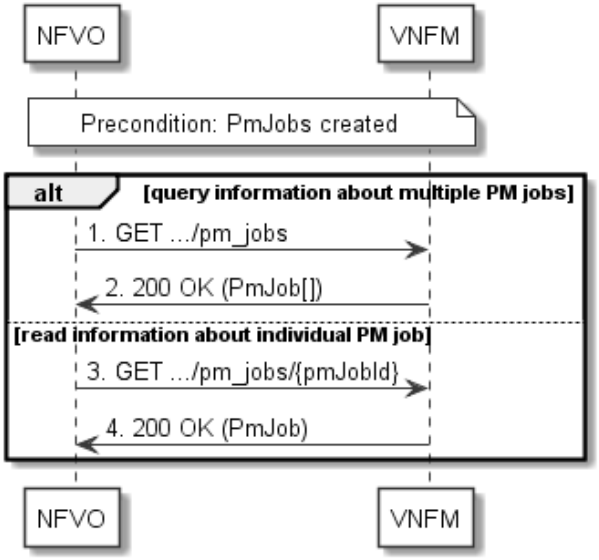

Figure 6.3.2-1: Flow of PM jobs query/read

PM jobs query/read, as illustrated in figure 6.3.2-1, consists of the following steps:

1) If the NFVO intends to query all PM jobs, it sends a GET request to the "PM jobs" resource.

2) The VNFM returns a "200 OK" response to the NFVO and includes zero or more data structures of type "PmJob" in the payload body.

3) If the NFVO intends to read information about a particular PM job, it sends a GET request to the "Individual PM job" resource, addressed by the appropriate PM job identifier in its resource URI.

4) The VNFM returns a "200 OK" response to the NFVO and includes one data structure of type "PmJob" in the payload body.

Error handling: In case of failure, appropriate error information is provided in the response.

6.3.3 Flow of deleting a PM job

This clause describes a sequence for deleting a performance management job.

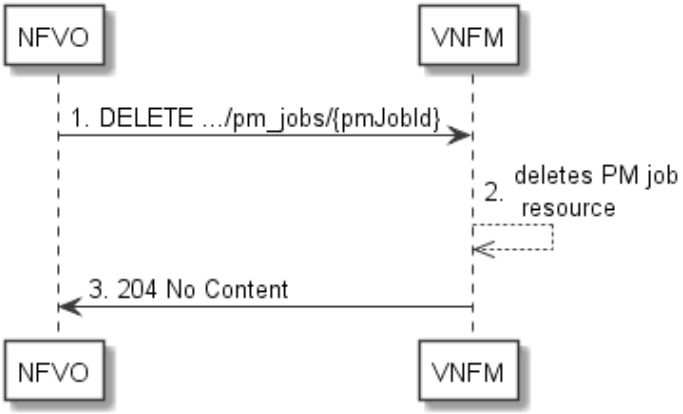

Figure 6.3.3-1: Flow of PM job deletion

PM job deletion, as illustrated in figure 6.3.3-1, consists of the following steps:

1) If the NFVO intends to delete a PM job, it sends a DELETE request to the "Individual PM job" resource, addressed by the appropriate PM job identifier in its resource URI.

2) The VNFM deletes the "Individual PM job" resource.

3) The VNFM returns a response with a "204 No Content" response code and an empty payload body to the NFVO.

Error handling: In case of failure, appropriate error information is provided in the response.

6.3.4 Flow of obtaining performance reports

This clause describes a sequence for obtaining performance reports.

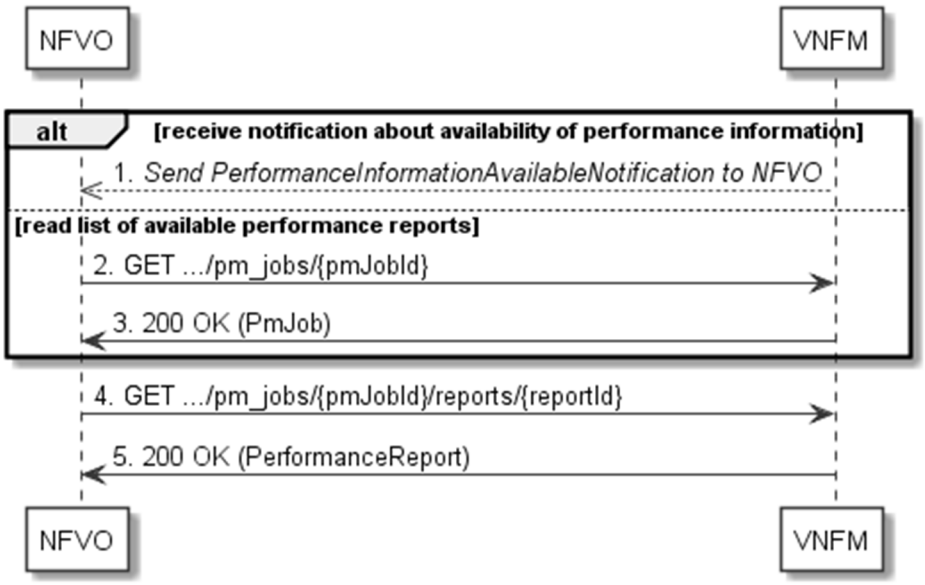

Figure 6.3.4-1: Flow of obtaining performance reports

Obtaining a performance report, as illustrated in figure 6.3.4-1, consists of the following steps:

1) The VNFM sends to the NFVO a PerformanceInformationAvailableNotification (see clause 6.3.9) that indicates the availability of a new performance report, including a link from which the report can be obtained.

2) Alternatively, the NFVO sends a GET request to the "Individual PM job" resource, to obtain a representation of the resource including information about performance reports that are available for this PM job, including their URIs.

3) In that case, the VNFM returns a "200 OK" response to the NFVO and includes a data structure of type "PmJob" in the payload body.

4) The NFVO sends to the VNFM a GET request to the URI obtained either in step 1) or step 3), in order to read an "Individual performance report" resource.

5) The VNFM returns a "200 OK" response to the NFVO and includes a data structure of type "PerformanceReport" in the payload body.

6.3.5 Flow of creating a threshold

This clause describes a sequence for creating a performance management threshold.

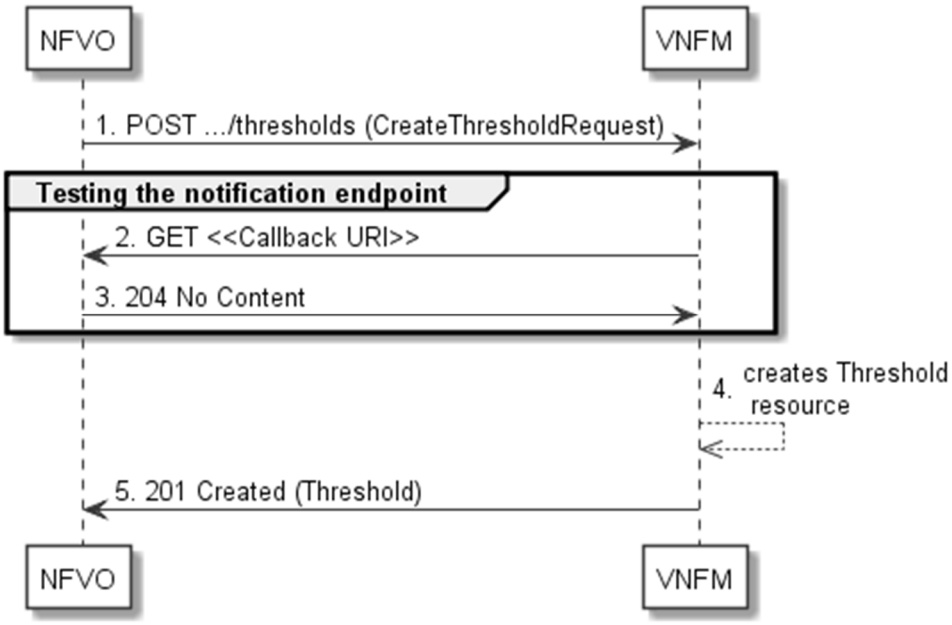

Figure 6.3.5-1: Flow of threshold creation

Threshold creation, as illustrated in figure 6.3.5-1, consists of the following steps:

1) If the NFVO intends to create a threshold, it sends a POST request to the "Thresholds" resource, including a data structure of type "CreateThresholdRequest" in the payload body.

2) To test the notification endpoint that was registered by the NFVO during threshold creation, the VNFM sends a GET request to the notification endpoint URI.

3) The NFVO returns a "204 No Content" response to indicate success.

4) The VNFM creates a threshold instance.

5) The VNFM returns a "201 Created" response to the NFVO and includes in the payload body a representation of the threshold just created.

Error handling: In case of failure, including an invalid notification endpoint, appropriate error information is provided in the response.

6.3.5a Flow of updating the callback URI of a threshold

This clause describes a sequence for updating the callback URI in a performance management threshold.

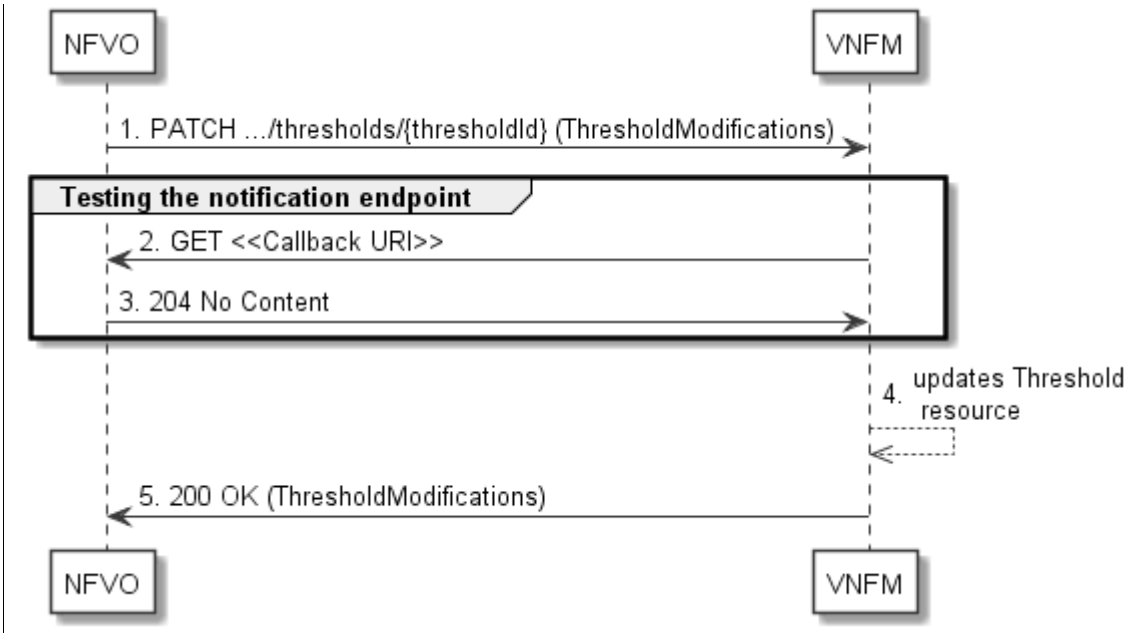

Figure 6.3.5a-1: Flow of threshold callback URI update

Threshold callback URI update, as illustrated in figure 6.3.5a-1, consists of the following steps:

1) If the NFVO intends to update the callback URI in a threshold, it sends a PATCH request to the "Individual threshold" resource, including a data structure of type "ThresholdModifications" in the payload body.

2) To test the notification endpoint that is addressed by the new callback URI, the VNFM sends a GET request to the notification endpoint URI.

3) The NFVO returns a "204 No Content" response to indicate success.

4) The VNFM updates the callback URI of the "Individual threshold" resource.

5) The VNFM returns a "200 OK " response to the NFVO and includes in the payload body a data structure of type "ThresholdModifications" to indicate the performed modifications.

Error handling: In case of failure, including an invalid notification endpoint, appropriate error information is provided in the response.

6.3.6 Flow of querying/reading thresholds

This clause describes a sequence for querying/reading performance management thresholds.

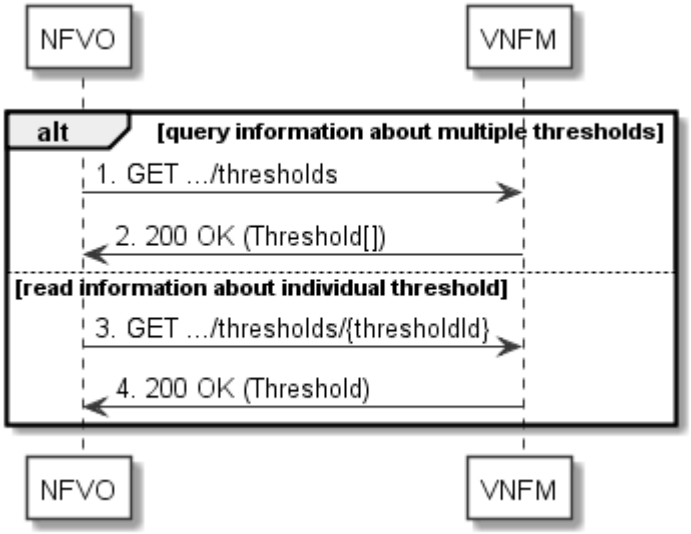

Figure 6.3.6-1: Flow of thresholds query/read

Threshold query/read, as illustrated in figure 6.3.6-1, consists of the following steps:

1) If the NFVO intends to query all thresholds, it sends a GET request to the "Thresholds" resource.

2) The VNFM returns a "200 OK" response to the NFVO and includes zero or more data structures of type "Threshold" in the payload body.

3) If the NFVO intends to read information about a particular threshold, it sends a GET request to the "Individual threshold" resource addressed by the appropriate threshold identifier in its resource URI.

4) The VNFM returns a "200 OK" response to the NFVO and includes a data structure of type "Threshold" in the payload body.

Error handling: In case of failure, appropriate error information is provided in the response.

6.3.7 Flow of deleting thresholds

This clause describes a sequence for deleting performance management thresholds.

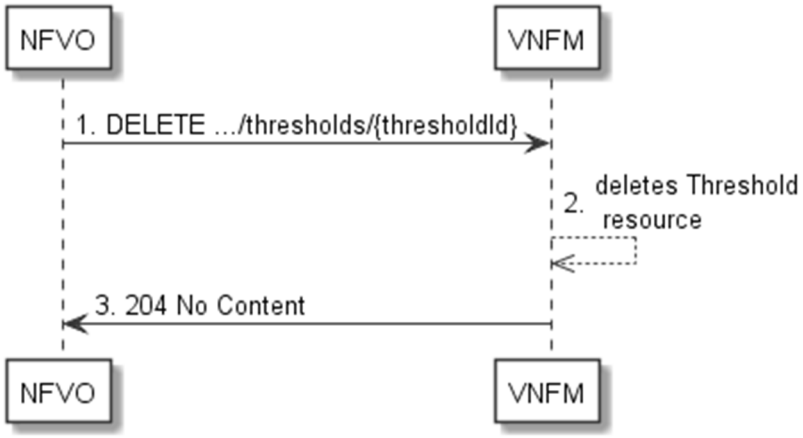

Figure 6.3.7-1: Flow of threshold deletion

Threshold deletion, as illustrated in figure 6.3.7-1, consists of the following steps:

1) If the NFVO intends to delete a particular threshold, it sends a DELETE request to the "Individual threshold" resource, addressed by the appropriate threshold identifier in its resource URI.

2) The VNFM deletes the "Individual threshold" resource.

3) The VNFM returns a "204 No Content" response code to the NFVO. The response body shall be empty.

Error handling: In case of failure, appropriate error information is provided in the response.

6.3.8 Void

6.3.9 Flow of sending notifications

This clause describes the procedure for sending notifications related to VNF performance management.

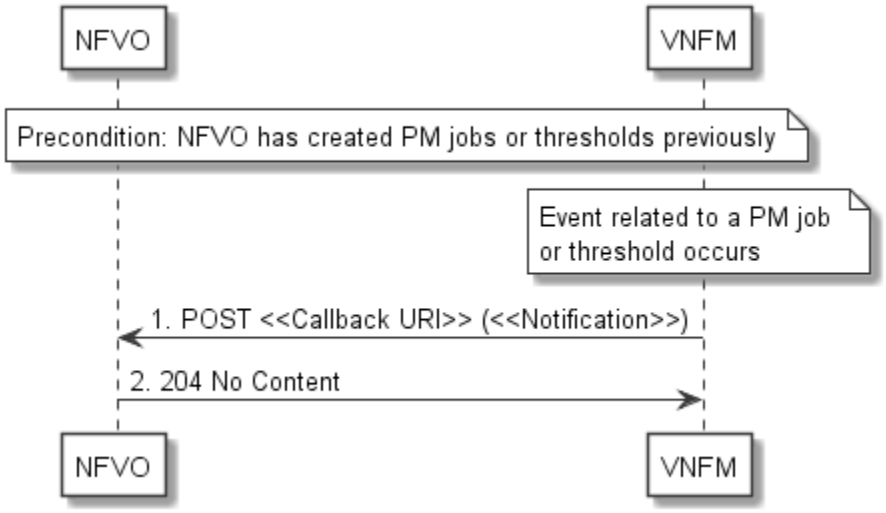

Figure 6.3.9-1: Flow of sending notifications

Precondition: The NFVO has previously created thresholds and/or PM jobs which trigger notifications related to VNF performance management.

The procedure consists of the following steps as illustrated in figure 6.3.9-1:

1) If an event occurs that indicates a threshold crossing or availability of performance information in a PM job, the VNFM generates a notification that includes information about the event and sends it in the body of a POST request to the URI which the NFVO has registered as part of creating the threshold or PM job. The variable <<Notification>> in the flow is a placeholder for the different types of notifications that can be sent by this API.

2) The NFVO acknowledges the successful delivery of the notification by returning a "204 No Content" response.

Error handling: If the VNFM does not receive the "204 No Content" response from the NFVO, it can retry sending the notification.

6.4 Resources

6.4.1 Introduction

This clause defines all the resources and methods provided by the performance management interface.

6.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [8] are part of the VNF performance management interface.

6.4.2 Resource: PM jobs

6.4.2.1 Description

This resource represents PM jobs. The API consumer can use this resource to create and query PM jobs.

6.4.2.2 Resource definition

The resource URI is:

{apiRoot}/vnfpm/{apiMajorVersion}/pm_jobs

This resource shall support the resource URI variables defined in table 6.4.2.2-1.

Table 6.4.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 6.1a.

6.4.2.3 Resource methods

6.4.2.3.1 POST

The POST method creates a PM job.

This method shall follow the provisions specified in tables 6.4.2.3.1-1 and 6.4.2.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual PM job" resource as defined in clause 6.4.3 shall have been created.

Table 6.4.2.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.2.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	CreatePmJobRequest	1	PM job creation request	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	PmJob	1	201 Created	Shall be returned when the PM job has been created successfully. The response body shall contain a representation of the created "Individual PM job" resource, as defined in clause 6.5.2.7. The HTTP response shall include a "Location" HTTP header that points to the created "Individual PM job" resource.
	ProblemDetails	1	422 Unprocessable Entity	Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this resource, the response code 422 shall also be returned if the VNFM has tested the Notification endpoint as described in clause 6.4.9.3.2 and the test has failed. In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

6.4.2.3.2 GET

The API consumer can use this method to retrieve information about PM jobs.

This method shall follow the provisions specified in tables 6.4.2.3.2-1 and 6.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.2.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [8]. The VNFM shall support receiving this parameter as part of the URI query string. The NFVO may supply this parameter. All attribute names that appear in the PmJob and in data types referenced from it shall be supported by the VNFM in the filter expression.
all_fields	0..1	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The VNFM shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The VNFM should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The VNFM should support this parameter.

Name	Cardinality	Description
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The VNFM shall support this parameter. The following attributes shall be excluded from the PmJob structure in the response body if this parameter is provided, or none of the parameters "all_fields," "fields", "exclude_fields", "exclude_default" are provided: - reports
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource.

Table 6.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	PmJob	0..N	200 OK	Shall be returned when information about zero or more PM jobs has been queried successfully. The response body shall contain in an array the representations of zero or more PM jobs, as defined in clause 6.5.2.7. If the "filter" URI parameter or one of the "all_fields", "fields" (if supported), "exclude_fields" (if supported) or "exclude_default" URI parameters was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clauses 5.2.2 and 5.3.2 of ETSI GS NFV-SOL 013 [8], respectively. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute selector. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

6.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

6.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

6.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

6.4.3 Resource: Individual PM job

6.4.3.1 Description

This resource represents an individual PM job. The API consumer can use this resource to delete and read the underlying PM job.

6.4.3.2 Resource definition

The resource URI is:

{apiRoot}/vnfpm/{apiMajorVersion}/pm_jobs/{pmJobId}

This resource shall support the resource URI variables defined in table 6.4.3.2-1.

Table 6.4.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 6.1a.
pmJobId	Identifier of the PM job. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual PM job" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

6.4.3.3 Resource methods

6.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

6.4.3.3.2 GET

The API consumer can use this method for reading an individual PM job.

This method shall follow the provisions specified in tables 6.4.3.3.2-1 and 6.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.3.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	PmJob	1	200 OK	Shall be returned when information about an individual PM job has been read successfully. The response body shall contain a representation of the "Individual PM job" resource, as defined in clause 6.5.2.7.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

6.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

6.4.3.3.4 PATCH

This method allows to modify an "Individual PM job" resource.

This method shall follow the provisions specified in tables 6.4.3.3.4-1 and 6.4.3.3.4-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.3.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.3.3.4-2: Details of the PATCH request/response on this resource

	Data type	Cardinality		
Request body	PmJobModifications	1	Parameters for the PM job modification. The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [5].	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	PmJobModifications	1	200 OK	Shall be returned when the request has been processed successfully. The response body shall contain a data structure of type "PmJobModifications".
	ProblemDetails	0..1	412 Precondition failed	Shall be returned upon the following error: A precondition given in an HTTP request header is not fulfilled. Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity. The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	1	422 Unprocessable Entity	Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this resource, the response code 422 shall also be returned if the VNFM has tested the Notification endpoint as described in clause 6.4.9.3.2 and the test has failed. In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

6.4.3.3.5 DELETE

This method terminates an individual PM job.

This method shall follow the provisions specified in tables 6.4.3.3.5-1 and 6.4.3.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual PM job" resource shall not exist any longer.

Table 6.4.3.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.3.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned when the PM job has been deleted successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

6.4.4 Resource: Individual performance report

6.4.4.1 Description

This resource represents an individual performance report that has been collected by a PM job. The API consumer can use this resource to read the performance report. The URI of this report can be obtained from a PerformanceInformationAvailableNotification (see clause 6.5.2.5) or from the representation of the "Individual PM job" resource.

It is determined by means outside the scope of the present document, such as configuration or policy, how long an individual performance report is available.

6.4.4.2 Resource definition

The resource URI is:

{apiRoot}/vnfpm/{apiMajorVersion}/pm_jobs/{pmJobId}/reports/{reportId}

This resource shall support the resource URI variables defined in table 6.4.4.2-1.

Table 6.4.4.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 6.1a.
pmJobId	Identifier of the PM job.
reportId	Identifier of the performance report.

6.4.4.3 Resource methods

6.4.4.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

6.4.4.3.2 GET

The API consumer can use this method for reading an individual performance report.

This method shall follow the provisions specified in tables 6.4.4.3.2-1 and 6.4.4.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.4.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.4.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	PerformanceReport	1	200 OK	Shall be returned when information of an individual performance report has been read successfully. The response body shall contain a representation of the "Individual performance report" resource, as defined in clause 6.5.2.10.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

6.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

6.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

6.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

6.4.5 Resource: Thresholds

6.4.5.1 Description

This resource represents thresholds. The API consumer can use this resource to create and query thresholds.

6.4.5.2 Resource definition

The resource URI is:

{apiRoot}/vnfpm/{apiMajorVersion}/thresholds

This resource shall support the resource URI variables defined in table 6.4.5.2-1.

Table 6.4.5.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 6.1a.

6.4.5.3 Resource methods

6.4.5.3.1 POST

The POST method can be used by the API consumer to create a threshold.

This method shall follow the provisions specified in tables 6.4.5.3.1-1 and 6.4.5.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual threshold" resource as defined in clause 6.4.6 shall have been created.

Table 6.4.5.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.5.3.1-2: Details of the POST request/response on this resource

Request body	**Data type**	**Cardinality**	**Description**	
	CreateThresholdRequest	1	Request parameters to create a new "Individual threshold" resource.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	Threshold	1	201 Created	Shall be returned when a threshold has been created successfully. The response body shall contain a representation of the created "Individual threshold" resource, as defined in clause 6.5.2.9. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created resource.
	ProblemDetails	1	422 Unprocessable Entity	Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this resource, the response code 422 shall also be returned if the VNFM has tested the Notification endpoint as described in clause 6.4.9.3.2 and the test has failed. In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

6.4.5.3.2 GET

The API consumer can use this method to query information about thresholds.

This method shall follow the provisions specified in tables 6.4.5.3.2-1 and 6.4.5.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.5.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [8]. The VNFM shall support receiving this parameter as part of the URI query string. The NFVO may supply this parameter. All attribute names that appear in the Thresholds data type and in data types referenced from it shall be supported by the VNFM in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource.
NOTE: There are no attribute selectors defined for this resource as the threshold attributes with cardinality 0..1 or 0..N are not structurally complex in nature.		

Table 6.4.5.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	Threshold	0..N	200 OK	Shall be returned when information about zero or more thresholds has been queried successfully. If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [8]. The response body shall contain in an array the representations of zero or more thresholds, as defined in clause 6.5.2.9. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

6.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

6.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

6.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

6.4.6 Resource: Individual threshold

6.4.6.1 Description

This resource represents an individual threshold.

6.4.6.2 Resource definition

The resource URI is:

{apiRoot}/vnfpm/{apiMajorVersion}/thresholds/{thresholdId}

This resource shall support the resource URI variables defined in table 6.4.6.2-1.

Table 6.4.6.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 6.1a.
thresholdId	Identifier of the threshold. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual threshold" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

6.4.6.3 Resource methods

6.4.6.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

6.4.6.3.2 GET

The API consumer can use this method for reading an individual threshold.

This method shall follow the provisions specified in tables 6.4.6.3.2-1 and 6.4.6.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.6.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.6.3.2-2: Details of the GET request/response on this resource

Request body	**Data type**	**Cardinality**	**Description**	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	Threshold	1	200 OK	Shall be returned when information about an individual threshold has been read successfully. The response body shall contain a representation of the threshold, as defined in clause 6.5.2.9.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

6.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

6.4.6.3.4 PATCH

This method allows to modify an "Individual threshold" resource.

This method shall follow the provisions specified in tables 6.4.6.3.4-1 and 6.4.6.3.4-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.6.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.6.3.4-2: Details of the PATCH request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	ThresholdModifications	1		Parameters for the threshold modification. The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [5].
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	ThresholdModifications	1	200 OK	Shall be returned when the request has been processed successfully. The response body shall contain a data structure of type "ThresholdModifications".
	ProblemDetails	0..1	412 Precondition failed	Shall be returned upon the following error: A precondition given in an HTTP request header is not fulfilled. Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity. The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	ProblemDetails	1	422 Unprocessable Entity	Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this resource, the response code 422 shall also be returned if the VNFM has tested the Notification endpoint as described in clause 6.4.9.3.2 and the test has failed. In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

6.4.6.3.5 DELETE

This method allows to delete a threshold.

This method shall follow the provisions specified in tables 6.4.6.3.5-1 and 6.4.6.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual threshold" resource shall not exist any longer.

Table 6.4.6.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.6.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned when the threshold has been deleted successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

6.4.7 Void

6.4.8 Void

6.4.9 Resource: Notification endpoint

6.4.9.1 Description

This resource represents a notification endpoint for VNF performance management.

The API producer can use this resource to send notifications related to performance management events to an API consumer which has provided the URI of this resource during the PM job or threshold creation process.

6.4.9.2 Resource definition

The resource URI is provided by the API consumer when creating the PM job or threshold.

This resource shall support the resource URI variables defined in table 6.4.9.2-1.

Table 6.4.9.2-1: Resource URI variables for this resource

Name	Definition
none supported	

6.4.9.3 Resource methods

6.4.9.3.1 POST

The POST method delivers a notification regarding a performance management event from API producer to an API consumer. The API consumer shall have previously created an "Individual PM job" resource or "Individual threshold" resource.

This method shall follow the provisions specified in tables 6.4.9.3.1-1 and 6.4.9.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.9.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.9.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Description	
Request body	PerformanceInformationAvailableNotification	1	Notification about performance information availability	
	ThresholdCrossedNotification	1	Notification about threshold crossing	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned when the notification has been delivered successfully.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

6.4.9.3.2 GET

The GET method allows the API producer to test the notification endpoint that is provided by the API consumer, e.g. during creation of the PM job or threshold resource.

This method shall follow the provisions specified in tables 6.4.9.3.2-1 and 6.4.9.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.9.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.9.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned to indicate that the notification endpoint has been tested successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

6.4.9.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

6.4.9.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

6.4.9.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

6.5 Data Model

6.5.1 Introduction

This clause defines the request and response data structures of the VNF Performance Management interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error and may choose to ignore them.

6.5.2 Resource and notification data types

6.5.2.1 Introduction

This clause defines the data structures to be used in resource representations and notifications.

6.5.2.2 Void

6.5.2.3 Void

6.5.2.4 Type: ThresholdCrossedNotification

This type represents a notification that is sent when a threshold has been crossed. It shall comply with the provisions defined in table 6.5.2.4-1.

NOTE: The timing of sending this notification is determined by the capability of the producing entity to evaluate the threshold crossing condition.

The notification shall be triggered by the VNFM when a threshold has been crossed.

Table 6.5.2.4-1: Definition of the ThresholdCrossedNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "ThresholdCrossedNotification" for this notification type.
timeStamp	DateTime	1	Date and time of the generation of the notification.
thresholdId	Identifier	1	Identifier of the threshold which has been crossed.
crossingDirection	CrossingDirectionType	1	An indication of whether the threshold was crossed in upward or downward direction.
objectType	String	1	Type of the measured object. The applicable measured object type for a measurement is defined in clause 7.2 of ETSI GS NFV-IFA 027 [6].
objectInstanceId	Identifier	1	Identifier of the measured object instance as per clause 6.2 of ETSI GS NFV-IFA 027 [6].

Attribute name	Data type	Cardinality	Description
subObjectInstanceId	IdentifierInVnf	0..1	Identifier of the sub-object of the measured object to which the measurement applies. Shall be present if this is required in clause 6.2 of ETSI GS NFV-IFA 027 [6] for the related measured object type. See note.
performanceMetric	String	1	Performance metric associated with the threshold. This attribute shall contain the related "Measurement Name" value as defined in clause 7.2 of ETSI GS NFV-IFA 027 [6].
performanceValue	(any type)	1	Value of the metric that resulted in threshold crossing. The type of this attribute shall correspond to the related "Measurement Unit" as defined in clause 7.2 of ETSI GS NFV-IFA 027 [6].
context	KeyValuePairs	0..1	Measurement context information related to the measured value. The set of applicable keys is defined per measurement in the related "Measurement Context" in clause 7.2 of ETSI GS NFV-IFA 027 [6].
_links	Structure (inlined)	1	Links to resources related to this notification
>objectInstance	NotificationLink	0..1	Link to the resource representing the measured object instance to which the notified change applies. Shall be present if the measured object instance information is accessible as a resource.
>threshold	NotificationLink	1	Link to the resource that represents the threshold that was crossed.
NOTE: The sub-object allows to structure the measured object, but is not to be confused with sub-counters which allow to structure the measurement.			

6.5.2.5 Type: PerformanceInformationAvailableNotification

This notification informs the receiver that performance information is available. It shall comply with the provisions defined in table 6.5.2.5-1.

The notification shall be triggered by the VNFM when new performance information collected by a PM job is available. The periodicity of triggering this notification is influenced by the "reportingPeriod" attribute in the "PmJobCriteria" data structure as defined in clause 6.5.3.3.

Table 6.5.2.5-1: Definition of the PerformanceInformationAvailableNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "PerformanceInformationAvailableNotification" for this notification type.
timeStamp	DateTime	1	Date and time of the generation of the notification.
pmJobId	Identifier	1	Identifier of the PM job for which performance information is available.
objectType	String	1	Type of the measured object. The applicable measured object type for a measurement is defined in clause 7.2 of ETSI GS NFV-IFA 027 [6].
objectInstanceId	Identifier	1	Identifier of the measured object instance as per clause 6.2 of ETSI GS NFV-IFA 027 [6].

Attribute name	Data type	Cardinality	Description
subObjectInstanceIds	IdentifierInVnf	0..N	Identifiers of the sub-object instances of the measured object instance for which the measurements have been taken. Shall be present if the related PM job has been set up to measure only a subset of all sub-object instances of the measured object instance and a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [6] for the related measured object type. Shall be absent otherwise.
_links	Structure (inlined)	1	Links to resources related to this notification.
>objectInstance	NotificationLink	0..1	Link to the resource representing the measured object instance to which the notification applies. Shall be present if the measured object instance information is accessible as a resource.
>pmJob	NotificationLink	1	Link to the resource that represents the PM job for which performance information is available.
>performanceReport	NotificationLink	1	Link from which the available performance information of data type "PerformanceReport" (see clause 6.5.2.10) can be obtained. This link should point to an "Individual performance report" resource as defined in clause 6.4.4.

6.5.2.6 Type: CreatePmJobRequest

This type represents a request to create a PM job. It shall comply with the provisions defined in table 6.5.2.6-1.

Table 6.5.2.6-1: Definition of the CreatePmJobRequest data type

Attribute name	Data type	Cardinality	Description
objectType	String	1	Type of the measured object. The applicable measured object type for a measurement is defined in clause 7.2 of ETSI GS NFV-IFA 027 [6].
objectInstanceIds	Identifier	1..N	Identifiers of the measured object instances for which performance information is requested to be collected.
subObjectInstanceIds	IdentifierInVnf	0..N	Identifiers of the sub-object instances of the measured object instance for which performance information is requested to be collected. May be present if a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [6] for the related measured object type. f this attribute is present, the cardinality of the 'objectInstanceIds" attribute shall be 1. If this attribute is absent and a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [6] for the related measured object type, measurements will be taken for all sub-object instances of the measured object instance.
criteria	PmJobCriteria	1	Criteria of the collection of performance information.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this PM job, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [8]. This attribute shall only be present if the API consumer requires authorization of notifications.

6.5.2.7 Type: PmJob

This type represents a PM job. It shall comply with the provisions defined in table 6.5.2.7-1.

Table 6.5.2.7-1: Definition of the PmJob data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this PM job.
objectType	String	1	Type of the measured object. The applicable measured object type for a measurement is defined in clause 7.2 of ETSI GS NFV-IFA 027 [6].
objectInstanceIds	Identifier	1..N	Identifiers of the measured object instances for which performance information is collected.
subObjectInstanceIds	IdentifierInVnf	0..N	Identifiers of the sub-object instances of the measured object instance for which performance information is requested to be collected. May be present if a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [6] for the related measured object type. If this attribute is present, the cardinality of the "objectInstanceIds" attribute shall be 1. If this attribute is absent and a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [6] for the related measured object type, measurements will be taken for all sub-object instances of the measured object instance.
criteria	PmJobCriteria	1	Criteria of the collection of performance information.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
reports	Structure (inlined)	0..N	Information about available reports collected by this PM job.
>href	Uri	1	The URI where the report can be obtained.
>readyTime	DateTime	1	The time when the report was made available.
>expiryTime	DateTime	0..1	The time when the report will expire.
>fileSize	UnsigendInt	0..1	The size of the report file in bytes, if known.
_links	Structure (inlined)	1	Links for this resource.
>self	Link	1	URI of this resource.
>objects	Link	0..N	Links to resources representing the measured object instances for which performance information is collected. Shall be present if the measured object instance information is accessible as a resource.

6.5.2.8 Type: CreateThresholdRequest

This type represents a request to create a threshold. It shall comply with the provisions defined in table 6.5.2.8-1.

Table 6.5.2.8-1: Definition of the CreateThresholdRequest data type

Attribute name	Data type	Cardinality	Description
objectType	String	1	Type of the measured object. The applicable measured object type for a measurement is defined in clause 7.2 of ETSI GS NFV-IFA 027 [6].
objectInstanceId	Identifier	1	Identifier of the measured object instance associated with this threshold.

Attribute name	Data type	Cardinality	Description
subObjectInstanceIds	IdentifierInVnf	0..N	Identifiers of the sub-object instances of the measured object instance associated with this threshold. May be present if a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [6] for the related measured object type. If this attribute is absent and a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [6] for the measured object type, measurements will be taken for all sub-object instances of the measured object instance.
criteria	ThresholdCriteria	1	Criteria that define this threshold.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this threshold, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [8]. This attribute shall only be present if the API consumer requires authorization of notifications.

6.5.2.9 Type: Threshold

This type represents a threshold. It shall comply with the provisions defined in table 6.5.2.9-1.

Table 6.5.2.9-1: Definition of the Threshold data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this threshold resource.
objectType	String	1	Type of the measured object. The applicable measured object type for a measurement is defined in clause 7.2 of ETSI GS NFV-IFA 027 [6].
objectInstanceId	Identifier	1	Identifier of the measured object instance associated with the threshold.
subObjectInstanceIds	IdentifierInVnf	0..N	Identifiers of the sub-object instances of the measured object instance associated with the threshold. May be present if a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [6] for the related measurement type. If this attribute is absent and a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [6] for the related measured object type, measurements will be taken for all sub-object instances of the measured object instance.
criteria	ThresholdCriteria	1	Criteria that define this threshold.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
_links	Structure (inlined)	1	Links for this resource.
>self	Link	1	URI of this resource.
>object	Link	0..1	Link to a resource representing the measured object instance for which performance information is collected. Shall be present if the measured object instance information is accessible as a resource.

6.5.2.10 Type: PerformanceReport

This type defines the format of a performance report provided by the VNFM to the NFVO as a result of collecting performance information as part of a PM job. The type shall comply with the provisions defined in table 6.5.2.10-1.

Table 6.5.2.10-1: Definition of the PerformanceReport data type

Attribute name	Data type	Cardinality	Description
entries	Structure (inlined)	1..N	List of performance information entries. Each performance report entry is for a given metric of a given object (i.e. VNF instance), but can include multiple collected values.
>objectType	String	1	Type of the measured object. The applicable measured object type for a measurement is defined in clause 7.2 of ETSI GS NFV-IFA 027 [6].
>objectInstanceId	Identifier	1	Identifier of the measured object instance for which the performance metric is reported.
>subObjectInstanceId	IdentifierInVnf	0..1	Identifier of the sub-object instance of the measured object instance for which the performance metric is reported. Shall be present if this is required in clause 6.2 of ETSI GS NFV-IFA 027 [6] for the related measured object type. See note.
>performanceMetric	String	1	Name of the metric collected. This attribute shall contain the related "Measurement Name" value as defined in clause 7.2 of ETSI GS NFV-IFA 027 [6].
>performanceValues	Structure (inlined)	1..N	List of performance values with associated timestamp.
>>timeStamp	DateTime	1	Time stamp indicating when the data has been collected.
>>value	(any type)	1	Value of the metric collected. The type of this attribute shall correspond to the related "Measurement Unit" as defined in clause 7.2 of ETSI GS NFV-IFA 027 [6].
>>context	KeyValuePairs	0..1	Measurement context information related to the measured value. The set of applicable keys is defined per measurement in the related "Measurement Context" in clause 7.2 of ETSI GS NFV-IFA 027 [6].
NOTE: The sub-object allows to structure the measured object but is not to be confused with sub-counters which allow to structure the measurement value. EXAMPLE: Measured object: VnfInstanceXYZ Sub-object: VnfcInstance1 Measurement: vCPU_utilization Sub-counters: vCPU utilization of each of the vCPUs of VnfcInstance1 (vCPU utilization.vCPU1, vCPU_utilization.vCPU2, etc.).			

6.5.2.11 Type: ThresholdModifications

This type represents modifications to a threshold. It shall comply with the provisions defined in table 6.5.2.11-1.

Table 6.5.2.11-1: Definition of the ThresholdModifications data type

Attribute name	Data type	Cardinality	Description
callbackUri	Uri	0..1	New value of the "callbackUri" attribute. The value "null" is not permitted. See note.
authentication	SubscriptionAuthentication	0..1	New value of the "authentication" attribute, or "null" to remove the attribute. If present in a request body, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [5]). This attribute shall not be present in response bodies. See note.
NOTE: At least one of the attributes defined in this type shall be present in request bodies.			

6.5.2.12 Type: PmJobModifications

This type represents modifications to a PM job. It shall comply with the provisions defined in table 6.5.2.12-1.

Table 6.5.2.12-1: Definition of the PmJobModifications data type

Attribute name	Data type	Cardinality	Description
callbackUri	Uri	0..1	New value of the "callbackUri" attribute. The value "null" is not permitted. See note.
authentication	SubscriptionAuthentication	0..1	New value of the "authentication" attribute, or "null" to remove the attribute. If present in a request body, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [5]). This attribute shall not be present in response bodies. See note.
NOTE: At least one of the attributes defined in this type shall be present in request bodies.			

6.5.3 Referenced structured data types

6.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but can neither be resource representations nor bound to any subscribe/notify mechanism.

6.5.3.2 Void

6.5.3.3 Type: PmJobCriteria

This type represents collection criteria for PM jobs. It shall comply with the provisions defined in table 6.5.3.3-1.

Table 6.5.3.3-1: Definition of the PmJobCriteria data type

Attribute name	Data type	Cardinality	Description
performanceMetric	String	0..N	This defines the types of performance metrics for the specified object instances. Valid values are specified as "Measurement Name" values in clause 7.2 of ETSI GS NFV-IFA 027 [6]. At least one of the two attributes (performance metric or group) shall be present.
performanceMetricGroup	String	0..N	Group of performance metrics. A metric group is a pre-defined list of metrics, known to the API producer that it can decompose to individual metrics. Valid values are specified as "Measurement Group" values in clause 7.2 of ETSI GS NFV-IFA 027 [6]. At least one of the two attributes (performance metric or group) shall be present.
collectionPeriod	UnsignedInt	1	Specifies the periodicity at which the API producer will collect performance information. The unit shall be seconds. See notes 1 and 2.
reportingPeriod	UnsignedInt	1	Specifies the periodicity at which the API producer will report to the API consumer. about performance information. The unit shall be seconds. See notes 1 and 2.

Attribute name	Data type	Cardinality	Description
reportingBoundary	DateTime	0..1	Identifies a time boundary after which the reporting will stop. The boundary shall allow a single reporting as well as periodic reporting up to the boundary.
NOTE 1: At the end of each reportingPeriod, the API producer will inform the API consumer about availability of the performance data collected for each completed collection period during this reportingPeriod. The reportingPeriod should be equal to or a multiple of the collectionPeriod. In the latter case, the performance data for the collection periods within one reporting period are reported together. NOTE 2: In particular when choosing short collection and reporting periods, the number of PM jobs that can be supported depends on the capability of the producing entity.			

6.5.3.4 Type: ThresholdCriteria

This type represents criteria that define a threshold. It shall comply with the provisions defined in table 6.5.3.4-1.

Table 6.5.3.4-1: Definition of the ThresholdCriteria data type

Attribute name	Data type	Cardinality	Description
performanceMetric	String	1	Defines the performance metric associated with the threshold. Valid values are specified as "Measurement Name" values in clause 7.2 of ETSI GS NFV-IFA 027 [6].
thresholdType	Enum (inlined)	1	Type of threshold. This attribute determines which other attributes are present in the data structure. Permitted values: - SIMPLE: Single-valued static threshold. See note 1.
simpleThresholdDetails	Structure (inlined)	0..1	Details of a simple threshold. Shall be present if thresholdType="SIMPLE".
>thresholdValue	Number	1	The threshold value. Shall be represented as a floating point number.
>hysteresis	Number	1	The hysteresis of the threshold. Shall be represented as a non-negative floating point number. A notification with crossing direction "UP" will be generated if the measured value reaches or exceeds "thresholdValue" + "hysteresis". A notification with crossing direction "DOWN" will be generated if the measured value reaches or undercuts "thresholdValue" - "hysteresis". See note 2.
NOTE 1: In the present document, simple thresholds are defined. The definition of additional threshold types is left for future specification. NOTE 2: The hysteresis is defined to prevent storms of threshold crossing notifications. When processing a request to create a threshold, implementations should enforce a suitable minimum value for this attribute (e.g. override the value or reject the request).			

6.5.4 Referenced simple data types and enumerations

6.5.4.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

6.5.4.2 Simple data types

No particular simple data types are defined for this interface, in addition to those defined in clause 4.4.

6.5.4.3 Enumeration: CrossingDirectionType

The enumeration CrossingDirectionType shall comply with the provisions defined in table 6.5.4.3-1.

Table 6.5.4.3-1: Enumeration CrossingDirectionType

Enumeration value	Description
UP	The threshold was crossed in upward direction.
DOWN	The threshold was crossed in downward direction.

7 VNF Fault Management interface

7.1 Description

This interface allows the NFVO to subscribe to notifications regarding VNF alarms provided by the VNFM, and API version information retrieval.

Virtualised resource alarms collected by the VNFM are filtered, correlated and modified by the VNFM and mapped to the corresponding VNF instance, resulting in alarms on that VNF instance which contain information on the VNFC(s) affected by the fault.

Reasons for creating alarms include the following:

- faults detected by the VNFM;
- faults generated by the VNFM due to changes in the state of virtualised resources used by the VNFs and their constituent VNFC instances managed by the VNFM, including changes in the state of the virtualised resources due to upcoming NFVI operation and maintenance; and
- faults generated by the VIM on virtualised resources used by the VNFs and their constituent VNFC instances managed by the VNFM.

NOTE: The present document specifies values of the attribute "faultType" and other attributes only for the case of faults generated by the VNFM because of changes in the state of virtualised due to upcoming NFVI operation and maintenance. The other cases listed above are not specified in the present document.

The operations provided through this interface are:

- Get Alarm List
- Acknowledge Alarm
- Subscribe
- Query Subscription Information
- Terminate Subscription
- Notify

7.1a API version

For the VNF fault management interface version as specified in the present document, the MAJOR version field shall be 1, the MINOR version field shall be 10 and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [8] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v1".

7.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [8]. The string "vnffm" shall be used to represent {apiName}. All resource URIs in the clauses below are defined relative to the above base URI.

Figure 7.2-1 shows the overall resource URI structure defined for the VNF fault management interface.

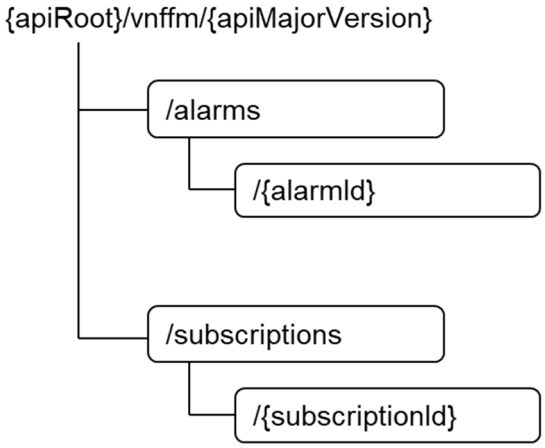

Figure 7.2-1: Resource URI structure of the VNF Fault Management interface

Table 7.2-1 lists the individual resources defined, and the applicable HTTP methods.

The VNFM shall support responding to requests for all HTTP methods on the resources in table 7.2-1 that are marked as "M" (mandatory) in the "Cat" column. The VNFM shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [8].

Table 7.2-1: Resources and methods overview of the VNF Fault Management interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
Alarms	/alarms	GET	M	Query alarms related to VNF instances
Individual alarm	/alarms/{alarmId}	GET	M	Read individual alarm
		PATCH	M	Acknowledge individual alarm
Subscriptions	/subscriptions	POST	M	Subscribe to VNF alarms
		GET	M	Query multiple subscriptions
Individual subscription	/subscriptions/{subscriptionId}	GET	M	Read an individual subscription
		DELETE	M	Terminate a subscription
Notification endpoint	(provided by API consumer)	POST	See note	Notify about VNF alarms.
		GET	See note	Test the notification endpoint.
NOTE: The VNFM shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the NFVO. If the NFVO supports invoking the POST method on the "Subscriptions" resource towards the VNFM, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.				

7.3 Sequence diagrams (informative)

7.3.1 Flow of the Get Alarm List operation

This clause describes a sequence flow for querying one or multiple alarms.

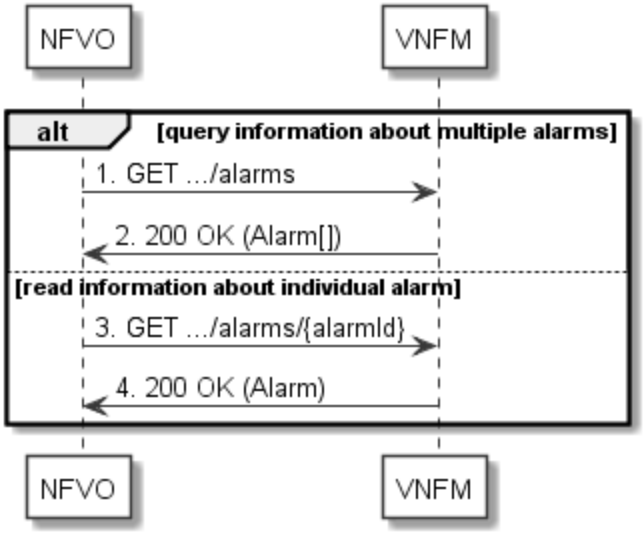

Figure 7.3.1-1: Flow of alarm query/read

Alarm query, as illustrated in figure 7.3.1-1, consists of the following steps:

1) If the NFVO intends to query all alarms, it sends a GET request to the "Alarms " resource.

2) The VNFM returns a "200 OK" response to the NFVO and includes zero or more data structures of type "Alarm" in the payload body.

3) If the NFVO intends to read a particular alarm, it sends a GET request to the "Individual alarm" resource, addressed by the appropriate alarm identifier in its resource URI.

4) The VNFM returns a "200 OK" response to the NFVO and includes a data structure of type "Alarm" in the payload body.

Error handling: In case of failure, appropriate error information is provided in the response.

7.3.2 Flow of acknowledging alarm

This clause describes the procedure to acknowledge an individual alarm.

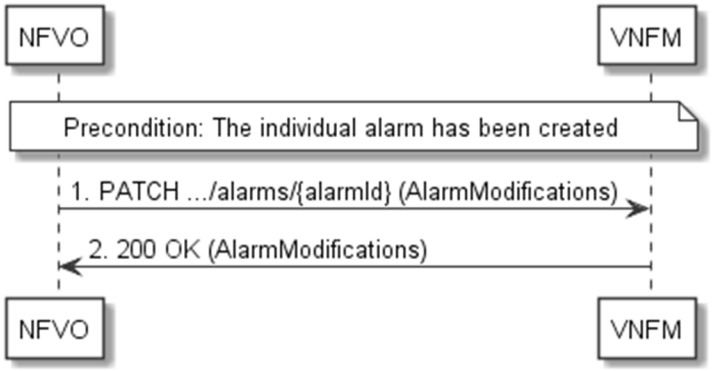

Figure 7.3.2-1: Flow of acknowledging alarm

Precondition: The resource representing the individual alarm has been created.

Acknowledge alarm, as illustrated in figure 7.3.2-1, consists of the following steps:

1) The NFVO sends a PATCH request to the individual alarm.

2) The VNFM returns a "200 OK" response to the NFVO and includes a data structure of type "AlarmModifications" in the payload body.

Error handling: In case of failure, appropriate error information is provided in the response.

7.3.3 Flow of managing subscriptions

This clause describes the procedure for creating, reading and terminating subscriptions to notifications related to VNF fault management.

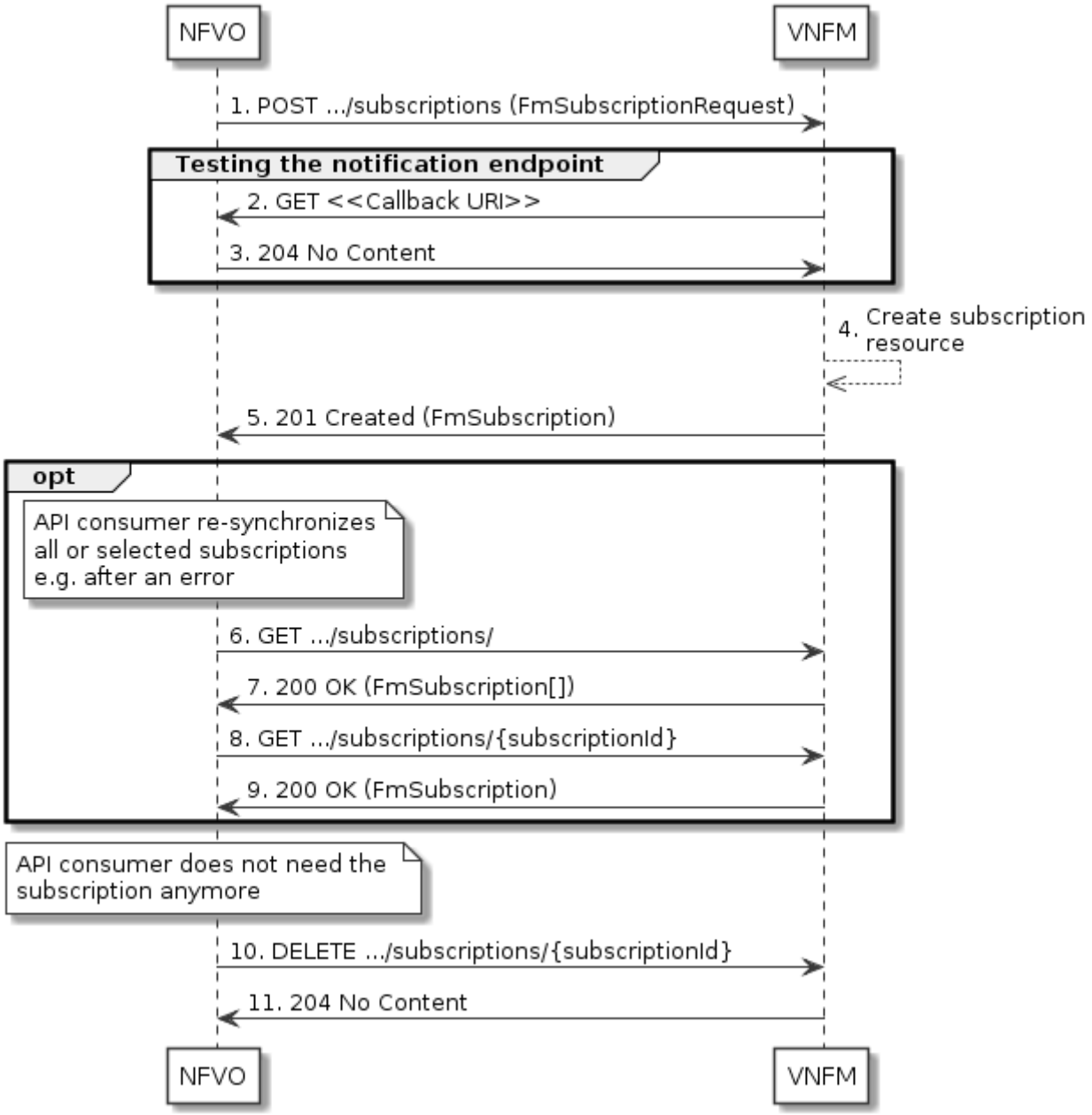

Figure 7.3.3-1: Flow of managing subscriptions

The procedure consists of the following steps as illustrated in figure 7.3.3-1:

1) The NFVO sends a POST request to the "Subscriptions" resource including in the payload body a data structure of type "FmSubscriptionRequest". This data structure contains filtering criteria and a callback URI to which the VNFM will subsequently send notifications about events that match the filter.

2) To test the notification endpoint that has been registered by the NFVO as part of the subscription, the VNFM sends a GET request to the notification endpoint URI.

3) The NFVO returns a "204 No Content" response to indicate success.

4) The VNFM creates a new subscription for notifications related to VNF fault management, and a resource that represents this subscription.

5) The VNFM returns a "201 Created" response containing a data structure of type "FmSubscription," representing the "Individual subscription" resource just created by the VNFM and provides the URI of the newly-created resource in the "Location" HTTP header.

6) If desired, e.g. to recover from an error situation, the NFVO can query information about its subscriptions by sending a GET request to the "Subscriptions" resource.

7) In that case, the VNFM returns a "200 OK" response that contains the list of representations of all existing subscriptions that were created by the NFVO.

8) If desired, e.g. to recover from an error situation, the NFVO can read information about a particular subscription by sending a GET request to the resource representing that individual subscription.

9) In that case, the VNFM returns a "200 OK" response that contains a representation of that individual subscription.

10) When the NFVO does not need the subscription anymore, it terminates the subscription by sending a DELETE request to the resource that represents the individual subscription.

11) The VNFM acknowledges the successful termination of the subscription by returning a "204 No Content" response.

Error handling: The VNFM rejects a subscription if the subscription information is not valid: endpoint cannot be reached, subscription information is malformed, etc.

7.3.4 Flow of sending notifications

This clause describes the procedure for sending notifications related to VNF fault management.

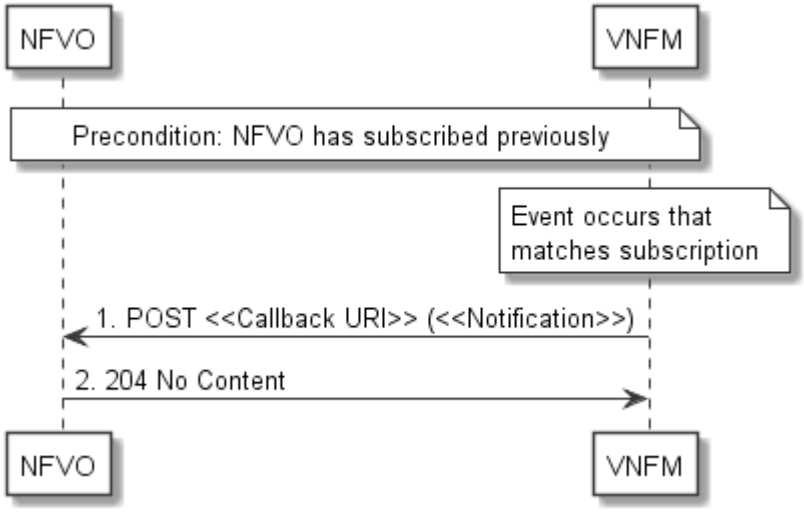

Figure 7.3.4-1: Flow of sending notifications

Precondition: The NFVO has subscribed previously for notifications related to VNF fault management.

The procedure consists of the following steps as illustrated in figure 7.3.4-1:

1) If an event occurs that matches the filtering criteria which are part of the subscription, the VNFM generates a notification that includes information about the event and sends it in the body of a POST request to the URI which the NFVO has registered as part of the subscription request. The variable <<Notification>> in the flow is a placeholder for the different types of notifications that can be sent by this API (see clauses 7.5.2.5, 7.5.2.6 and 7.5.2.7).

2) The NFVO acknowledges the successful delivery of the notification by returning a "204 No Content" response.

Error handling: If the VNFM does not receive the "204 No Content" response from the NFVO, it can retry sending the notification.

7.4 Resources

7.4.1 Introduction

This clause defines all the resources and methods provided by the VNF fault management interface.

7.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [8] are part of the VNF fault management interface.

7.4.2 Resource: Alarms

7.4.2.1 Description

This resource represents a list of alarms related to VNF instances.

7.4.2.2 Resource definition

The resource URI is:

{apiRoot}/vnffm/{apiMajorVersion}/alarms

This resource shall support the resource URI variables defined in table 7.4.2.2-1.

Table 7.4.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 7.1a.

7.4.2.3 Resource methods

7.4.2.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

7.4.2.3.2 GET

The API consumer can use this method to retrieve information about the alarm list.

This method shall follow the provisions specified in tables 7.4.2.3.2-1 and 7.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.2.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [8]. The VNFM shall support receiving this parameter as part of the URI query string. The NFVO may supply this parameter. The following attribute names shall be supported by the VNFM in the attribute-based filtering expression: id, managedObjectId, rootCauseFaultyResource/faultyResourceType, eventType, perceivedSeverity, probableCause.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource.

NOTE: There are no attribute selectors defined for this resource as the Alarm attributes with cardinality 0..1 or 0..N are not structurally complex in nature.

Table 7.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	Alarm	0..N	200 OK	Shall be returned when information about zero or more alarms has been queried successfully. The response body shall contain in an array the representations of zero or more alarms as defined in clause 7.5.2.4. If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [8]. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

7.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

7.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

7.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

7.4.3 Resource: Individual alarm

7.4.3.1 Description

This resource represents an individual alarm.

7.4.3.2 Resource definition

The resource URI is:

{apiRoot}/vnffm/{apiMajorVersion}/alarms/{alarmId}

This resource shall support the resource URI variables defined in table 7.4.3.2-1.

Table 7.4.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 7.1a.
alarmId	Identifier of the alarm. See note.
NOTE: This identifier can be retrieved from the "id" attribute of the "alarm" attribute in the AlarmNotification or AlarmClearedNotification. It can also be retrieved from the "id" attribute of the applicable array element in the payload body of the response to a GET request to the "Alarms" resource.	

7.4.3.3 Resource methods

7.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

7.4.3.3.2 GET

The API consumer can use this method to read an individual alarm.

This method shall follow the provisions specified in tables 7.4.3.3.2-1 and 7.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.3.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	Alarm	1	200	Shall be returned when information about an individual alarm has been read successfully. The response body shall contain a representation of the individual alarm.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

7.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

7.4.3.3.4 PATCH

This method modifies an "Individual alarm" resource.

This method shall follow the provisions specified in tables 7.4.3.3.4-1 and 7.4.3.3.4-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.3.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.3.3.4-2: Details of the PATCH request/response on this resource

	Data type	Cardinality	Description	
Request body	AlarmModifications	1	The parameter for the alarm modification, as defined in clause 7.5.2.8. The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [5].	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	AlarmModifications	1	200 OK	Shall be returned when the request has been accepted and completed. The response body shall contain attribute modifications for an "Individual alarm" resource (see clause 7.5.2.4).
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the "Individual alarm" resource. Typically, this is due to the fact that the alarm is already in the state that is requested to be set (such as trying to acknowledge an already-acknowledged alarm). The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	0..1	412 Precondition failed	Shall be returned upon the following error: A precondition given in an HTTP request header is not fulfilled. Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity. The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

7.4.3.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

7.4.4 Resource: Subscriptions

7.4.4.1 Description

This resource represents subscriptions. The API consumer can use this resource to subscribe to notifications related to VNF alarms and to query its subscriptions.

7.4.4.2 Resource definition

The resource URI is:

{apiRoot}/vnffm/{apiMajorVersion}/subscriptions

This resource shall support the resource URI variables defined in table 7.4.4.2-1.

Table 7.4.4.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 7.1a.

7.4.4.3 Resource methods

7.4.4.3.1 POST

The POST method creates a new subscription.

This method shall follow the provisions specified in tables 7.4.4.3.1-1 and 7.4.4.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual subscription" resource as defined in clause 7.4.5 shall have been created. This method shall not trigger any notification.

Creation of two "Individual subscription" resources with the same callback URI and the same filter can result in performance degradation and will provide duplicates of notifications to the NFVO, and might make sense only in very rare use cases. Consequently, the VNFM may either allow creating a new "Individual subscription" resource if another "Individual subscription" resource with the same filter and callback URI already exists (in which case it shall return the "201 Created" response code), or may decide to not create a duplicate "Individual subscription" resource (in which case it shall return a "303 See Other" response code referencing the existing "Individual subscription" resource with the same filter and callback URI).

Table 7.4.4.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Remarks
none supported		

Table 7.4.4.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	FmSubscriptionRequest	1		Details of the subscription to be created, as defined in clause 7.5.2.2.
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	FmSubscription	1	201 Created	Shall be returned when the subscription has been created successfully. The response body shall contain a representation of the created "Individual subscription" resource. The HTTP response shall include a "Location" HTTP header that points to the created "Individual subscription" resource.
	n/a		303 See Other	Shall be returned when a subscription with the same callback URI and the same filter already exists and the policy of the VNFM is to not create redundant subscriptions. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the existing "Individual subscription" resource. The response body shall be empty.
	ProblemDetails	1	422 Unprocessable Entity	Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this resource, the response code 422 shall also be returned if the VNFM has tested the Notification endpoint as described in clause 7.4.6.3.2 and the test has failed. In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

7.4.4.3.2 GET

The API consumer can use this method to retrieve the list of active subscriptions for VNF alarms subscribed by the API consumer. It can be used e.g. for resynchronization after error situations.

This method shall follow the provisions specified in tables 7.4.4.3.2-1 and 7.4.4.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.4.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Remarks
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [8]. The VNFM shall support receiving this parameter as part of the URI query string. The NFVO may supply this parameter. All attribute names that appear in the FmSubscription and in data types referenced from it shall be supported by the VNFM in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource.

Table 7.4.4.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	FmSubscription	0..N	200 OK	Shall be returned when the list of subscriptions has been queried successfully. The response body shall contain in an array the representations of all active subscriptions of the functional block that invokes the method, i.e. zero or more representations of FM subscriptions as defined in clause 7.5.2.3. If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [8]. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [8].

Request body	Data type	Cardinality	Description	
	n/a			
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

7.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

7.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

7.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

7.4.5 Resource: Individual subscription

7.4.5.1 Description

This resource represents an individual subscription for VNF alarms. The API consumer can use this resource to read and to terminate a subscription to notifications related to VNF fault management.

7.4.5.2 Resource definition

The resource URI is:

{apiRoot}/vnffm/{apiMajorVersion}/subscriptions/{subscriptionId}

This resource shall support the resource URI variables defined in table 7.4.5.2-1.

Table 7.4.5.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 7.1a.
subscriptionId	Identifier of this subscription. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual subscription" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

7.4.5.3 Resource methods

7.4.5.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

7.4.5.3.2 GET

The API consumer can use this method for reading an individual subscription for VNF alarms subscribed by the API consumer.

This method shall follow the provisions specified in tables 7.4.5.3.2-1 and 7.4.5.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.5.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.5.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	FmSubscription	1	200 OK	Shall be returned when information about an individual subscription has been read successfully. The response body shall contain a representation of the "Individual subscription" resource.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

7.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

7.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

7.4.5.3.5 DELETE

This method terminates an individual subscription.

This method shall follow the provisions specified in tables 7.4.5.3.5-1 and 7.4.5.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual subscription" resource shall not exist any longer. This means that no notifications for that subscription shall be sent to the formerly-subscribed API consumer.

NOTE: Due to race conditions, some notifications might still be received by the formerly-subscribed API consumer for a certain time period after the deletion.

Table 7.4.5.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.5.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned when the "Individual subscription" resource has been deleted successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

7.4.6 Resource: Notification endpoint

7.4.6.1 Description

This resource represents a notification endpoint for VNF alarms.

The API producer can use this resource to send notifications related to VNF alarms or about a rebuilt alarm list to a subscribed API consumer, which has provided the URI of this resource during the subscription process.

7.4.6.2 Resource definition

The resource URI is provided by the API consumer when creating the subscription.

This resource shall support the resource URI variables defined in table 7.4.6.2-1.

Table 7.4.6.2-1: Resource URI variables for this resource

Name	Definition
none supported	

7.4.6.3 Resource methods

7.4.6.3.1 POST

The POST method notifies a VNF alarm or that the alarm list has been rebuilt. The API consumer shall have previously created an "Individual subscription" resource with a matching filter.

This method shall follow the provisions specified in tables 7.4.6.3.1-1 and 7.4.6.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.6.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Each notification request body shall include exactly one of the alternatives defined in table 7.4.6.3.1-2.

Table 7.4.6.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Description	
Request body	AlarmNotification	1	Information of a VNF alarm	
	AlarmClearedNotification	1	Information of the clearance of a VNF alarm	
	AlarmListRebuiltNotification	1	Information that the alarm list has been rebuilt by the VNFM	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned when the notification has been delivered successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

7.4.6.3.2 GET

The GET method allows the API producer to test the notification endpoint that is provided by the API consumer, e.g. during subscription.

This method shall follow the provisions specified in tables 7.4.6.3.2-1 and 7.4.6.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.6.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.6.3.2-2: Details of the GET request/response on this resource

Request body	**Data type**	**Cardinality**	**Description**	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned to indicate that the notification endpoint has been tested successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

7.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

7.4.6.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

7.4.6.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

7.5 Data Model

7.5.1 Introduction

This clause defines the request and response data structures of the VNF fault management interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error and may choose to ignore them.

7.5.2 Resource and notification data types

7.5.2.1 Introduction

This clause defines the data structures to be used in the resource representations and notifications for the VNF fault management interface.

7.5.2.2 Type: FmSubscriptionRequest

This type represents a subscription request related to notifications about VNF faults. It shall comply with the provisions defined in table 7.5.2.2-1.

Table 7.5.2.2-1: Definition of the FmSubscriptionRequest data type

Attribute name	Data type	Cardinality	Description
filter	FmNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this subscription, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [8]. This attribute shall only be present if the subscriber requires authorization of notifications.

7.5.2.3 Type: FmSubscription

This type represents a subscription related to notifications about VNF faults. It shall comply with the provisions defined in table 7.5.2.3-1.

Table 7.5.2.3-1: Definition of the FmSubscription data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this "Individual subscription" resource.
filter	FmNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
_links	Structure (inlined)	1	Links for this resource.
>self	Link	1	URI of this resource.

7.5.2.4 Type: Alarm

The alarm data type encapsulates information about an alarm. It shall comply with the provisions defined in table 7.5.2.4-1.

Table 7.5.2.4-1: Definition of the Alarm data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this Alarm information element.
managedObjectId	Identifier	1	Identifier of the affected VNF instance.
rootCauseFaultyResource	FaultyResourceInfo	0..1	The virtualised resources that are causing the VNF fault. Shall be present if the alarm affects virtualised resources. See note 1.
alarmRaisedTime	DateTime	1	Time stamp indicating when the alarm is raised by the managed object.
alarmChangedTime	DateTime	0..1	Time stamp indicating when the alarm was last changed. It shall be present if the alarm has been updated.
alarmClearedTime	DateTime	0..1	Time stamp indicating when the alarm was cleared. It shall be present if the alarm has been cleared.
alarmAcknowledgedTime	DateTime	0..1	Time stamp indicating when the alarm was acknowledged. It shall be present if the alarm has been acknowledged.
ackState	Enum (inlined)	1	Acknowledgement state of the alarm. Permitted values: - UNACKNOWLEDGED. - ACKNOWLEDGED.
perceivedSeverity	PerceivedSeverityType	1	Perceived severity of the managed object failure.
eventTime	DateTime	1	Time stamp indicating when the fault was observed. See note 2.
eventType	EventType	1	Type of event.
faultType	String	0..1	Additional information to clarify the type of the fault. If the alarm is related to changes in the state of virtualised resources due to NFVI operation and maintenance, this attribute shall be set to "NFVI_OAM_VIRTUALISED_RESOURCE_STATE_CHANGE".
probableCause	String	1	Information about the probable cause of the fault. If the attribute "faultType" has the value "NFVI_OAM_VIRTUALISED_RESOURCE_STATE_CHANGE", the permitted values are: - "NFVI_COMPONENT_MAINTENANCE": Maintenance of NFVI components, e.g. physical maintenance/repair, hypervisor software updates, etc. - "NFVI_COMPONENT_EVACUATION": Evacuation of physical hosts. - "NFVI_COMPONENT_OPTIMIZATION": Operation and management of NFVI resources, e.g. to support energy efficiency or resource usage.
isRootCause	Boolean	1	Attribute indicating if this fault is the root for other correlated alarms. If true, then the alarms listed in the attribute "correlatedAlarmIds" are caused by this fault.
correlatedAlarmIds	Identifier	0..N	List of identifiers of other alarms correlated to this fault.
faultDetails	String	0..N	Provides additional information about the fault. See notes 1 and 2.
_links	Structure (inlined)	1	Links for this resource.
>self	Link	1	URI of this resource.
>objectInstance	Link	0..1	Link to the resource representing the VNF instance to which the notified alarm is correlated. Shall be present if the VNF instance information is accessible as a resource.

Attribute name	Data type	Cardinality	Description
NOTE 1: For an alarm about upcoming impact due to NFVI operation and maintenance (i.e. the attribute "faultType" has the value "NFVI_OAM_VIRTUALISED_RESOURCE_STATE_CHANGE"), the attribute "rootCauseFaultyResource" indicates a resource to be impacted. Further information on the upcoming impact (e.g. group of impacted resources, time of impact) is provided in the attribute "faultDetails". NOTE 2: When alarms are due to upcoming NFVI operation and maintenance (i.e. the attribute "faultType" has the value "NFVI_OAM_VIRTUALISED_RESOURCE_STATE_CHANGE"), the attribute "faultDetails" shall include information about the anticipated time of the maintenance. See provisions under the present table.			

If the attribute "faultType" has the value "NFVI_OAM_VIRTUALISED_RESOURCE_STATE_CHANGE", the following provisions apply for the values of the attribute "faultDetails" related to changes in the state of virtualised resources:

- One of the entries in the array shall provide information about the anticipated time of maintenance in the following format: "anticipatedTime=$time", wherein "$time" shall be formatted as a "DateTime", as specified in ETSI GS NFV-SOL 013 [8].
- One of the entries in the array shall provide identification information about the affinity/anti-affinity group defined in the VNFD that is associated to the affected virtualised resource indicated by "rootCauseFaultyResource" in the following format: "affinityOrAntiAffinityGroupId=$group", wherein "$group" shall be equal to the "affinityOrAntiAffinityGroupId" value in the corresponding "VduProfile" (for a VNFC/COMPUTE affected resource) or "VirtualLinkProfile" for a VL/NETWORK affected resource) in the VNFD, which is mapped by the VNFM to the virtualised resource group identifier in the virtualised resource change notification received by the VNFM from the VIM.

7.5.2.5 Type: AlarmNotification

This type represents an alarm notification about VNF faults. It shall comply with the provisions defined in table 7.5.2.5-1.

This notification shall be triggered by the VNFM when:

- An alarm has been created.
- An alarm has been updated, e.g. the severity of the alarm has changed.

Table 7.5.2.5-1: Definition of the AlarmNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "AlarmNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
alarm	Alarm	1	Information about an alarm including AlarmId, affected VNF identifier, and FaultDetails.
_links	Structure (inlined)	1	Links to resources related to this notification.
>subscription	NotificationLink	1	Link to the related subscription.

7.5.2.6 Type: AlarmClearedNotification

This type represents an alarm cleared notification about VNF faults. It shall comply with the provisions defined in table 7.5.2.6-1.

The notification shall be triggered by the VNFM when an alarm has been cleared.

Table 7.5.2.6-1: Definition of the AlarmClearedNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "AlarmClearedNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
alarmId	Identifier	1	Alarm identifier.
alarmClearedTime	DateTime	1	The time stamp indicating when the alarm was cleared.
_links	Structure (inlined)	1	Links to resources related to this notification.
>subscription	NotificationLink	1	Link to the related subscription.
>alarm	NotificationLink	1	Link to the resource that represents the related alarm.

7.5.2.7 Type: AlarmListRebuiltNotification

This type represents a notification that the alarm list has been rebuilt, e.g. if the VNFM detects its storage holding the alarm list is corrupted. It shall comply with the provisions defined in table 7.5.2.7-1.

The notification shall be triggered by the VNFM when the alarm list has been rebuilt, e.g. because the VNFM has detected that its storage holding the alarm list was corrupted.

Table 7.5.2.7-1: Definition of the AlarmListRebuiltNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "AlarmListRebuiltNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
_links	Structure (inlined)	1	Links to resources related to this notification.
>subscription	NotificationLink	1	Link to the related subscription.
>alarms	NotificationLink	1	Link to the alarm list, i.e. the "Alarms" resource.

7.5.2.8 Type: AlarmModifications

This type represents attribute modifications for an "Individual alarm" resource, i.e. modifications to a resource representation based on the "Alarm" data type. The attributes of "Alarm" that can be modified according to the provisions in clause 7.5.2.4 are included in the "AlarmModifications" data type.

The "AlarmModifications" data type shall comply with the provisions defined in table 7.5.2.8-1.

Table 7.5.2.8-1: Definition of the AlarmModifications data type

Attribute name	Data type	Cardinality	Description
ackState	Enum (inlined)	1	New value of the "ackState" attribute in "Alarm". Permitted values: - ACKNOWLEDGED - UNACKNOWLEDGED

7.5.3 Referenced structured data types

7.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but can neither be resource representations nor bound to any subscribe/notify mechanism.

7.5.3.2 Type: FmNotificationsFilter

This type represents a subscription filter related to notifications about VNF faults. It shall comply with the provisions defined in table 7.5.3.2-1.

At a particular nesting level in the filter structure, the following applies: All attributes shall match in order for the filter to match (logical "and" between different filter attributes). If an attribute is an array, the attribute shall match if at least one of the values in the array matches (logical "or" between the values of one filter attribute).

Table 7.5.3.2-1: Definition of the FmNotificationsFilter data type

Attribute name	Data type	Cardinality	Description
vnfInstanceSubscriptionFilter	VnfInstanceSubscriptionFilter	0..1	Filter criteria to select VNF instances about which to notify.
notificationTypes	Enum (inlined)	0..N	Match particular notification types. Permitted values: - AlarmNotification - AlarmClearedNotification - AlarmListRebuiltNotification See note.
faultyResourceTypes	FaultyResourceType	0..N	Match VNF alarms with a faulty resource type listed in this attribute.
perceivedSeverities	PerceivedSeverityType	0..N	Match VNF alarms with a perceived severity listed in this attribute.
eventTypes	EventType	0..N	Match VNF alarms with an event type listed in this attribute.
probableCauses	String	0..N	Match VNF alarms with a probable cause listed in this attribute.
NOTE: The permitted values of the "notificationTypes" attribute are spelled exactly as the names of the notification types to facilitate automated code generation systems.			

7.5.3.3 Type: FaultyResourceInfo

This type represents the faulty virtual resources that have a negative impact on a VNF. It shall comply with the provisions defined in table 7.5.3.3-1.

Table 7.5.3.3-1: Definition of the FaultyResourceInfo data type

Attribute name	Data type	Cardinality	Description
faultyResource	ResourceHandle	1	Information that identifies the faulty resource instance and its managing entity.
faultyResourceType	FaultyResourceType	1	Type of the faulty resource.

7.5.4 Referenced simple data types and enumerations

7.5.4.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

7.5.4.2 Simple data types

No particular simple data types are defined for this interface, in addition to those defined in clause 4.4.

7.5.4.3 Enumeration: PerceivedSeverityType

The enumeration PerceivedSeverityType shall comply with the provisions defined in table 7.5.4.3-1. It indicates the relative level of urgency for operator attention.

Table 7.5.4.3-1: Enumeration PerceivedSeverityType

Enumeration value	Description
CRITICAL	The Critical severity level indicates that a service affecting condition has occurred and an immediate corrective action is required. Such a severity can be reported, for example, when a managed object becomes totally out of service and its capability needs to be restored (Recommendation ITU-T X.733 [7]).
MAJOR	The Major severity level indicates that a service affecting condition has developed and an urgent corrective action is required. Such a severity can be reported, for example, when there is a severe degradation in the capability of the managed object and its full capability needs to be restored (Recommendation ITU-T X.733 [7]).
MINOR	The Minor severity level indicates the existence of a non-service affecting fault condition and that corrective action should be taken in order to prevent a more serious (for example, service affecting) fault. Such a severity can be reported, for example, when the detected alarm condition is not currently degrading the capacity of the managed object (Recommendation ITU-T X.733 [7]).
WARNING	The Warning severity level indicates the detection of a potential or impending service affecting fault, before any significant effects have been felt. Action should be taken to further diagnose (if necessary) and correct the problem in order to prevent it from becoming a more serious service affecting fault (Recommendation ITU-T X.733 [7]).
INDETERMINATE	The Indeterminate severity level indicates that the severity level cannot be determined (Recommendation ITU-T X.733 [7]).
CLEARED	The Cleared severity level indicates the clearing of one or more previously reported alarms. This alarm clears all alarms for this managed object that have the same Alarm type, Probable cause and Specific problems (if given) (Recommendation ITU-T X.733 [7]).

7.5.4.4 Enumeration: EventType

The enumeration EventType represents those types of events that trigger an alarm. It shall comply with the provisions defined in table 7.5.4.4-1.

Table 7.5.4.4-1: Enumeration EventType

Enumeration value	Description
COMMUNICATIONS_ALARM	An alarm of this type is associated with the procedure and/or process required conveying information from one point to another (Recommendation ITU-T X.733 [7]).
PROCESSING_ERROR_ALARM	An alarm of this type is associated with a software or processing fault (Recommendation ITU-T X.733 [7]).
ENVIRONMENTAL_ALARM	An alarm of this type is associated with a condition related to an enclosure in which the equipment resides (Recommendation ITU-T X.733 [7]).
QOS_ALARM	An alarm of this type is associated with degradation in the quality of a service (Recommendation ITU-T X.733 [7]).
EQUIPMENT_ALARM	An alarm of this type is associated with an equipment fault (Recommendation ITU-T X.733 [7]).

7.5.4.5 Enumeration: FaultyResourceType

The enumeration FaultyResourceType represents those types of faulty resource. It shall comply with the provisions defined in table 7.5.4.5-1.

Table 7.5.4.5-1: Enumeration FaultyResourceType

Enumeration value	Description
COMPUTE	Virtual compute resource
STORAGE	Virtual storage resource
NETWORK	Virtual network resource

8 VNF Indicator interface

8.1 Description

This interface allows the VNFM to provide information on value changes of VNF related indicators, and API version information retrieval.

VNF related indicators are declared in the VNFD. This interface is originally produced by the EM and/or VNF on the Ve-Vnfm-em and/or Ve-Vnfm-vnf reference point respectively (see ETSI GS NFV-SOL 002 [i.2]) and is re-exposed by the VNFM towards the NFVO.

The operations provided through this interface are:

- Get Indicator Value
- Subscribe
- Query Subscription Information
- Terminate Subscription
- Notify

8.1a API version

For the VNF indicator interface version as specified in the present document, the MAJOR version field shall be 1, the MINOR version field shall be 10 and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [8] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v1".

8.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [8]. The string "vnfind" shall be used to represent {apiName}. All resource URIs in the clauses below are defined relative to the above base URI.

Figure 8.2-1 shows the overall resource URI structure defined for the VNF Indicator interface.

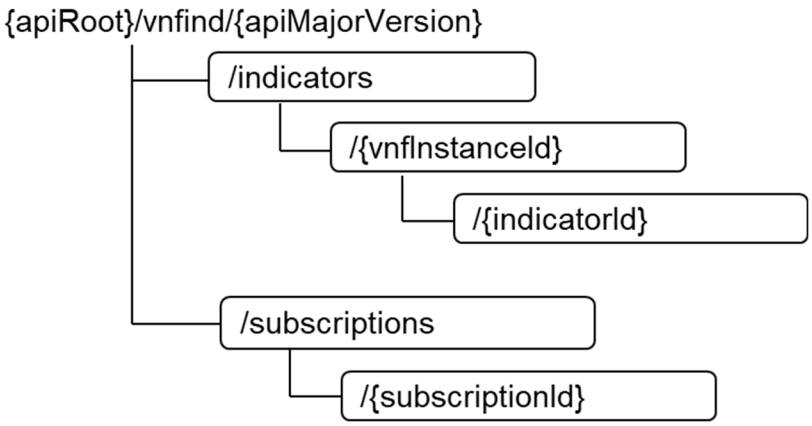

Figure 8.2-1: Resource URI structure of the VNF Indicator Interface

Table 8.2-1 lists the individual resources defined, and the applicable HTTP methods.

The VNFM shall support responding to requests for all HTTP methods on the resources in table 8.2-1 that are marked as "M" (mandatory) in the "Cat" column. The VNFM shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [8].

Table 8.2-1: Resources and methods overview of the VNF Indicator interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
VNF indicators	/indicators	GET	M	Query multiple VNF indicators. See note 1.
VNF indicators related to a VNF instance	/indicators/{vnfInstanceId}	GET	M	Query multiple VNF indicators related to one VNF instance. See note 1.
Individual VNF indicator	/indicators/{vnfInstanceId}/{indicatorId}	GET	M	Read an individual VNF indicator.
Subscriptions	/subscriptions	POST	M	Subscribe to VNF indicator change notifications.
		GET	M	Query multiple subscriptions.
Individual subscription	/subscriptions/{subscriptionId}	GET	M	Read an individual subscription.
		DELETE	M	Terminate a subscription.
Notification endpoint	(provided by API consumer)	POST	See note 2	Notify about VNF indicator change.
		GET	See note 2	Test the notification endpoint.
NOTE 1: This resource allows to query all VNF indicators that are known to the VNFM. NOTE 2: The VNFM shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the NFVO. If the NFVO supports invoking the POST method on the "Subscriptions" resource towards the VNFM, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.				

8.3 Sequence diagrams (informative)

8.3.1 Flow of querying VNF indicators

This clause describes a sequence for querying VNF indicators.

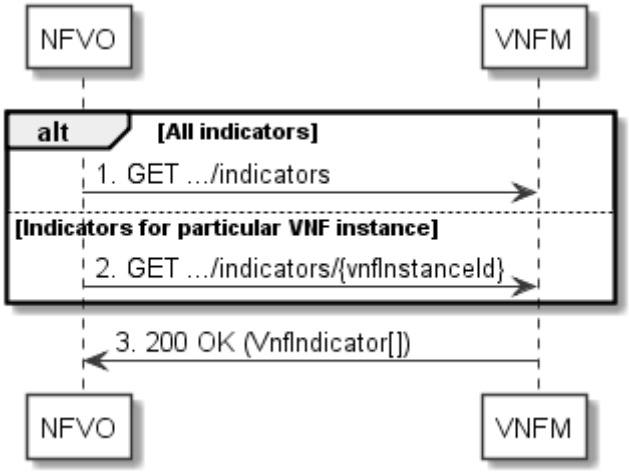

Figure 8.3.1-1: Flow of querying VNF indicators

VNF indicator query, as illustrated in figure 8.3.1-1, consists of the following steps:

1) If the NFVO intends to query all VNF indicators, it sends a GET request to the "VNF indicators" resource.

2) If the NFVO intends to query the VNF indicators of a particular VNF instance, it sends a GET request to the "VNF indicators related to a VNF instance" resource.

3) The VNFM returns a "200 OK" response to the NFVO and includes zero or more data structures of type "VnfIndicator" in the payload body.

Error handling: In case of failure, appropriate error information is provided in the response.

8.3.2 Flow of reading a VNF indicator

This clause describes a sequence for reading a VNF indicator, i.e. for getting the indicator value.

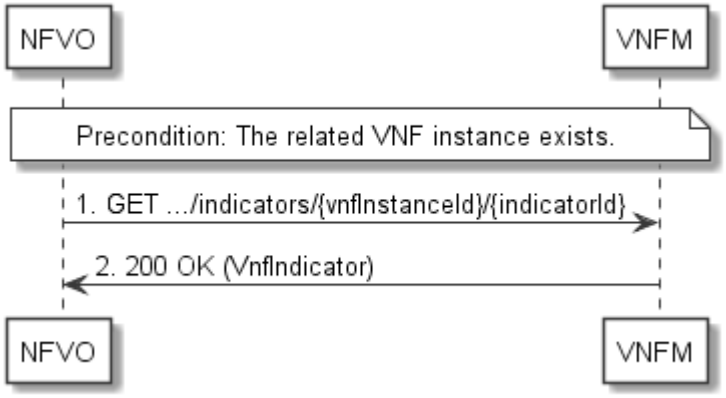

Figure 8.3.2-1: Flow of reading a VNF indicator

Precondition: The related VNF instance exists.

Reading a VNF indicator, as illustrated in figure 8.3.2-1, consists of the following steps:

1) The NFVO sends a GET request to the "Individual VNF indicator" resource that is to be read.

2) The VNFM returns a "200 OK" response to the NFVO and includes a data structure of type "VnfIndicator" in the payload body.

Error handling: In case of failure, appropriate error information is provided in the response.

8.3.3 Flow of managing subscriptions

This clause describes the procedure for creating, querying/reading and terminating subscriptions to notifications related to VNF indicator value changes.

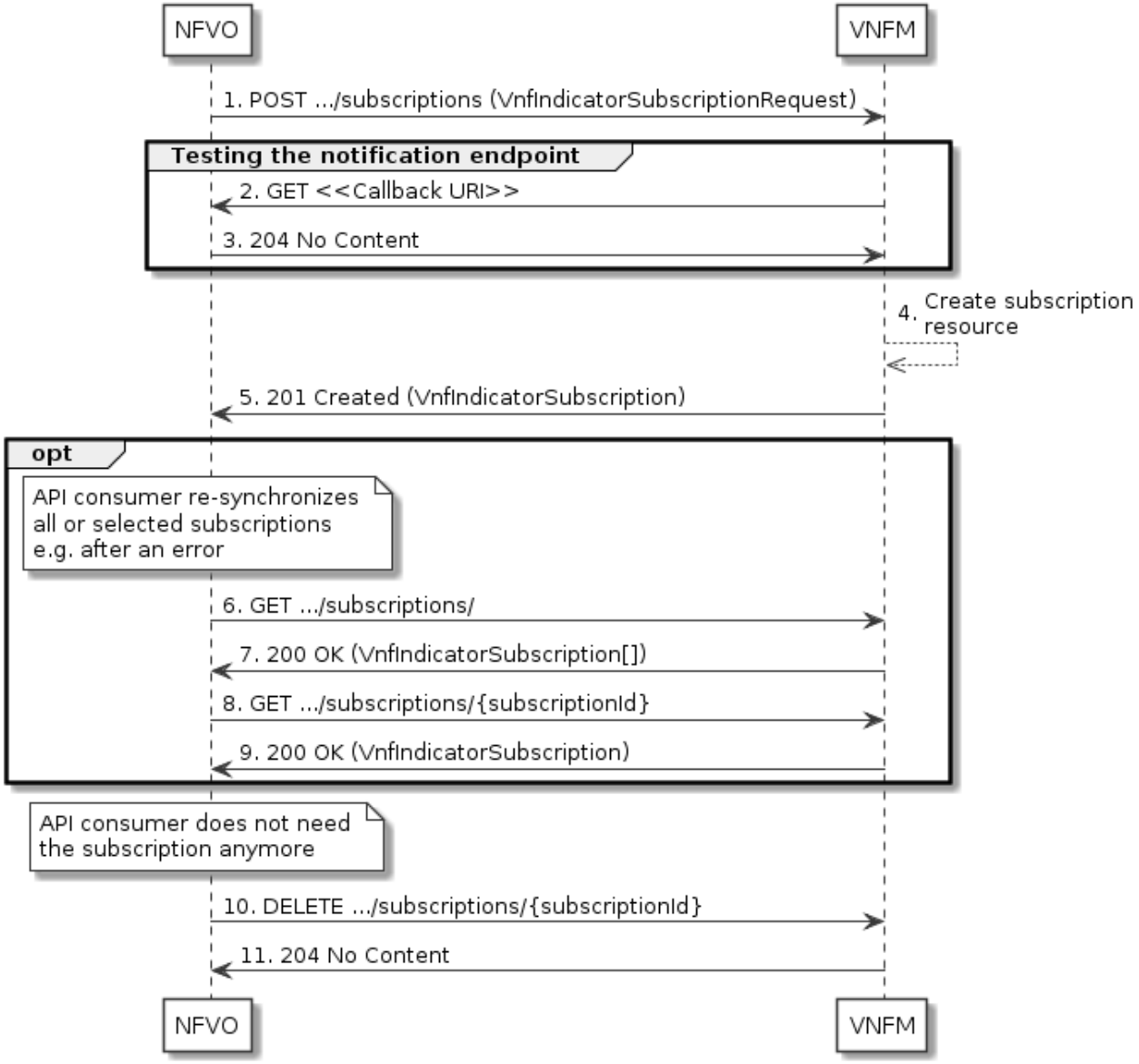

Figure 8.3.3-1: Flow of managing subscriptions

The procedure consists of the following steps as illustrated in figure 8.3.3-1:

1) The NFVO sends a POST request to the "Subscriptions" resource including in the payload body a data structure of type "VnfIndicatorSubscriptionRequest". That data structure contains filtering criteria and a callback URI to which the VNFM will subsequently send notifications about events that match the filter.

2) To test the notification endpoint that has been registered by the NFVO as part of the subscription, the VNFM sends a GET request to the notification endpoint URI.

3) The NFVO returns a "204 No Content" response to indicate success.

4) The VNFM creates a new subscription to notifications related to VNF indicator value changes, and a resource that represents this subscription.

5) The VNFM returns a 201 Created response containing a data structure of type "VnfIndicatorSubscription" representing the "Individual subscription" resource just created by the VNFM and provides the URI of the newly-created resource in the "Location" HTTP header.

6) If desired, e.g. to recover from an error situation, the NFVO can query information about its subscriptions by sending a GET request to the resource representing the subscriptions.

7) In that case, the VNFM returns a "200 OK" response that contains the list of representations of all existing subscriptions that were created by the NFVO.

8) If desired, e.g. to recover from an error situation, the NFVO can read information about a particular subscription by sending a GET request to the resource representing that individual subscription.

9) In that case, the VNFM returns a "200 OK" response that contains a representation of that individual subscription.

10) If the NFVO does not need the subscription anymore, it terminates the subscription by sending a DELETE request to the resource that represents the individual subscription to remove.

11) The NFVO acknowledges the successful termination of the subscription by returning a "204 No Content" response.

Error handling: The VNFM rejects a subscription if the subscription information is not valid: endpoint cannot be reached, subscription information is malformed, etc.

8.3.4 Flow of sending notifications

This clause describes the procedure for sending notifications.

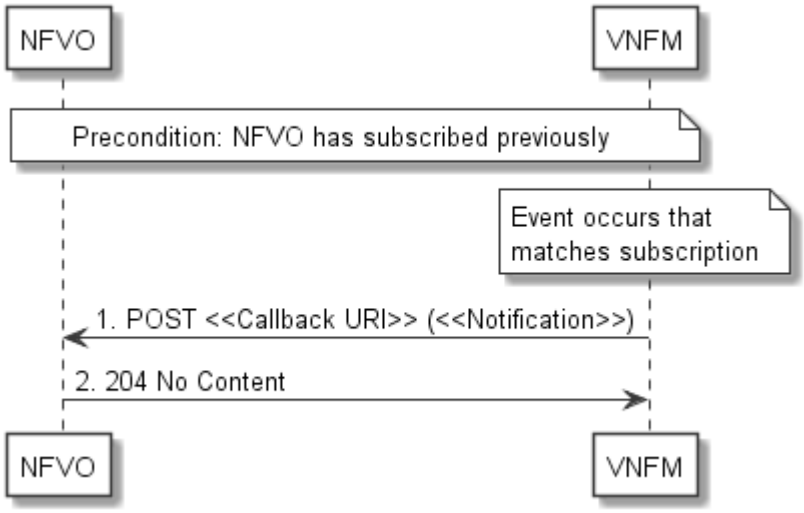

Figure 8.3.4-1: Flow of sending notifications

The procedure consists of the following steps as illustrated in figure 8.3.4-1.

Precondition: The NFVO has subscribed previously to notifications related to VNF indicator value changes.

1) If an event occurs that matches the filtering criteria which are part of the subscription, the VNFM generates a notification that includes information about the event and sends it in the body of a POST request to the callback URI which the NFVO has registered as part of the subscription request. The variable <<Notification>> in the flow is a placeholder for the different types of notifications that can be sent by this API (see clauses 8.5.2.5 and 8.5.2.6).

2) The NFVO acknowledges the successful delivery of the notification by returning a "204 No Content" response.

Error handling: If the VNFM does not receive the "204 No Content" response from the NFVO, it can retry sending the notification.

8.4 Resources

8.4.1 Introduction

This clause defines all the resources and methods provided by the VNF indicator interface.

8.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [8] are part of the VNF indicator interface.

8.4.2 Resource: VNF indicators

8.4.2.1 Description

This resource represents VNF indicators. The API consumer can use this resource to query multiple VNF indicators.

8.4.2.2 Resource definition

The resource URI is:

{apiRoot}/vnfind/{apiMajorVersion}/indicators

This resource shall support the resource URI variables defined in table 8.4.2.2-1.

Table 8.4.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 8.1a.

8.4.2.3 Resource methods

8.4.2.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.2.3.2 GET

The GET method queries multiple VNF indicators.

This method shall follow the provisions specified in tables 8.4.2.3.2-1 and 8.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.2.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [8]. The VNFM shall support receiving this parameter as part of the URI query string. The NFVO may supply this parameter. All attribute names that appear in the VnfIndicator data type and in data types referenced from it shall be supported by the VNFM in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource.

Table 8.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfIndicator	0..N	200 OK	Shall be returned when information about zero or more VNF indicators has been queried successfully. The response body shall contain in an array the representations of all VNF indicators that match the attribute filter, i.e. zero or more representations of VNF indicators as defined in clause 8.5.2.2. If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [8]. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

8.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.3 Resource: VNF indicators related to a VNF instance

8.4.3.1 Description

This resource represents VNF indicators related to a VNF instance. The API consumer can use this resource to query multiple VNF indicators that are related to a particular VNF instance.

8.4.3.2 Resource definition

The resource URI is:

{apiRoot}/vnfind/{apiMajorVersion}/indicators/{vnfInstanceId}

This resource shall support the resource URI variables defined in table 8.4.3.2-1.

Table 8.4.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 8.1a.
vnfInstanceId	Identifier of the VNF instance to which the VNF indicator applies. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual VNF instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

8.4.3.3 Resource methods

8.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.3.3.2 GET

The GET method queries multiple VNF indicators related to a VNF instance.

This method shall follow the provisions specified in tables 8.4.3.3.2-1 and 8.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [8]. The VNFM shall support receiving this parameter as part of the URI query string. The NFVO may supply this parameter. All attribute names that appear in the VnfIndicator data type and in data types referenced from it shall be supported by the VNFM in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource.

Table 8.4.3.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfIndicator	0..N	200 OK	Shall be returned when information about zero or more VNF indicators has been queried successfully. The response body shall contain in an array the representations of all VNF indicators that are related to the particular VNF instance and that match the attribute filter, i.e. zero or more representations of VNF indicators as defined in clause 8.5.2.2. If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [8]. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

8.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.3.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.3.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.4 Resource: Individual VNF indicator

8.4.4.1 Description

This resource represents an individual VNF indicator. The API consumer can use this resource to read an individual VNF indicator.

8.4.4.2 Resource definition

The resource URI is:

{apiRoot}/vnfind/{apiMajorVersion}/indicators/{vnfInstanceId}/{indicatorId}

This resource shall support the resource URI variables defined in table 8.4.4.2-1.

Table 8.4.4.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 8.1a.
vnfInstanceId	Identifier of the VNF instance to which the VNF indicator applies. See note 1.
indicatorId	Identifier of the VNF indicator. See note 2.
NOTE 1: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual VNF instance" resource. It can also be retrieved from the "id" attribute in the payload body of that response. NOTE 2: This identifier can be retrieved from the resource referenced by the payload body in the response to a POST request creating a new "Individual VNF instance" resource.	

8.4.4.3 Resource methods

8.4.4.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.4.3.2 GET

The GET method reads a VNF indicator.

This method shall follow the provisions specified in tables 8.4.4.3.2-1 and 8.4.4.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.4.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 8.4.4.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfIndicator	1	200 OK	Shall be returned when the VNF indicator has been read successfully. The response body shall contain the representation of the VNF indicator.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

8.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.5 Resource: Subscriptions

8.4.5.1 Description

This resource represents subscriptions. The API consumer can use this resource to subscribe to notifications related to VNF indicator value changes, and to query its subscriptions.

8.4.5.2 Resource definition

The resource URI is:

{apiRoot}/vnfind/{apiMajorVersion}/subscriptions

This resource shall support the resource URI variables defined in table 8.4.5.2-1.

Table 8.4.5.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 8.1a.

8.4.5.3 Resource methods

8.4.5.3.1 POST

The POST method creates a new subscription.

As the result of successfully executing this method, a new "Individual subscription" resource as defined in clause 8.4.6 shall have been created. This method shall not trigger any notification.

Creation of two "Individual subscription" resources with the same callback URI and the same filter can result in performance degradation and will provide duplicates of notifications to the NFVO, and might make sense only in very rare use cases. Consequently, the VNFM may either allow creating a new "Individual subscription" resource if another "Individual subscription" resource with the same filter and callback URI already exists (in which case it shall return the "201 Created" response code), or may decide to not create a duplicate "Individual subscription" resource (in which case it shall return a "303 See Other" response code referencing the existing "Individual subscription" resource with the same filter and callback URI).

This method shall follow the provisions specified in tables 8.4.5.3.1-1 and 8.4.5.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.5.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 8.4.5.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	VnfIndicatorSubscriptionRequest	1	Details of the subscription to be created.	
Response body	VnfIndicatorSubscription	1	201 Created	Shall be returned when the subscription has been created successfully. The response body shall contain a representation of the created "Individual subscription" resource. The HTTP response shall include a "Location" HTTP header that points to the created resource.
	n/a		303 See Other	Shall be returned when a subscription with the same callback URI and the same filter already exists and the policy of the VNFM is to not create redundant subscriptions. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the existing "Individual subscription" resource. The response body shall be empty.
	ProblemDetails	1	422 Unprocessable Entity	Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this resource, the response code 422 shall also be returned if the VNFM has tested the Notification endpoint as described in clause 8.4.7.3.2 and the test has failed. In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

8.4.5.3.2 GET

The GET method queries the list of active subscriptions of the functional block that invokes the method. It can be used e.g. for resynchronization after error situations.

This method shall follow the provisions specified in tables 8.4.5.3.2-1 and 8.4.5.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.5.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [8]. The VNFM shall support receiving this parameter as part of the URI query string. The NFVO may supply this parameter. All attribute names that appear in the VnfIndicatorSubscription data type and in data types referenced from it shall be supported by the VNFM in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource.

Table 8.4.5.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfIndicatorSubscription	0..N	200 OK	Shall be returned when the list of subscriptions has been queried successfully. The response body shall contain in an array the representations of all active subscriptions of the functional block that invokes the method which match the attribute filter, i.e. zero or more representations of VNF indicator subscriptions as defined in clause 8.5.2.4. If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [8]. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

8.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.6 Resource: Individual subscription

8.4.6.1 Description

This resource represents an individual subscription. The API consumer can use this resource to read and to terminate a subscription to notifications related to VNF indicator value changes.

8.4.6.2 Resource definition

The resource URI is:

{apiRoot}/vnfind/{apiMajorVersion}/subscriptions/{subscriptionId}

This resource shall support the resource URI variables defined in table 8.4.6.2-1.

Table 8.4.6.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 8.1a.
subscriptionId	Identifier of this subscription. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual subscription" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

8.4.6.3 Resource methods

8.4.6.3.1 POST

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.6.3.2 GET

The GET method reads an individual subscription.

This method shall follow the provisions specified in tables 8.4.6.3.2-1 and 8.4.6.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.6.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 8.4.6.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfIndicatorSubscription	1	200 OK	Shall be returned when information about an individual subscription has been read successfully. The response body shall contain a representation of the "Individual subscription" resource.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

8.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.6.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.6.3.5 DELETE

The DELETE method terminates an individual subscription.

This method shall follow the provisions specified in tables 8.4.6.3.5-1 and 8.4.6.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual subscription" resource shall not exist any longer. This means that no notifications for that subscription shall be sent to the formerly-subscribed API consumer.

NOTE: Due to race conditions, some notifications might still be received by the formerly-subscribed API consumer for a certain time period after the deletion.

Table 8.4.6.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 8.4.6.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned when the "Individual subscription" resource has been deleted successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

8.4.7 Resource: Notification endpoint

8.4.7.1 Description

This resource represents a notification endpoint.

The API producer can use this resource to send notifications to a subscribed API consumer, which has provided the URI of this resource during the subscription process.

8.4.7.2 Resource definition

The resource URI is provided by the API consumer when creating the subscription.

This resource shall support the resource URI variables defined in table 8.4.7.2-1.

Table 8.4.7.2-1: Resource URI variables for this resource

Name	Definition
none supported	

8.4.7.3 Resource methods

8.4.7.3.1 POST

The POST method delivers a notification from the API producer to an API consumer. The API consumer shall have previously created an "Individual subscription" resource with a matching filter.

This method shall follow the provisions specified in tables 8.4.7.3.1-1 and 8.4.7.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.7.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Each notification request body shall include exactly one of the alternatives defined in table 8.4.7.3.1-2.

Table 8.4.7.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Description	
Request body	VnfIndicatorValueChangeNotification	1	A notification about VNF indicator value changes.	
	SupportedIndicatorsChangeNotification	1	A notification about changes of the set of supported indicators.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned when the notification has been delivered successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

8.4.7.3.2 GET

The GET method allows the API producer to test the notification endpoint that is provided by the API consumer, e.g. during subscription.

This method shall follow the provisions specified in tables 8.4.7.3.2-1 and 8.4.7.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.7.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 8.4.7.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned to indicate that the notification endpoint has been tested successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

8.4.7.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.7.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.4.7.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

8.5 Data model

8.5.1 Introduction

This clause defines the request and response data structures of the VNF Indicator interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error and may choose to ignore them.

8.5.2 Resource and notification data types

8.5.2.1 Introduction

This clause defines the data structures to be used in resource representations and notifications.

8.5.2.2 Type: VnfIndicator

This type represents a VNF indicator value. It shall comply with the provisions defined in table 8.5.2.2-1.

Table 8.5.2.2-1: Definition of the VnfIndicator data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnfd	1	Identifier of this VNF indicator.
name	String	0..1	Human readable name of the indicator. Shall be present if defined in the VNFD.
value	Object	1	Provides the value of the indicator. The value format is defined in the VNFD. See note.
vnfInstanceId	Identifier	1	Identifier of the VNF instance which provides the indicator value.
_links	Structure (inlined)	1	Links for this resource.
>self	Link	1	URI of this resource.
>vnfInstance	Link	1	Link to the related "Individual VNF instance" resource.
NOTE: ETSI GS NFV-SOL 001 [i.4] specifies the structure and format of the VNFD based on TOSCA specifications.			

8.5.2.3 Type: VnfIndicatorSubscriptionRequest

This type represents a subscription request related to VNF indicator value change notifications. It shall comply with the provisions defined in table 8.5.2.3-1.

Table 8.5.2.3-1: Definition of the VnfIndicatorSubscriptionRequest data type

Attribute name	Data type	Cardinality	Description
filter	VnfIndicatorNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this subscription, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [8]. This attribute shall only be present if the subscriber requires authorization of notifications.

8.5.2.4 Type: VnfIndicatorSubscription

This type represents a subscription related to notifications about VNF indicator value changes. It shall comply with the provisions defined in table 8.5.2.4-1.

Table 8.5.2.4-1: Definition of the VnfIndicatorSubscription data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this "Individual subscription" resource.
filter	VnfIndicatorNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
_links	Structure (inlined)	1	Links for this resource.
>self	Link	1	URI of this resource.

8.5.2.5 Type: VnfIndicatorValueChangeNotification

This type represents a VNF indicator value change notification. It shall comply with the provisions defined in table 8.5.2.5-1.

The notification shall be triggered by the VNFM when the value of an indicator has changed.

Table 8.5.2.5-1: Definition of the VnfIndicatorValueChangeNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "VnfIndicatorValueChangeNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
vnfIndicatorId	IdentifierInVnfd	1	Identifier of the VNF indicator whose value has changed.
name	String	0..1	Human readable name of the VNF indicator. Shall be present if defined in the VNFD.
value	Object	1	Provides the value of the VNF indicator. The value format is defined in the VNFD. See note.
vnfInstanceId	Identifier	1	Identifier of the VNF instance which provides the indicator value.
_links	Structure (inlined)	1	Links for this resource.
>vnfInstance	NotificationLink	0..1	Link to the related "Individual VNF instance" resource. Shall be present if the VNF instance information is accessible as a resource.
>subscription	NotificationLink	1	Link to the related subscription.
NOTE: ETSI GS NFV-SOL 001 [i.4] specifies the structure and format of the VNFD based on TOSCA specifications.			

8.5.2.6 Type: SupportedIndicatorsChangeNotification

This type represents a notification to inform the receiver that the set of indicators supported by a VNF instance has changed. It shall comply with the provisions defined in table 8.5.2.6-1.

The notification shall be triggered by the VNFM when the set of supported VNF indicators has changed as a side effect of the "Change current VNF package" operation. It may be triggered by the VNFM when a VNF has been instantiated.

Table 8.5.2.6-1: Definition of the SupportedIndicatorsChangeNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "SupportedIndicatorsChangeNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
vnfInstanceId	Identifier	1	Identifier of the VNF instance which provides the indicator value.
supportedIndicators	Structure (inlined)	0..N	Set of VNF indicators supported by the VNF instance.
>vnfIndicatorId	IdentifierInVnfd	1	Identifier of the VNF indicator whose value has changed.
>name	String	0..1	Human readable name of the VNF indicator. Shall be present if defined in the VNFD. See note.
_links	Structure (inlined)	1	Links for this resource.
>vnfInstance	NotificationLink	0..1	Link to the related "Individual VNF instance" resource. Shall be present if the VNF instance information is accessible as a resource.
>subscription	NotificationLink	1	Link to the related subscription.
NOTE: ETSI GS NFV-SOL 001 [i.4] specifies the structure and format of the VNFD based on TOSCA specifications.			

8.5.3 Referenced structured data types

8.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but can neither be resource representations nor bound to any subscribe/notify mechanism.

8.5.3.2 Type: VnfIndicatorNotificationsFilter

This type represents a subscription filter for notifications related to VNF indicators. It shall comply with the provisions defined in table 8.5.3.2-1.

At a particular nesting level in the filter structure, the following applies: All attributes shall match in order for the filter to match (logical "and" between different filter attributes). If an attribute is an array, the attribute shall match if at least one of the values in the array matches (logical "or" between the values of one filter attribute).

Table 8.5.3.2-1: Definition of the VnfIndicatorNotificationsFilter data type

Attribute name	Data type	Cardinality	Description
vnfInstanceSubscriptionFilter	VnfInstanceSubscriptionFilter	0..1	Filter criteria to select VNF instances about which to notify.
notificationTypes	Enum (inlined)	0..N	Match particular notification types. Permitted values: - VnfIndicatorValueChangeNotification - SupportedIndicatorsChangeNotification See note.
indicatorIds	IdentifierInVnfd	0..N	Match particular VNF indicator identifiers.
NOTE: The permitted values of the "notificationTypes" attribute are spelled exactly as the names of the notification types to facilitate automated code generation systems.			

8.5.4 Referenced simple data types and enumerations

No particular simple data types and enumerations are defined for this interface, in addition to those defined in clause 4.4.

9 VNF Lifecycle Operation Granting interface

9.1 Description

This interface allows the VNFM to obtain from the NFVO permission and configuration parameters for a VNF lifecycle operation. Further, this interface allows API version information retrieval.

The operation provided through this interface is:

- Grant Lifecycle Operation.

This operation allows the VNFM to request a grant for authorization of a VNF lifecycle operation. This interface supports multiple use cases, such as:

- The NFVO can approve or reject a request based on policies (e.g. dependencies between VNFs) and available capacity.
- When applicable, the NFVO can reserve resources based on the VNFM's virtualised resources request. Depending on operator policies the NFVO can decide on whether to reserve virtualised resources or physical compute hosts.
- The NFVO can provide to the VNFM information about the VIM where cloud resources are allocated. This can include additional information such as the resource zone.
- The NFVO can provide to the VNFM container namespaces in which the MCIOs of a VNF with containerized components shall be deployed.

When requesting resource creation or modification, the VNFM references the resource definitions that are available to the NFVO in the VNFD. When resources are to be released or modified, the VNFM provides references to the existing resources in the request.

Per each VNFM, one of the following operator policies can be selected as a configuration, by means outside the scope of the present document, to determine how the NFVO and the VNFM handle resource reservations in a grant request:

1) Policy GRANT_APPROVE: The NFVO approves the VIM resources to be allocated by the VNFM. In general, resource availability is not guaranteed. No explicit reservation identifier is returned to the VNFM. Optionally, to guarantee resource availability, the NFVO may do a reservation and use implicit reservation identification towards the VNFM, i.e. associate the reservation to the VIM access information.

2) Policy GRANT_RESERVE: The NFVO guarantees the availability of the VIM resources to be allocated. The NFVO provides to the VNFM reservation identifier(s). Each such identifier identifies the reservation which is applicable to the resource requirements and which the VNFM shall use in the subsequent resource management operation.

These policies are used to configure the behaviour of both the NFVO and the VNFM identically, also considering the resource reservation capabilities of the VIM.

In the case of VNF with containerized components, information exchanges associated to resources is performed at two different levels: OS container and MCIO. As specified in clause 5.2.1.2 of ETSI GS NFV-IFA 040 [i.15], the properties of Compute MCIO and Storage MCIO are specified in the VDU of the VNFD, and there is a 1:1 relationship between an instantiated VNFC of a VNF and the Compute MCIO. For storage, the resource requirements are specified as VirtualStorageDesc of the VNFD:

- With regards to the VNF lifecycle operation granting, the NFVO handles the granting of resource requests at MCIO level but with additional exchange of information related to OS containers. Even though there is exposure of information at OS container level, there is no management of OS containers performed neither by the VNFM nor the NFVO. The NFVO can collect detailed information about OS container resource requests to perform the granting of resources as part of the overall resource orchestration. The requests for resources to be allocated to an MCIO is derived from the OsContainerDesc resource templates referenced in the ResourceDefinition. The OsContainerDesc(s) are also referred per VDU in the ResourceDefinition (via the attribute "vduId"). This is determined based on the relationship between VDU and OsContainerDesc as defined in the VNFD (see clause 7.1.6.2 of ETSI GS NFV-IFA 011 [10]). Based on the number of occurrences of ResourceDefinitions, the NFVO can derive the number of VNFC instances affected (e.g. to be added, removed, updated) by the VNF LCM operation.

- With regards to the VNF lifecycle management, the NFVO can collect runtime information of the VNF instance at the MCIO level with the QueryVnf operation and VnfLcmOperationOccurrenceNotification specified in clauses 7.2.9 and 7.2.15 of the present document, respectively. The "VnfInfo" provides information about the mapping of the compute, storage and network resources with the MCIO whose declarative descriptor specify corresponding resource requests (refer to clauses 5.5.3.5, 5.5.3.6, 5.5.3.7 and 4.4.1.7 of the present document). For each VnfcResourceInfo, a mapping to the vduId is also provided, similarly as with the VM-based case for components of a VNF. The NFVO can correlate the number of occurrences of VnfcResourceInfo in the VnfInfo with the information derived from the granting exchanges.

The NFVO is expected to consider the container namespaces, including their resource quota, when evaluating granting requests for lifecycle operations on containerized workloads. Requests for resources to be allocated to MCIOs are derived from the OsContainerDesc resource templates referenced in the grant request. The containerized workloads based on a particular MCIOP are deployed within one container namespace. To determine the container namespace to be returned in the grant response, the NFVO shall resolve the association of the OsContainerDesc resource templates to the corresponding MCIOP. The VNFD contains profiles of all MCIOPs in the VNF deployment flavour information element. An MCIOP profile contains a list of associated VDUs which in turn reference the OsContainerDesc resource templates. By using these associations, the NFVO can return the namespace for the resource definitions related to the MCIOP.

In the GrantVnfLifecycleOperation response, the NFVO can return information that allows to distribute the resources of a VNF over multiple resource zones. This decision is guided by affinity/anti-affinity rules in the VNFD as well as by placement constraints passed in the GrantVnfLifecycleOperation request. The NFVO can also return information that allows to manage the resources of a VNF using multiple VIMs, guided by VIM selection constraints passed in the GrantVnfLifecycleOperation request.

In the present document, as part of the granting mechanism, attributes are defined for signalling the decision to use multiple VIMs per VNF. For the support of VNFs that include resources managed by multiple VIMs, the mechanisms to manage the VNF-internal Virtual Link (VL) requirements across multiple VIMs leverage the concept of "externally managed VL" and the support of "multi-site connectivity services". With these mechanisms, the NFVO can manage the connectivity for the VNF, not only within a given NFVI-PoP (or site), but also across different NFVI-PoPs.

When the VDUs of the VNF are realized by a set of OS containers, the NFVO determines whether the namespace quota associated with the VNF will be impacted by this LCM operation and needs to be changed. If the change is needed, then the NFVO further initiates a request of modification of the namespace quota associated with the VNF to the Container Infrastructure Service Management (CISM) function.

When the VDUs of the VNF are realized by a set of virtual machines, the following applies: Some LCM operations, including HealVnf and ChangeCurrentVnfPkg, might try different resource management operations for the same resource to achieve a certain goal. For instance, healing might first try to reboot a virtual compute resource. If this does not fix the problem, it might delete and re-create the virtual compute resource. Because there is a single granting exchange per LCM operation, this granting exchange needs to obtain permission up front, without knowing which of the operations will actually lead to success. Further, temporary resources may not be needed in all cases, depending on the state of the resources and the progress of the operation.

The following provisions therefore apply when the VDUs of the VNF are realized by a set of virtual machines:

1) When requesting a grant, the VNFM need not include information about planned resource management operations that do not change attributes of the VIM-level resource. For example, rebooting a virtual compute resource or rebuilding a virtual compute resource using the image, bootdata and flavour that were used when creating the resource will not lead to observable changes of the virtual compute resource.

2) Even though a grant for a resource management operation has been approved by the NFVO, it is not mandatory for the VNFM to execute the granted resource management operation as part of the LCM operation. The NFVO can learn which resources are affected by an LCM operation e.g. by listening to LcmOpOccNotifications or by reading the related LcmOpOcc object.

3) If, for a particular resource, there are resource modifications and resource deletion/re-creation as alternatives, the VNFM shall include in the grant request information about the more impactful operation i.e. deletion and re-creation. If the deletion/re-creation was granted, a modification of the affected resource may be executed by the VNFM instead.

9.1a API version

For the VNF lifecycle operation granting interface as specified in the present document, the MAJOR version field shall be 1, the MINOR version field shall be 10 and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [8] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v1".

9.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [8]. The string "grant" shall be used to represent {apiName}. All resource URIs in the clauses below are defined relative to the above base URI.

Figure 9.2-1 shows the overall resource URI structure defined for the VNF Lifecycle Operation Granting interface.

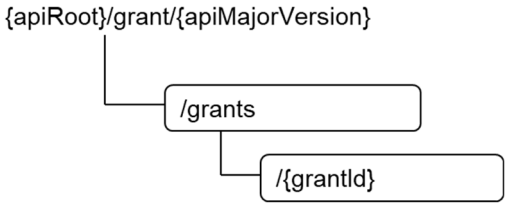

Figure 9.2-1: Resource URI structure of the VNF Lifecycle Operation Granting Interface

Table 9.2-1 lists the individual resources defined, and the applicable HTTP methods.

The NFVO shall support responding to requests for all HTTP methods on the resources in table 9.2-1 that are marked as "M" (mandatory) in the "Cat" column. The NFVO shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [8].

Table 9.2-1: Resources and methods overview of the VNF Lifecycle Operation Granting interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
Grants	/grants	POST	M	Request a grant
Individual grant	/grants/{grantId}	GET	M	Read a grant

9.3 Sequence diagrams (informative)

9.3.1 Flow of grant request with synchronous response

This clause describes a sequence for a grant request with synchronous (i.e. immediate) response. If the NFVO can decide immediately what to respond to a grant request, it returns the response immediately.

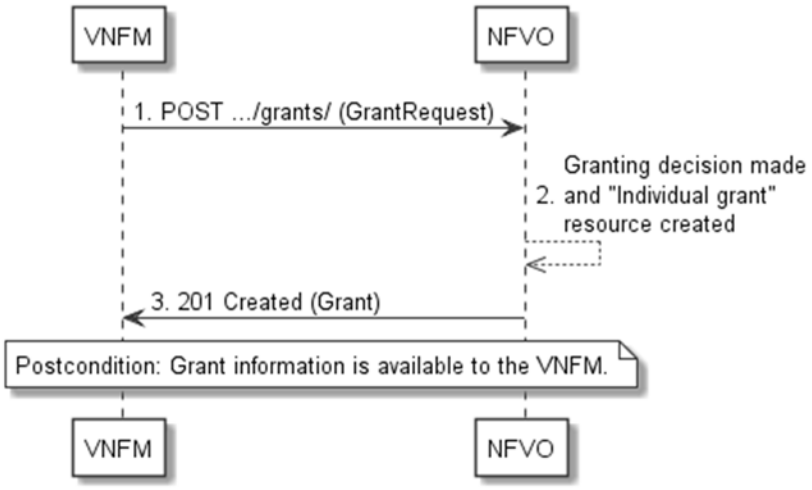

Figure 9.3.1-1: Flow of granting with synchronous response

Granting with synchronous response, as illustrated in figure 9.3.1-1, consists of the following steps:

1) The VNFM sends a POST request to the "Grants" resource with a "GrantRequest" data structure in the body.

2) The NFVO makes the granting decision and creates a new "Individual grant" resource.

3) The NFVO returns to the VNFM a "201 Created" response with a "Grant" data structure in the body and a "Location" HTTP header that points to the new "Individual grant" resource.

Postcondition: The grant information is available to the VNFM.

9.3.2 Flow of grant request with asynchronous response

This clause describes a sequence for a grant request with asynchronous (i.e. delayed) response. If the NFVO cannot decide immediately what to respond to a grant request, and therefore runs the risk of a timeout of the http connection while waiting for the completion of the decision, it returns the response in an asynchronous (delayed) fashion.

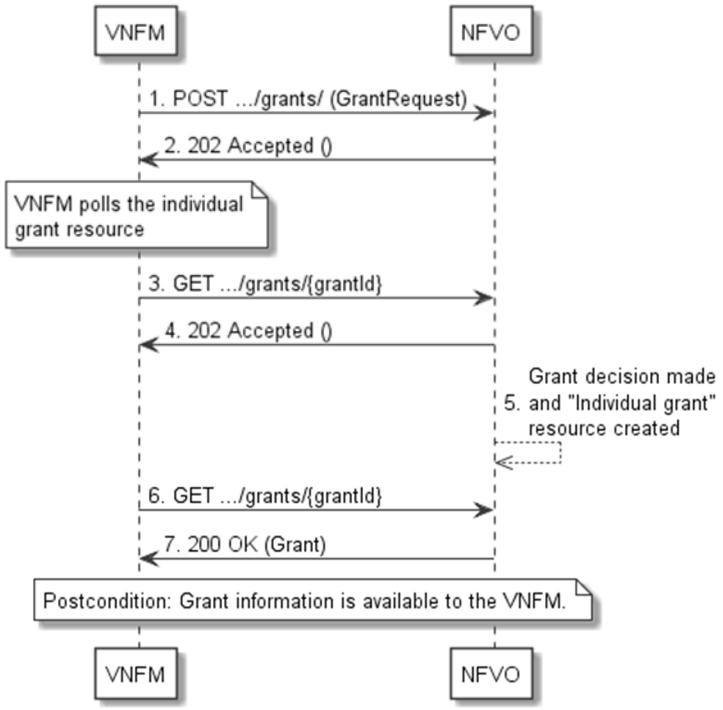

Figure 9.3.2-1: Flow of granting with asynchronous response

Granting with asynchronous response, as illustrated in figure 9.3.2-1, consists of the following steps:

1) The VNFM sends a POST request to the "Grants" resource with a "GrantRequest" data structure in the body.

2) The NFVO returns to the VNFM a "202 Accepted" response with an empty body and a "Location" HTTP header that indicates the URI of the "Individual grant" resource that will be created once the granting decision will have been made.

3) The VNFM tries to obtain the grant by sending a GET request to the NFVO, using the URI that was returned in step 2) in the "Location" header.

4) As there is no result of the granting decision available yet and consequently the "Individual grant" resource is still in the process of being created, the NFVO returns a "202 Accepted" response with an empty body.

5) The NFVO finalizes the granting decision and creates the "Individual grant" that contains the grant.

6) The VNFM tries to obtain the grant by sending a GET request to the NFVO, using the URI that was returned in step 2) in the "Location" header.

7) This time, the grant is available, and the NFVO returns a "200 OK" response with a "Grant" data structure in the body.

Postcondition: The grant information is available to the VNFM.

9.4 Resources

9.4.1 Introduction

This clause defines all the resources and methods provided by the VNF lifecycle operation granting interface.

9.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [8] are part of the VNF lifecycle operation granting interface.

9.4.2 Resource: Grants

9.4.2.1 Description

This resource represents grants. The API consumer can use this resource to obtain permission from the NFVO to perform a particular VNF lifecycle operation.

9.4.2.2 Resource definition

The resource URI is:

{apiRoot}/grant/{apiMajorVersion}/grants

This resource shall support the resource URI variables defined in table 9.4.2.2-1.

Table 9.4.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 9.1a.

9.4.2.3 Resource methods

9.4.2.3.1 POST

The POST method requests a grant for a particular VNF lifecycle operation.

This method shall follow the provisions specified in tables 9.4.2.3.1-1 and 9.4.2.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully processing this request, a new "Individual grant" resource shall be created. In the synchronous case which is indicated by responding with "201 Created", that resource shall be created before the 200 OK response is returned. In the asynchronous case which is indicated by responding with "202 Accepted", this resource may be created after the response is returned.

Table 9.4.2.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.2.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	GrantRequest	1		
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	Grant	1	201 Created	Shall be returned when the grant has been created successfully (synchronous mode). A representation of the created "Individual grant" resource shall be returned in the response body. The HTTP response shall include a "Location" HTTP header that indicates the URI of the "Individual grant" resource just created.
	n/a		202 Accepted	Shall be returned when the request has been accepted for processing, and it is expected to take some time to create the grant (asynchronous mode). The response body shall be empty. The HTTP response shall include a "Location" HTTP header that indicates the URI of the "Individual grant" resource that will be created once the granting decision has been made.
	ProblemDetails	1	403 Forbidden	Shall be returned upon the following error: The grant has been rejected. A ProblemDetails structure shall be included in the response to provide more details about the rejection in the "details" attribute.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

9.4.2.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

9.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

9.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

9.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

9.4.3 Resource: Individual grant

9.4.3.1 Description

This resource represents an individual grant. The API consumer can use this resource to read the grant.

It is determined by means outside the scope of the present document, such as configuration or policy, how long an individual grant is available.

9.4.3.2 Resource definition

The resource URI is:

{apiRoot}/grant/{apiMajorVersion}/grants/{grantId}

This resource shall support the resource URI variables defined in table 9.4.3.2-1.

Table 9.4.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 9.1a.
grantId	Identifier of the grant. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request granting a new VNF lifecycle operation. It can also be retrieved from the "id" attribute in the payload body of that response.	

9.4.3.3 Resource methods

9.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

9.4.3.3.2 GET

The GET method reads a grant.

This method shall follow the provisions specified in tables 9.4.3.3.2-1 and 9.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.3.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	Grant	1	200 OK	Shall be returned when the grant has been read successfully. A representation of the "Individual grant" resource shall be returned in the response body.
	n/a		202 Accepted	Shall be returned when the process of creating the grant is ongoing, no grant is available yet. The response body shall be empty.
	ProblemDetails	1	403 Forbidden	Shall be returned upon the following error: The grant has been rejected. A ProblemDetails structure shall be included in the response to provide more details about the rejection in the "details" attribute.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

9.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

9.4.3.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

9.4.3.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

9.5 Data model

9.5.1 Introduction

This clause defines the request and response data structures of the VNF Lifecycle Operation Granting interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error and may choose to ignore them.

9.5.2 Resource and notification data types

9.5.2.1 Introduction

This clause defines data structures to be used in resource representations and notifications.

9.5.2.2 Type: GrantRequest

This type represents a grant request. It shall comply with the provisions defined in table 9.5.2.2-1.

Table 9.5.2.2-1: Definition of the GrantRequest data type

Attribute name	Data type	Cardinality	Description
vnfInstanceId	Identifier	1	Identifier of the VNF instance which this grant request is related to. Shall also be provided for VNFs that not yet exist but are planned to exist in the future, i.e. if the grant is requested for InstantiateVNF.
vnfLcmOpOccId	Identifier	1	The identifier of the VNF lifecycle management operation occurrence associated to the GrantRequest.
vnfdId	Identifier	1	Identifier of the VNFD that defines the VNF for which the LCM operation is to be granted. In case of the "Change current VNF package operation, this identifier refers to the VNFD which defines the VNF before the LCM operation to be granted.
dstVnfdId	Identifier	0..1	Identifier of the "destination" VNFD which will define the VNF after executing the "Change current VNF package" LCM operation to be granted. Shall be included if the operation changes the current VNF Package and shall be absent otherwise.
flavourId	Identifier	0..1	Identifier of the VNF deployment flavour of the VNFD that defines the VNF for which the LCM operation is to be granted. Shall be provided when instantiating the VNF or changing the deployment flavour of the VNF instance.
operation	GrantedLcmOperationType	1	The lifecycle management operation for which granting is requested. See note 1.
isAutomaticInvocation	Boolean	1	Set to true if this VNF LCM operation occurrence has been triggered by an automated procedure inside the VNFM (i.e. ScaleVnf/ScaleVnfToLevel triggered by auto-scale, or HealVnf triggered by auto-heal). Set to false otherwise.
instantiationLevelId	Identifier	0..1	If operation=INSTANTIATE, the identifier of the instantiation level may be provided as an alternative way to define the resources to be added. This attribute shall only be used if operation=INSTANTIATE. See notes 2 and 7.
targetScaleLevelInfo	ScaleInfo	0..N	This attribute shall only be used for Instantiate VNF requests. This is applicable if VNF supports target scale level instantiation. This attribute provides an alternative way to define the resources to be added for the VNFs. For each scaling aspect of the current deployment flavour, the attribute specifies the scale level of VNF constituents (e.g. VDU level) to be instantiated. See notes 2, 7 and 8.

Attribute name	Data type	Cardinality	Description
addResources	ResourceDefinition	0..N	List of resource definitions in the VNFD for resources to be added by the LCM operation which is related to this grant request, with one entry per resource. See note 2.
tempResources	ResourceDefinition	0..N	List of resource definitions in the VNFD for resources to be temporarily instantiated during the runtime of the LCM operation which is related to this grant request, with one entry per resource. See note 3.
removeResources	ResourceDefinition	0..N	Provides the definitions of resources to be removed by the LCM operation which is related to this grant request, with one entry per resource.
updateResources	ResourceDefinition	0..N	Provides the definitions of resources to be modified by the LCM operation which is related to this grant request, with one entry per resource.
placementConstraints	PlacementConstraint	0..N	Placement constraints that the VNFM may send to the NFVO in order to influence the resource placement decision. If sent, the NFVO shall take the constraints into consideration when making resource placement decisions and shall reject the grant if they cannot be honoured. See notes 4, 5 and 6.
vimConstraints	VimConstraint	0..N	Used by the VNFM to require that multiple resources are managed through the same VIM connection. If sent, the NFVO shall take the constraints into consideration when making VIM selection decisions and shall reject the grant if they cannot be honoured. This attribute shall be supported if VNF-related Resource Management in direct mode is applicable. The applicability and further details of this attribute for indirect mode are left for future specification.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the VNFM, specific to the VNF and the LCM operation.
_links	Structure (inlined)	1	Links to resources related to this request.
>vnfLcmOpOcc	Link	1	Related lifecycle management operation occurrence.
>vnfInstance	Link	1	Related VNF instance.

Attribute name	Data type	Cardinality	Description
NOTE 1: The VNF LCM operations CreateVnfIdentifier, DeleteVnfIdentifier, QueryVnf and ModifyVnfInformation can be executed by the VNFM without requesting granting. NOTE 2: If the granting request is for InstantiateVNF, only one of the three parameters (instantiationLevel or targetScaleLevelInfo or addResources) shall be present. NOTE 3: The NFVO will assume that the VNFM will be responsible to both allocate and release the temporary resource during the runtime of the LCM operation. This means, the resource can be allocated and consumed after the "start" notification for the LCM operation is sent by the VNFM, and the resource will be released before the "result" notification of the VNF LCM operation is sent by the VNFM. NOTE 4: For the affinity/anti-affinity rules defined in the VNFD and the placement constraints in the GrantVnfLifecycleOperation as defined in this clause, the following applies: Assuming unlimited capacity, the combination of all the aforementioned rules shall be satisfiable by at least one possible placement of the new resources, with the exception that some of the rules with fallbackBestEffort may be unsatisfiable due to the actual placement of existing resources. In this case, rules with fallbackBestEffort are allowed to be violated only in relation to the placement of existing resources. NOTE 5: Passing constraints allows the VNFM or the lifecycle management scripts to influence resource placement decisions by the NFVO to ensure VNF properties such as performance or fault tolerance. NOTE 6: If fallbackBestEffort is present and set to "true" in one or more placement constraints and the NFVO cannot find a resource placement that satisfies all of these due to limited capacity, the NFVO shall process each such affinity/antiAffinity constraint in a best effort manner, in which case, if specified resources cannot be allocated based on the specified placement constraint, the NFVO shal look for an alternate best effort placement for the specified resources to be granted. In the best effort anti-affinity case, the resources are expected to be spread optimally over all available instances of scope (e.g. zones), and in the best effort affinity case, they are expected to be distributed optimally over as few instances of scope as possible. Being able to satisfy a "best-effort" constraint in a best effort manner only, as defined above, shall not lead to rejecting the grant. NOTE 7: The target size for VNF instantiation may be specified in either instantiationLevelId or targetScaleLevelInfo, but not both. NOTE 8: If targetScaleLevelInfo is specified, information provided in targetScaleLevelInfo shall be used for scalable constituents of the VNF (e.g, VDUs/VLs) in the granting process. For scaling aspects not specified in targetScaleLevelInfo or for the VNF constituents (e.g.VDUs/VLs) that are not scalable, the default instantiation level as declared in the VNFD shall be used in the granting process.			

9.5.2.3 Type: Grant

This type represents a grant. It shall comply with the provisions defined in table 9.5.2.3-1.

Table 9.5.2.3-1: Definition of the Grant data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the grant.
vnfInstanceId	Identifier	1	Identifier of the related VNF instance. See note 6.
vnfLcmOpOccId	Identifier	1	Identifier of the related VNF lifecycle management operation occurrence. See note 6.
vimConnectionInfo	map(VimConnectionInfo)	0..N	Provides information regarding VIM and/or CISM connections that are approved to be used by the VNFM to allocate resources and provides parameters of these VIM and/or CISM connections. The VNFM shall update the "vimConnectionInfo" attribute of the "VnfInstance" structure by adding unknown entries received in this attribute. This attribute is not intended for the modification of VimConnectionInfo entries passed earlier; for that, the VnfInfoModificationRequest structure shall be used. This attribute shall only be supported when

Attribute name	Data type	Cardinality	Description
			- all or part of the granted resources are managed by a VIM and VNF-related Resource Management in direct mode is applicable. - all or part of the granted resources are managed by a CISM. In direct mode, this parameter shall be absent if the VIM or CISM information was configured to the VNFM in another way, present otherwise. See note 1.
cirConnectionInfo	map(VimConnectionInfo)	0..N	Provides information regarding a CIR connection that is approved to be used by the VNFM to obtain information about OS container images. It shall be absent in case of rejection or if the CIR information was configured to the VNFM in another way or if the granted resources are not managed by a CISM, present otherwise
mciopRepositoryInfo	map(VimConnectionInfo)	0..N	Provides information regarding a MCIOP repository. It shall be absent in case of rejection or if the MCIOP repository information was configured to the VNFM in another way or if the granted resources are not managed by a CISM, present otherwise.
zones	ZoneInfo	0..N	Identifies resource zones where the resources are approved to be allocated by the VNFM.
zoneGroups	ZoneGroupInfo	0..N	Information about groups of resource zones that are related and that the NFVO has chosen to fulfil a zoneGroup constraint in the GrantVnfLifecycleOperation request. This information confirms that the NFVO has honoured the zoneGroup constraints that were passed as part of "placementConstraints" in the GrantRequest.
addResources	GrantInfo	0..N	List of resources that are approved to be added, with one entry per resource. Shall be set when resources are approved to be added and shall contain the same set of resources requested to be added in the related GrantRequest.
tempResources	GrantInfo	0..N	List of resources that are approved to be temporarily instantiated during the runtime of the lifecycle operation, with one entry per resource. Shall be set when resources are approved to be temporarily instantiated and shall contain the same set of resources requested to be temporarily instantiated in the related GrantRequest.

Attribute name	Data type	Cardinality	Description
removeResources	GrantInfo	0..N	List of resources that are approved to be removed, with one entry per resource. Shall be set when resources are approved to be removed and shall contain the same set of resources requested to be removed in the related GrantRequest.
updateResources	GrantInfo	0..N	List of resources that are approved to be modified, with one entry per resource. Shall be set when resources are approved to be updated and shall contain the same set of resources requested to be updated in the related GrantRequest.
vimAssets	Structure (inlined)	0..1	Information about assets for the VNF that are defined in VNFD and managed by the NFVO in the VIM/NFVI, such as software images and virtualised compute resource flavours. See note 3.
>computeResourceFlavours	VimComputeResourceFlavour	0..N	Mappings between virtual compute descriptors defined in the VNFD and compute resource flavours managed in the VIM.
>softwareImages	VimSoftwareImage	0..N	Mappings between software images defined in the VNFD and software images managed in the VIM.
>snapshotResources	VimSnapshotResource	0..N	Mappings between snapshot resources defined in the VNF snapshot package and resources managed in the VIM.
>storageAssets	StorageAsset	0..N	Mappings between virtual storages defined in the VNFD and virtual storages managed in the NFVI.
extVirtualLinks	ExtVirtualLinkData	0..N	Information about external VLs to connect the VNF to. See notes 5, 7 and 8. If this attribute is present according to note 5 or note 7, it need not contain those entries that are unchanged compared to the entries that were passed in the LCM operation which is related to this granting exchange.
extManagedVirtualLinks	ExtManagedVirtualLinkData	0..N	Information about internal VLs that are managed by other entities than the VNFM. See notes 4, 5, 7 and 8.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the NFVO, specific to the VNF and the LCM operation.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>vnfLcmOpOcc	Link	1	Related VNF lifecycle management operation occurrence.
>vnfInstance	Link	1	Related VNF instance.

Attribute name	Data type	Cardinality	Description

NOTE 1: This interface allows to signal the use of multiple VIMs per VNF. The specification for managing the VNF-internal VL requirements across multiple VIMs is supported via "externally-managed internal VLs" and "multi-site connectivity services".

NOTE 2: Void.

NOTE 3: The Grant response allows the NFVO to pass to the VNFM VIM assets related to the VNF package that is identified in the corresponding Grant request. Because only the operations InstantiateVnf, ChangeCurrentVnfPkg and the update of the vnfdId by PATCH can imply changes in the set of VIM assets, the NFVO shall not change this set in any other case. To signal the set of VIM assets, the following applies:

a) If the current LCM operation is InstantiateVnf, the NFVO shall send in the Grant response the full set of VIM assets related to the VNF package defined by the vnfdId in the related Grant request.
b) If the current LCM operation is ChangeCurrentVnfPkg, the NFVO shall send in the Grant response the full set of VIM assets related to the VNF package defined by the destVnfdId in the related Grant request.
c) For any other operation: If the set of VIM assets has changed since the end of the previous granted operation because a PATCH operation has changed the vnfdId between the previous granted operation and the operation to which the current granting exchange relates, the NFVO shall send in the Grant response the full set of VIM assets related to the VNF package defined by the vnfdId in the related Grant request. Otherwise, the NFVO shall either send in the Grant response the full set of VIM assets related to the VNF package defined by the vnfdId in the related Grant request, or shall not send any VIM assets at all.

During each LCM operation occurrence other than a "ChangeCurrentVnfPkg" operation, the VIM assets that relate to the VNF package identified by the current value of the vnfdId attribute in the "VnfInstance" structure shall be used by the VNFM for newly created and updated resources.

NOTE 4: The indication of externally-managed internal VLs is needed in case networks have been pre-configured for use with certain VNFs, for instance to ensure that these networks have certain properties such as security or acceleration features, or to address particular network topologies (e.g. multi-site connectivity). The present document assumes that externally-managed internal VLs are managed by the NFVO and created towards the VIM and WIM.

NOTE 5: For any VNF lifecycle management operation request that allows to pass "extVirtualLinks" and/or "extManagedVirtualLinks" parameters, such as InstantiateVnf, ChangeVnfFlavour, ChangeExtVnfConnectivity or ChangeCurrentVnfPackage, the NFVO may provide the "extVirtualLinks" and/or "extManagedVirtualLinks" attributes in the Grant to override the values passed in these parameters previously in the associated VNF lifecycle management request, if the lifecycle management request has originated from the NFVO itself. The NFVO shall not provide the "extVirtualLinks" and/or "extManagedVirtualLinks" attributes in the Grant in case the LCM request did not originate from the NFVO itself or the granted LCM operation request does not allow to pass these parameters.

NOTE 6: The NFVO shall set the value of the attribute by copying the value from the associated GrantRequest.

NOTE 7: In case of granting an InstantiateVnf request that has originated from the NFVO and that did not contain the "extVirtualLinks" attribute, this attribute shall be set by the NFVO. Further in case of granting an InstantiateVnf request that has originated from the NFVO and that did not contain the "extManagedVirtualLinks" attribute, this attribute shall be set by the NFVO if there is the need to provide information about externally-managed virtual links.

NOTE 8: The resulting "extVirtualLinks" or "extManagedVirtualLinks" information is calculated as follows: In a first step, the information passed in the related LCM operation is applied to the baseline information known from earlier LCM operation executions. In a second step, the information passed in the Grant is applied to the information resulting from the first step.

9.5.3 Referenced structured data types

9.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but are not resource representations.

9.5.3.2 Type: ResourceDefinition

This type provides information of an existing or proposed resource used by the VNF. It shall comply with the provisions defined in table 9.5.3.2-1.

Table 9.5.3.2-1: Definition of the ResourceDefinition data type

Attribute name	Data type	Cardinality	Description
id	IdentifierLocal	1	Identifier of this "ResourceDefinition" structure, unique at least within the scope of the "GrantRequest" structure.
type	Enum (inlined)	1	Type of the resource definition referenced. Permitted values: - COMPUTE - VL - STORAGE - LINKPORT - OSCONTAINER - VIRTUALCP
vduId	IdentifierInVnfd	0..1	Reference to the related VDU in the VNFD applicable to this resource. Shall only be present if a VDU is applicable to this resource, i.e. if "type" has the value "COMPUTE".
vnfdId	Identifier	0..1	Identifier of the VNFD to which resourceTemplateId and vduId refer. Shall be present if the operation to be granted changes the current VNF Package. May be absent otherwise. See note 2.
resourceTemplateId	IdentifierInVnfd	1..N	Reference to the applicable resource template in the VNFD as follows: - If type="VL": VnfVirtualLinkDesc - If type="COMPUTE": VirtualComputeDesc, - If type="LINKPORT": VnfExtCpd, - If type="STORAGE": VirtualStorageDesc - If type="OSCONTAINER": OsContainerDesc Cardinality may be greater than "1" when type="OSCONTAINER" and multiple references to OsContainerDesc are present in the VDU indicated by the "vduId". Cardinality shall be "1" otherwise.
secondaryResourceTemplateId	IdentifierInVnfd	0..1	Reference to a secondary resource template (VnfExtCpd) in the VNFD. Shall be present if type="LINKPORT" and the linkport is shared by two external CP instances, one exposing a VNFC CP instance (based on a VnfExtCpd referenced by "resourceTemplateId") and another one exposing a VIP CP instance (based on a VnfExtCpd referenced by this attribute). Shall be absent otherwise. See note 1.
resource	ResourceHandle	0..1	Resource information for an existing resource. Shall be present for resources that are planned to be deleted or modified. Shall be absent otherwise.
snapshotResDef	SnapshotResourceDefinition	0..1	Information to identify a snapshot resource. Shall only be present if the operation to be granted concerns to creating a VNF snapshot from the VNF or to reverting the VNF to a VNF snapshot.

NOTE 1: The use cases UC#4 and UC#5 in clause A.4 of ETSI GS NFV-IFA 007 [1] provide examples for such a configuration.

NOTE 2: In the context of an operation that changes the current VNF package, the following applies: If this ResourceDefinition is related to a resource to be created or modified, the "vnfdId" attribute shall contain the identifier of the destination VNFD. If this ResourceDefinition is related to a resource to be deleted, the "vnfdId" attribute shall contain the identifier of the source VNFD. If this ResourceDefinition is related to a temporary resource, the "vnfdId" attribute shall contain the identifier of either the source VNFD or the destination VNFD.

9.5.3.3 Type: GrantInfo

This type contains information about a Compute, storage or network resource whose addition/update/deletion has been granted. It shall comply with the provisions defined in table 9.5.3.3-1.

Table 9.5.3.3-1: Definition of the GrantInfo data type

Attribute name	Data type	Cardinality	Description
resourceDefinitionId	IdentifierLocal	1	Identifier of the related "ResourceDefinition" structure from the related "GrantRequest" structure.
reservationId	IdentifierInVim	0..1	The reservation identifier applicable to the VNFC/VirtualLink/VirtualStorage/compute host. It shall be present for new resources when policy is GRANT_RESERVE and an applicable reservation exists; shall not be present otherwise.
vimConnectionId	Identifier	0..1	Identifier of the VIM or CISM connection to be used to manage this resource. Shall be present for new resources and shall be absent for resources that have already been allocated. The applicable "VimConnectionInfo" structure, which is referenced by vimConnectionId, can be obtained from the "vimConnectionInfo" attribute of the "VnfInstance" structure. This attribute shall only be supported when - all or part of the granted resources are managed by a VIM and VNF-related Resource Management in direct mode is applicable. - all or part of the granted resources are managed by a CISM.
resourceProviderId	Identifier	0..1	Identifies the entity responsible for the management of the virtualised resource. Shall be present for new resources and shall be absent for resources that have already been allocated. This attribute shall only be supported when VNF-related Resource Management in indirect mode is applicable. The identification scheme is outside the scope of the present document.
zoneId	IdentifierLocal	0..1	Reference to the identifier of the "ZoneInfo" structure in the "Grant" structure defining the resource zone into which this resource is to be placed. Shall be present for new resources if the zones concept is applicable to them (typically, Compute resources) and shall be absent for resources that have already been allocated.
resourceGroupId	IdentifierInVim	0..1	Identifier of the "infrastructure resource group", logical grouping of virtual resources assigned to a tenant within an Infrastructure Domain, to be provided when allocating the resource. If the VIM connection referenced by "vimConnectionId" applies to multiple infrastructure resource groups, this attribute shall be present for new resources. If the VIM connection referenced by "vimConnectionId" applies to a single infrastructure resource group, this attribute may be present for new resources. This attribute shall be absent for resources that have already been allocated.

Attribute name	Data type	Cardinality	Description
containerNamespace	String	0..1	The value of the namespace in which the MCIOs of a VNF with containerized components shall be deployed. This attribute shall be present if the granted resources are managed by a CISM. The attribute shall be absent if the granted resources are not managed by a CISM.
mcioConstraints	KeyValuePair	0..N	The constraint values to be assigned to MCIOs of a VNF with containerized components. The key in the key-value pair indicates the parameter name of the MCIO constraint in the MCIO declarative descriptor and shall be one of the possible enumeration values of the "mcioConstraintsParams" attribute as specified in clause 7.1.6.2.2 of ETSI GS NFV-IFA 011 [10]. The value in the key-value pair indicates the value to be assigned to the MCIO constraint. This attribute shall be present if the granted resources are managed by a CISM. The attribute shall be absent if the granted resources are not managed by a CISM.

9.5.3.4 Type: ZoneInfo

This type provides information regarding a resource zone. It shall comply with the provisions defined in table 9.5.3.4-1.

Table 9.5.3.4-1: Definition of the ZoneInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierLocal	1	The identifier of this ZoneInfo instance, for the purpose of referencing it from other structures in the "Grant" structure.
zoneId	Identifier	1	The identifier of the resource zone, as managed by the resource management layer (typically, the VIM).
vimConnectionId	Identifier	0..1	Identifier of the connection to the VIM that manages the resource zone. The applicable "VimConnectionInfo" structure, which is referenced by vimConnectionId, can be obtained from the "vimConnectionInfo" attribute of the "VnfInstance" structure. This attribute shall only be supported and present when VNF-related Resource Management in direct mode is applicable.
resourceProviderId	Identifier	0..1	Identifies the entity responsible for the management the resource zone. This attribute shall only be supported and present when VNF-related Resource Management in indirect mode is applicable. The identification scheme is outside the scope of the present document.

9.5.3.5 Type: ZoneGroupInfo

This type provides information regarding a resource zone group. A resource zone group is a group of one or more related resource zones which can be used in resource placement constraints. To fulfil such constraint, the NFVO may decide to place a resource into any zone that belongs to a particular group.

NOTE: A resource zone group can be used to support overflow from one resource zone into another, in case a particular deployment supports only non-elastic resource zones.

The ZoneGroupInfo type shall comply with the provisions defined in table 9.5.3.5-1.

Table 9.5.3.5-1: Definition of the ZoneGroupInfo data type

Attribute name	Data type	Cardinality	Description
zoneId	IdentifierLocal	1..N	References of identifiers of "ZoneInfo" structures, each of which provides information about a resource zone that belongs to this group.

9.5.3.6 Type: PlacementConstraint

This type provides information regarding a resource placement constraint. A set of such constraints may be sent by the VNFM to the NFVO to influence the resource placement decisions made by the NFVO as part of the granting process. A placement constraint defines a condition to the placement of new resources, considering other new resources as well as existing resources.

EXAMPLE: The following rules influence the placement of a set of resources such that they are placed in the same Network Function Virtualisation Infrastructure Point of Presence (NFVI-PoP) but in different resource zones:

```
{type="AFFINITY"; scope="NFVI_POP"; {resource1,resource2}}
{type="ANTI_AFFINITY"; scope="ZONE"; {resource1,resource2}}
```

Annex B in ETSI GS NFV-IFA 011 [10] provides additional description and examples about the usage of the affinity/anti-affinity rules.

The PlacementConstraint type shall comply with the provisions defined in table 9.5.3.6-1.

Table 9.5.3.6-1: Definition of the PlacementConstraint data type

Attribute name	Data type	Cardinality	Description
affinityOrAntiAffinity	Enum (inlined)	1	The type of the constraint. Permitted values: - AFFINITY - ANTI_AFFINITY
scope	Enum (inlined)	1	The scope of the placement constraint indicating the category of the "place" where the constraint applies. Permitted values: - NFVI_POP - ZONE - ZONE_GROUP - NFVI_NODE - CIS_NODE - CONTAINER_NAMESPACE See note.
resource	ConstraintResourceRef	2..N	References to resources in the constraint rule.
fallbackBestEffort	Boolean	0..1	Indication if the constraint is handled with fall back best effort. Default value is "false". If set to true, the Affinity/Anti_Affinity placement constraint need not be fully satisfied due to capacity constraints and/or due to the actual placement of existing resources, i.e. if resource placement cannot honour the placement constraint, the request is processed in a best effort manner.
NOTE: The "CIS_NODE" and "CONTAINER_NAMESPACE" scopes shall only be used to express affinity or anti-affinity relationship between containerized workloads.			

9.5.3.7 Type: VimConstraint

This type provides information regarding a VIM selection constraint. A set of such constraints may be sent by the VNFM to the NFVO to influence the VIM selection decisions made by the NFVO as part of the granting process.

The VimConstraint type shall comply with the provisions defined in table 9.5.3.7-1.

Table 9.5.3.7-1: Definition of the VimConstraint data type

Attribute name	Data type	Cardinality	Description
sameResourceGroup	Boolean	0..1	If present and set to true, this signals that the constraint applies not only to the same VIM connection, but also to the same infrastructure resource group.
resource	ConstraintResourceRef	2..N	References to resources in the constraint rule. The NFVO shall ensure that all resources in this list are managed through the same VIM connection. If "sameResourceGroup" is set to true, the NFVO shall further ensure that all resources in this list are part of the same infrastructure resource group in that VIM connection.

9.5.3.8 Type: ConstraintResourceRef

This type references a resource either by its VIM-level identifier for existing resources, or by the identifier of a "ResourceDefinition" structure in the "GrantRequest" structure for new resources.

The ConstraintResourceRef type shall comply with the provisions defined in table 9.5.3.8-1.

Table 9.5.3.8-1: Definition of the ConstraintResourceRef data type

Attribute name	Data type	Cardinality	Description
idType	Enum (inlined)	1	The type of the identifier. Permitted values: - RES_MGMT: Resource-management-level identifier; this identifier is managed by the VIM in the direct mode of VNF-related resource management, and is managed by the NFVO in the indirect mode) - GRANT: Reference to the identifier of a "ResourceDefinition" structure in the "GrantRequest" structure.
resourceId	IdentifierInVim	1	An actual resource-management-level identifier (idType=RES_MGMT), or an identifier that references a "ResourceDefinition" structure in the related "GrantRequest" structure (idType=GRANT).
vimConnectionId	Identifier	0..1	Identifier of the VIM connection for managing the resource. It shall only be present when idType = RES_MGMT. The applicable "VimConnectionInfo" structure, which is referenced by vimConnectionId, can be obtained from the "vimConnectionInfo" attribute of the "VnfInstance" structure. This attribute shall only be supported when VNF-related resource management in direct mode is applicable.

Attribute name	Data type	Cardinality	Description
resourceProviderId	Identifier	0..1	Identifier of the resource provider. It shall only be present when idType = RES_MGMT. This attribute shall only be supported when VNF-related resource management in indirect mode is applicable. The identification scheme is outside the scope of the present document.

9.5.3.9 Type: VimComputeResourceFlavour

If the VIM requires the use of virtual compute resource flavours during compute resource instantiation, it is assumed that such flavours are selected or created by the NFVO based on the information in the virtual compute descriptor defined in the VNFD.

This type defines the mapping between a virtual compute descriptor in the VNFD and the corresponding compute resource flavour managed by the NFVO in the VIM. It shall comply with the provisions defined in table 9.5.3.9-1.

Table 9.5.3.9-1: Definition of the VimComputeResourceFlavour data type

Attribute name	Data type	Cardinality	Description
vimConnectionId	Identifier	0..1	Identifier of the VIM connection to access the flavour referenced in this structure. The applicable "VimConnectionInfo" structure, which is referenced by vimConnectionId, can be obtained from the "vimConnectionInfo" attribute of the "VnfInstance" structure. This attribute shall only be supported and present if VNF-related resource management in direct mode is applicable.
resourceProviderId	Identifier	0..1	Identifies the entity responsible for the management of the virtualised resource. This attribute shall only be supported and present if VNF-related resource management in indirect mode is applicable. The identification scheme is outside the scope of the present document.
vnfdVirtualComputeDescId	IdentifierInVnfd	1	Identifier which references the virtual compute descriptor in the VNFD that maps to this flavour.
vimFlavourId	IdentifierInVim	1	Identifier of the compute resource flavour in the resource management layer (i.e. VIM).

9.5.3.10 Type: VimSoftwareImage

This type contains a mapping between a software image definition the VNFD and the corresponding software image managed by the NFVO in the VIM or CIR which is needed during compute resource instantiation. It shall comply with the provisions defined in table 9.5.3.10-1.

Table 9.5.3.10-1: Definition of the VimSoftwareImage data type

Attribute name	Data type	Cardinality	Description
vimConnectionId	Identifier	0..1	Identifier of the VIM or CIR connection to access the software image referenced in this structure. The applicable "VimConnectionInfo" structure, which is referenced by vimConnectionId, can be obtained from the "vimConnectionInfo" attribute of the "VnfInstance" structure. This attribute shall only be supported and present if - the software image is to be loaded in a virtual machine and VNF-related resource management in direct mode is applicable. - the software image is to be loaded in an OS container managed by a CISM.

Attribute name	Data type	Cardinality	Description
resourceProviderId	Identifier	0..1	Identifies the entity responsible for the management of the virtualised resource. This attribute shall only be supported and present if VNF-related resource management in indirect mode is applicable. The identification scheme is outside the scope of the present document.
vnfdSoftwareImageId	IdentifierInVnfd	1	Identifier which references the software image descriptor in the VNFD.
vimSoftwareImageId	IdentifierInVim	1	Identifier of the software image in the resource management layer (i.e. VIM). See note.
NOTE: For an OS container image, the value of this attribute is a string concatenating the name and tag of the image in the CIR separated by a colon ':' with no spaces, e.g. "dbImage:001".			

9.5.3.11 Type: SnapshotResourceDefinition

This type represents resource definition information related to a snapshot resource. It shall comply with the provisions defined in table 9.5.3.11-1.

Table 9.5.3.11-1: Definition of the SnapshotResourceDefinition data type

Attribute name	Data type	Cardinality	Description
vnfSnapshotId	Identifier	1	Identifier of the VNF snapshot related to the resource change for the VNF instance. Shall only be present if the operation to be granted concerns to creating a VNF snapshot from the VNF or to reverting the VNF to a VNF snapshot.
vnfcSnapshotId	IdentifierLocal	0..1	Reference to the information about a specific VNFC snapshot (refer to "VnfcSnapshotInfo") of the VNF snapshot. The identifier is unique within the scope of a VNF snapshot, identified by the "vnfSnapshotId" attribute. Shall only be present if the operation to be granted concerns to reverting the VNF to a VNF snapshot, and the resource is planned to be added based on the VNFC snapshot, and the type of resource is "COMPUTE" or "STORAGE". See notes 1 and 2.
storageSnapshotId	IdentifierInVnf	0..1	Reference to a snapshotted storage resource associated to the VNFC snapshot. Shall only be present if the operation to be granted concerns to reverting the VNF to a VNF snapshot, and the storage resource is planned to be added based on the VNFC snapshot, and the type of resource is "STORAGE". See note 2.
snapshotResource	ResourceHandle	0..1	Resource information for an existing snapshot resource. Shall only be present if the operation to be granted concerns to reverting the VNF to a VNF snapshot and the resource is planned to be added based on an existing VNF snapshot that has been created by the VNFM. Shall be absent otherwise. See note 2.
NOTE 1: If present, the value of the "vduId" (for a related VDU) in the "VnfcResourceInfo" referred by the "vnfcInfoId" of the "VnfcSnapshotInfo" shall match the value of the "vduId" in the resource definition that is signalled in the granting request. NOTE 2: For snapshot resource definitions extracted from a VNF snapshot package, only the "vnfcSnapshotId" and "storageSnapshotId" (in case of a storage type of resource) are applicable. If the snapshot resource definition is generated as part of a VNF snapshot created by the VNFM (that is, not extracted from a VNF snapshot package), the "snapshotResource" is applicable. This is a similar specification as the one defined with the "vduId", "resourceTemplateId" and "resource" attributes provided in the ResourceDefinition, but in this case applicable to resources that are defined from VNF snapshots instead of VNFD.			

9.5.3.12 Type: VimSnapshotResource

This type contains a mapping between a snapshot resource definition related to a VNF snapshot and the corresponding resource managed by the NFVO in the VIM which is needed during the revert to VNF snapshot operation. It shall comply with the provisions defined in table 9.5.3.12-1.

Table 9.5.3.12-1: Definition of the VimSnapshotResource data type

Attribute name	Data type	Cardinality	Description
vimConnectionId	Identifier	0..1	Identifier of the VIM connection to access the software image referenced in this structure. The applicable "VimConnectionInfo" structure, which is referenced by vimConnectionId, can be obtained from the "vimConnectionInfo" attribute of the "VnfInstance" structure. This attribute shall only be supported and present if VNF-related resource management in direct mode is applicable.
resourceProviderId	Identifier	0..1	Identifies the entity responsible for the management of the virtualised resource. This attribute shall only be supported and present if VNF-related resource management in indirect mode is applicable. The identification scheme is outside the scope of the present document.
vnfSnapshotId	Identifier	1	Identifier of the VNF snapshot (referring to the "id" attribute in the "VnfSnapshot" data structure) related to this VIM snapshot resource.
vnfcSnapshotId	IdentifierLocal	1	Identifier of the information about a specific VNFC snapshot (refer to "VnfcSnapshotInfo") of the VNF snapshot. The identifier is unique within the scope of a VNF snapshot, identified by the "vnfSnapshotId" attribute.
storageSnapshotId	IdentifierInVnf	0..1	Identifier of the virtual storage resource that has been snapshotted as referred in the VNFC snapshot information. Shall only be present if the snapshot resource in the VIM is a storage resource (as indicated by "type=STORAGE" in the parent resource definition).
vimSnapshotResourceId	IdentifierInVim	1	Identifier of the snapshot resource in the resource management layer (i.e. VIM).

9.5.3.13 Type: StorageAsset

This type provides a mapping between a VirtualStorageDesc in the VNFD and the corresponding virtual storage managed by the NFVO in the NFVI. It shall comply with the provisions defined in table 9.5.3.13-1.

Table 9.5.3.13-1: Definition of the StorageAsset data type

Attribute name	Data type	Cardinality	Description
vimConnectionId	Identifier	0..1	Identifier of the VIM or CISM connection to access the virtual storage referenced in this data type. The applicable "VimConnectionInfo" structure, which is referenced by vimConnectionId, can be obtained from the "vimConnectionInfo" attribute of the "VnfInstance" structure. This attribute shall only be supported and present if - all or part of the granted resources are managed by a VIM and VNF related resource management in direct mode is applicable. - all or part of the granted resources are managed by a CISM.

Attribute name	Data type	Cardinality	Description
resourceProviderId	Identifier	0..1	Identifier used by NFVO to determine the entity responsible for the management of the storage asset. This attribute shall only be supported and present if VNF-related resource management in indirect mode is applicable. The identification scheme is outside the scope of the present document.
virtualStorageDescId	IdentifierInVnfd	1	Identifier of the virtual storage descriptor in the VNFD.
storageClassName	String	1	Name of storage class, which represents features and policies concerning a virtual storage.

9.5.4 Referenced simple data types and enumerations

9.5.4.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

9.5.4.2 Simple data types

No particular simple data types are defined for this interface, in addition to those defined in clause 4.4.

9.5.4.3 Enumeration: GrantedLcmOperationType

The enumeration GrantedLcmOperationType defines the permitted values to represent VNF lifecycle operation types in grant requests. It shall comply with the provisions defined in table 9.5.4.3-1.

Table 9.5.4.3-1: Enumeration GrantedLcmOperationType

Enumeration value	Description
INSTANTIATE	Represents the "Instantiate VNF" LCM operation.
SCALE	Represents the "Scale VNF" LCM operation.
SCALE_TO_LEVEL	Represents the "Scale VNF to Level" LCM operation.
CHANGE_FLAVOUR	Represents the "Change VNF Flavour" LCM operation.
TERMINATE	Represents the "Terminate VNF" LCM operation.
HEAL	Represents the "Heal VNF" LCM operation.
OPERATE	Represents the "Operate VNF" LCM operation.
CHANGE_EXT_CONN	Represents the "Change external VNF connectivity" LCM operation.
CHANGE_VNFPKG	Represents the "Change current VNF package" LCM operation.
CREATE_SNAPSHOT	Represents the "Create VNF snapshot" LCM operation.
REVERT_TO_SNAPSHOT	Represents the "Revert to VNF snapshot" LCM operation.

10 VNF Package Management interface

10.1 Description

This interface allows the VNFM to obtain VNF package information from the NFVO, and to retrieve API version information.

The operations provided through this interface are:

- Query VNF Package, including obtaining the VNFD

- Fetch VNF Package
- Fetch VNF Package Artifacts
- Subscribe
- Query Subscription Info
- Terminate Subscription
- Notify

10.1a API version

For the VNF package management interface version as specified in the present document, the MAJOR version field shall be 2, the MINOR version field shall be 10 and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [8] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v2".

10.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [8]. The string "vnfpkgm" shall be used to represent {apiName}. All resource URIs in the clauses below are defined relative to the above base URI.

Due to the specific structure how VNF packages are identified, there are two resource sub-trees with identical structure provided which only differ in the identifier per "Individual VNF package" resource. VNF packages can be identified by an NFVO-managed identifier known as vnfPkgId which is assigned during the VNF Package onboarding process, or by an identifier known as vnfdId defined by the VNF vendor during VNF package creation. The set of packages identified by the vnfdId is a subset of the VNF packages identified by the vnfPkgId, containing all those packages that have completed their onboarding process and are available for use by the VNFM.

For any given vnfdId value, there shall be at most one associated vnfPkgId value in the whole resource tree visible to the VNFM.

Figure 10.2-1 shows the overall resource URI structure defined for the VNF Package Management interface.

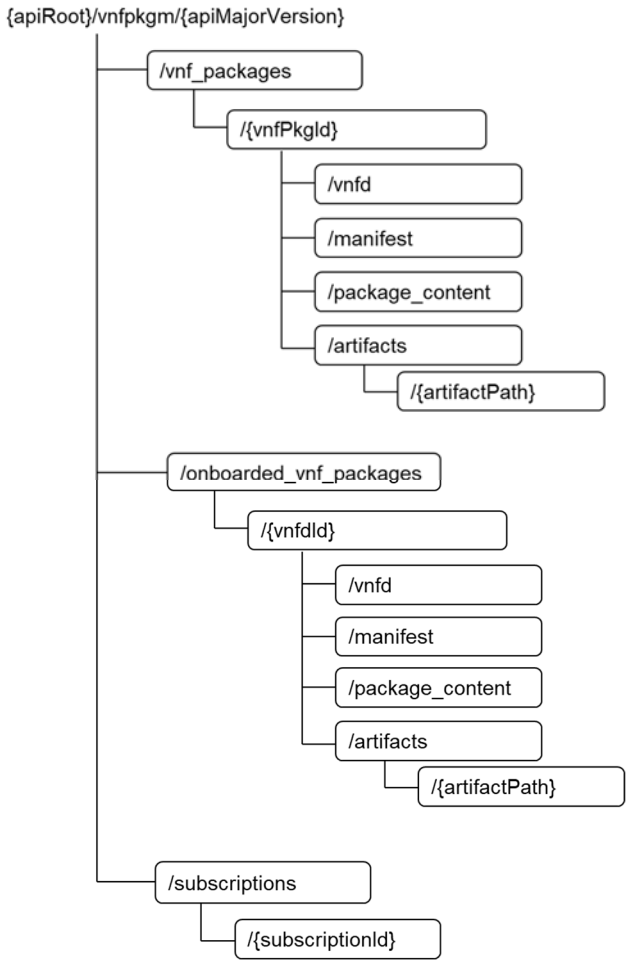

Figure 10.2-1: Resource URI structure of the VNF Package Management Interface

Table 10.2-1 lists the individual resources defined, and the applicable HTTP methods.

The NFVO shall support responding to requests for all HTTP methods on the resources in table 10.2-1 that are marked as "M" (mandatory) in the "Cat" column. The NFVO shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [8]. The "vnf_packages" subtree is deprecated; support can be made optional or removed in subsequent versions of the present document.

Table 10.2-1: Resources and methods overview of the VNF Package Management interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
VNF packages	/vnf_packages /onboarded_vnf_packages	GET	M M	Query VNF packages information
Individual VNF package	/vnf_packages/{vnfPkgId} /onboarded_vnf_packages/{vnfdId}	GET	M M	Read information about an individual VNF package
VNFD of an individual VNF package	/vnf_packages/{vnfPkgId}/vnfd /onboarded_vnf_packages/{vnfdId}/vnfd	GET	M M	Read VNFD of an on-boarded VNF package
Manifest of an individual VNF package	/vnf_packages/{vnfPkgId}/manifest /onboarded_vnf_packages/{vnfdId}/manifest	GET	M M	Read the manifest of an on-boarded VNF package
VNF package content	/vnf_packages/{vnfPkgId}/package_content /onboarded_vnf_packages/{vnfdId}/package_content	GET	M M	Fetch an on-boarded VNF package
VNF package artifacts	/vnf_packages/{vnfPkgId}/artifacts/ /onboarded_vnf_packages/{vnfdId}/artifacts/	GET	M M	Bulk-fetch artifacts that are not images.

Resource name	Resource URI	HTTP Method	Cat	Meaning
Individual VNF package artifact	/vnf_packages/{vnfPkgId}/artifacts/{artifactPath} /onboarded_vnf_packages/{vnfdId}/artifacts/{artifactPath}	GET	M M	Fetch individual VNF package artifact
Subscriptions	/subscriptions	POST	M	Subscribe to notifications related to on-boarding and/or changes of VNF packages
		GET	M	Query multiple subscriptions
Individual subscription	/subscriptions/{subscriptionId}	GET	M	Read information about an individual subscription
		DELETE	M	Terminate a subscription
Notification endpoint	(provided by API consumer)	POST	See note	Notify about VNF package on-boarding or change.
		GET	See note	Test the notification endpoint.
NOTE: The NFVO shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the VNFM. If the VNFM supports invoking the POST method on the "Subscriptions" resource towards the NFVO, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.				

10.3 Sequence diagrams (informative)

10.3.1 Flow of querying/reading VNF package information

This clause describes a sequence for querying information about one or multiple VNF packages.

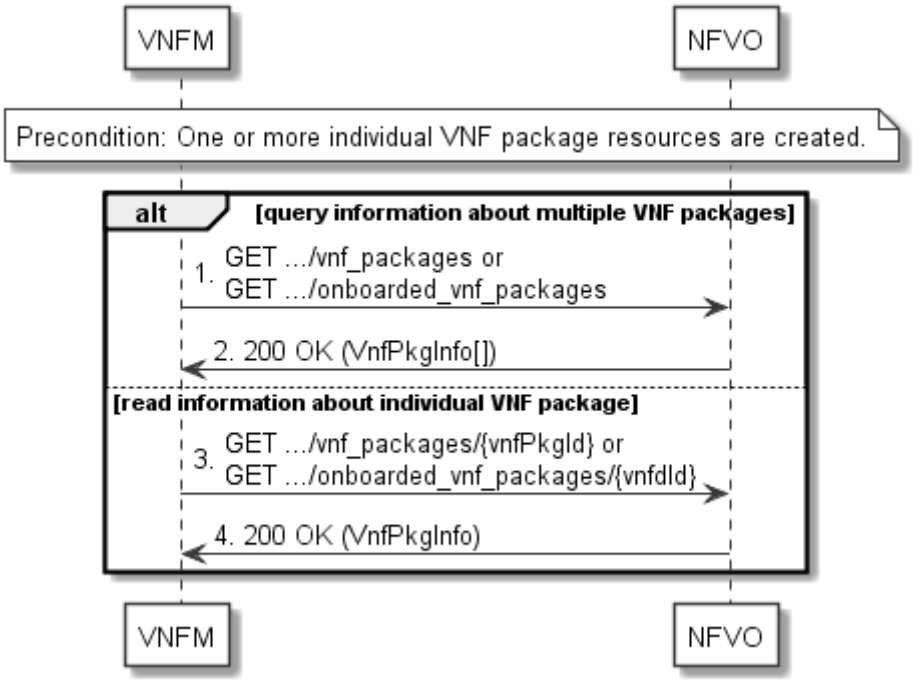

Figure 10.3.1-1: Flow of querying/reading VNF package information

Precondition: One or more "Individual VNF package" resources are created.

VNF package information query, as illustrated in figure 10.3.1-1, consists of the following steps:

1) If the VNFM intends to query information about multiple VNF packages, it sends a GET request to the "VNF packages" resource.

2) The NFVO returns a "200 OK" response and includes in the payload body zero or more data structures of type "VnfPkgInfo".

3) If the VNFM intends to read information about a particular VNF package, the VNFM sends a GET request to the "Individual VNF package" resource, addressed by the appropriate VNF package identifier in its resource URI.

4) The NFVO returns a "200 OK" response and includes in the payload body a data structure of type "VnfPkgInfo".

Postcondition: Upon successful completion, the VNFM gets the information of the VNF packages or the VNF package.

Error handling: In case of failure, appropriate error information is provided in the response.

10.3.2 Flow of reading the VNFD of an on-boarded VNF package

This clause describes the procedure for reading the VNFD of an on-boarded VNF package.

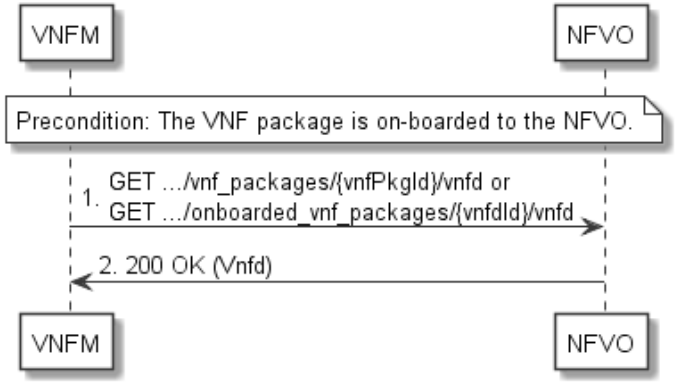

Figure 10.3.2-1: Flow of reading VNFD

Precondition: The VNF package is on-boarded to the NFVO.

The procedure consists of the following steps as illustrated in figure 10.3.2-1:

1) The VNFM sends a GET request to the "VNFD in an individual VNF package" resource.

2) The NFVO returns a "200 OK" response and includes a copy of the VNFD from the VNF package in the payload body.

10.3.2a Flow of fetching the VNF package manifest

This clause describes a sequence for fetching the VNF package manifest.

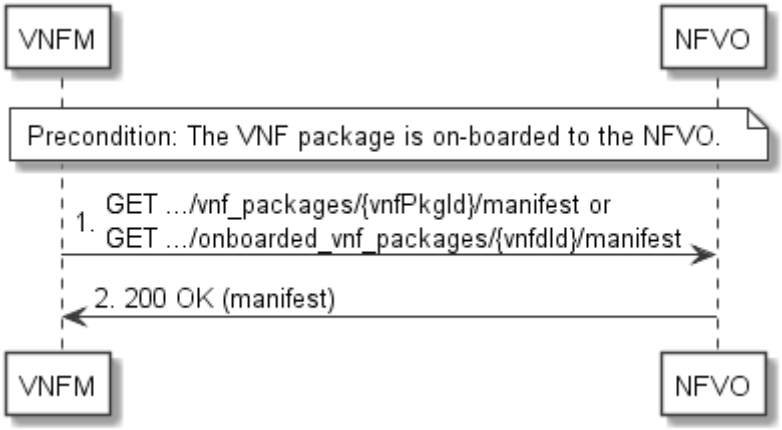

Figure 10.3.2a-1: Flow of fetching the VNF package manifest

Precondition: The VNF package is on-boarded to the NFVO.

Reading the manifest of an on-boarded VNF package, as illustrated in figure 10.3.2a-1, consists of the following steps:

1) The VNFM sends a GET request to the "Manifest of an individual VNF package" resource.

2) The NFVO returns a "200 OK" response with a payload body that contains a copy of the manifest file in the VNF package.

Error handling: In case of failure, appropriate error information is provided in the response.

10.3.3 Flow of fetching an on-boarded VNF package

This clause describes a sequence for fetching the content of an on-boarded VNF package.

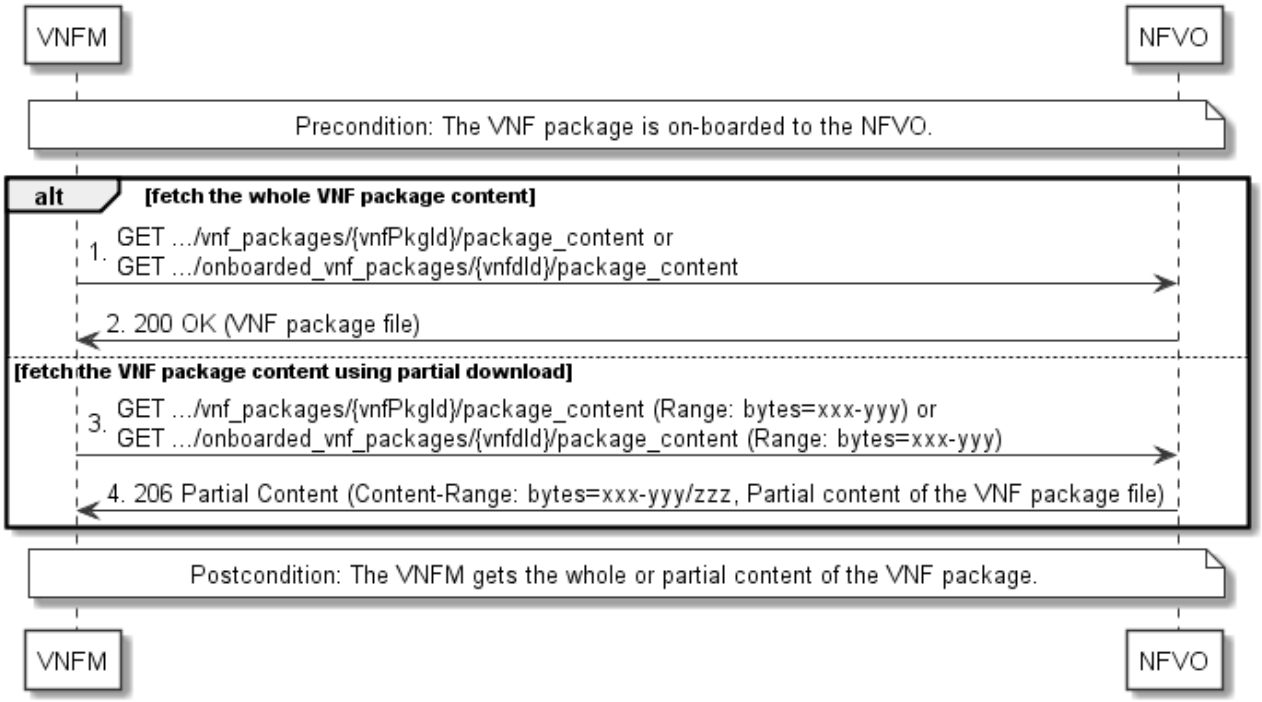

Figure 10.3.3-1: Flow of fetching an on-boarded VNF package

Precondition: The VNF package is on-boarded to the NFVO.

Fetching an on-boarded VNF package, as illustrated in figure 10.3.3-1, consists of the following steps:

1) If fetching the whole VNF package content, the VNFM sends a GET request to the "VNF package content" resource.

2) The NFVO returns a "200 OK" response and includes a copy of the VNF package file in the payload body.

3) If fetching the VNF package content using partial download, the VNFM sends a GET request to the "VNF package content" resource and includes a "Range" HTTP header indicating the partition of the VNF package content needs to be transferred.

4) The NFVO returns a "206 Partial Content" response with a payload body containing the partial content of the VNF package and a "Content-Range" HTTP header indicating the byte range enclosed in the payload and the complete length of the VNF package content.

Postcondition: Upon successful completion, the VNFM gets the whole or partial content of the VNF package.

Error handling: In case of failure, appropriate error information is provided in the response.

10.3.4 Flow of fetching a VNF package artifact

This clause describes a sequence for fetching an individual artifact contained in an on-boarded VNF package.

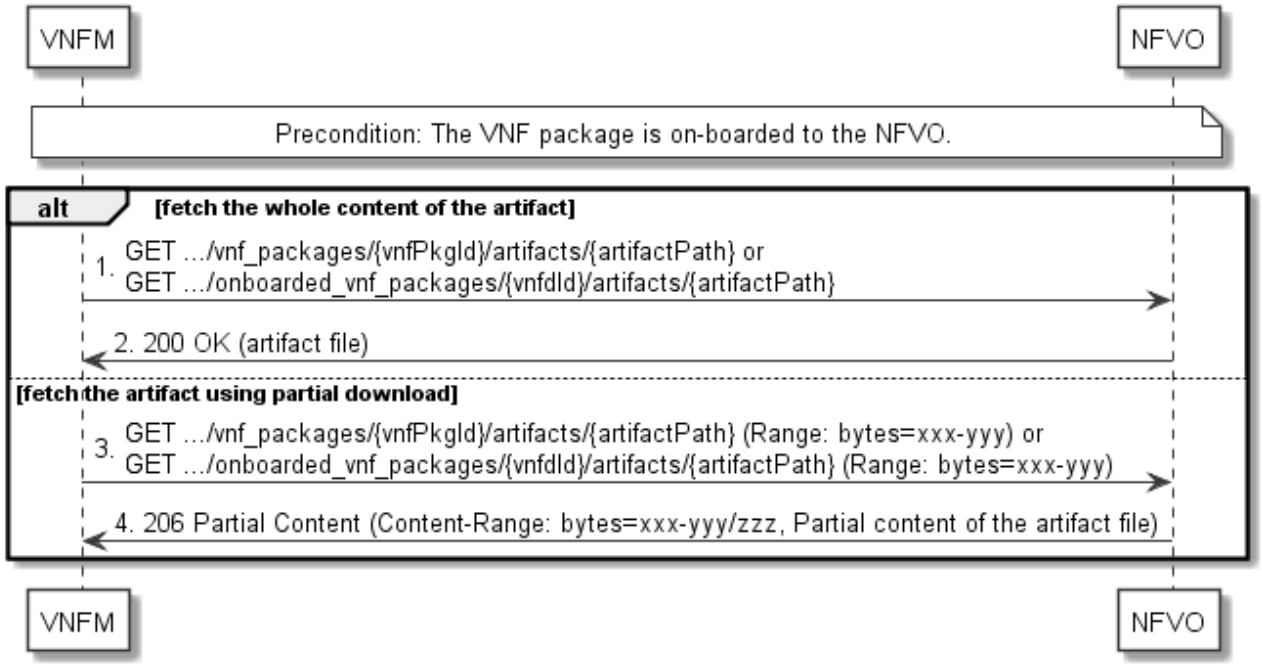

Figure 10.3.4-1: Flow of fetching a VNF package artifact

Precondition: The VNF package is on-boarded to the NFVO.

Fetching an individual artifact contained in an on-boarded VNF package, as illustrated in figure 10.3.4-1, consists of the following steps:

1) If fetching the whole content of the artifact, the VNFM sends a GET request to the "Individual VNF package artifact" resource.

2) The NFVO returns a "200 OK" response and includes a copy of the applicable artifact file from the VNF package in the payload body.

3) If fetching the artifact using partial download, the VNFM sends a GET request to the "Individual VNF package artifact" resource and includes a "Range" HTTP header indicating the partition of the artifact needs to be transferred.

4) The NFVO returns a "206 Partial Content" response with a payload body containing the partial content of the artifact file, and a "Content-Range" HTTP header indicating the byte range enclosed in the payload and the complete length of the artifact file.

Error handling: In case of failure, appropriate error information is provided in the response.

10.3.4a Flow of bulk-fetching VNF package artifacts that are not images

This clause describes a sequence for bulk-fetching artifacts that are not images from an on-boarded VNF package.

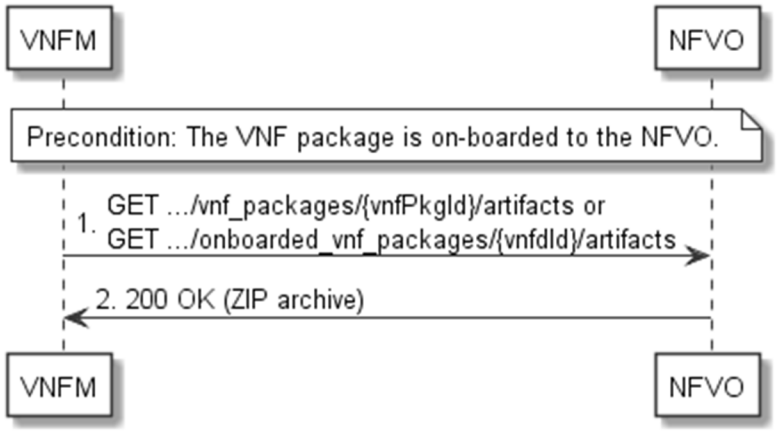

Figure 10.3.4a-1: Flow of bulk-fetching VNF package artifacts

Precondition: The VNF package is on-boarded to the NFVO.

Bulk-fetching of artifacts from an on-boarded VNF package allows the VNFM to request all artifacts that are not images, and further may allow to specify additional filtering criteria for the artifacts to be included in that set.

NOTE: It is assumed that, due to their size, images are excluded from the bulk fetch operation. These can be fetched as individual artifacts.

The procedure consists of the following steps, as illustrated in figure 10.3.4a-1:

1) The VNFM sends a GET request to the "VNF package artifacts" resource and specifies, if supported, the appropriate URI query parameters to define the requested set of artifacts.

2) The NFVO returns a "200 OK" response with a payload body that contains a ZIP archive which contains the applicable artifacts, embedded in the appropriate directory structure in the VNF package.

Error handling: In case of failure, appropriate error information is provided in the response.

10.3.5 Flow of managing subscriptions

This clause describes the procedure for creating, reading and terminating subscriptions to notifications related to VNF package management.

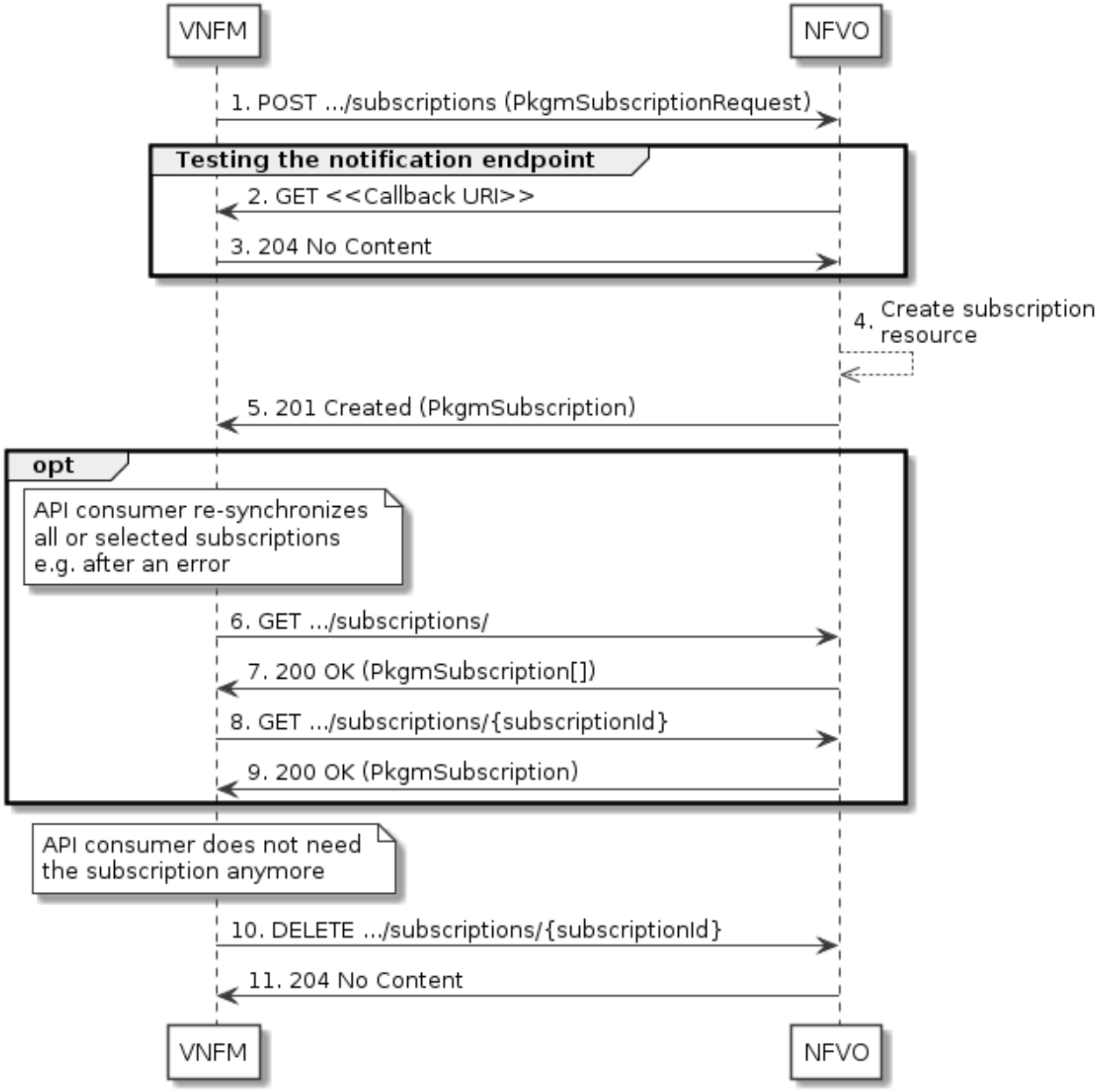

Figure 10.3.5-1: Flow of managing subscriptions

The procedure consists of the following steps as illustrated in figure 10.3.5-1:

1) The VNFM sends a POST request to the "Subscriptions" resource including in the payload body a data structure of type "PkgmSubscriptionRequest". That data structure contains filtering criteria and a callback URI to which the NFVO will subsequently send notifications about events that match the filter.

2) To test the notification endpoint that has been registered by the VNFM as part of the subscription, the NFVO sends a GET request to the notification endpoint URI.

3) The VNFM returns a "204 No Content" response to indicate success.

4) The NFVO creates a new subscription to notifications related to VNF package on-boarding or changes, and a resource that represents this subscription.

5) The NFVO returns a "201 Created" response containing a data structure of type "PkgmSubscription" representing the "Individual subscription" resource just created by the NFVO and provides the URI of the newly-created resource in the "Location" HTTP header.

6) If desired, e.g. to recover from an error situation, the VNFM can obtain information about its subscriptions by sending a GET request to the resource representing the subscriptions.

7) In that case, the NFVO returns a "200 OK" response that contains the list of representations of all existing subscriptions that were created by the VNFM.

8) If desired, e.g. to recover from an error situation, the VNFM can obtain information about a particular subscription by sending a GET request to the resource representing that individual subscription.

9) In that case, the NFVO returns a "200 OK" response that contains a representation of that individual subscription.

10) If the VNFM does not need the subscription anymore, it terminates the subscription by sending a DELETE request to the resource that represents the individual subscription to remove.

11) The NFVO acknowledges the successful termination of the subscription by returning a "204 No Content" response.

Error handling: The NFVO rejects a subscription if the subscription information is not valid: endpoint cannot be reached, subscription information is malformed, etc.

10.3.6 Flow of sending notifications

This clause describes the procedure for sending notifications related to VNF package management.

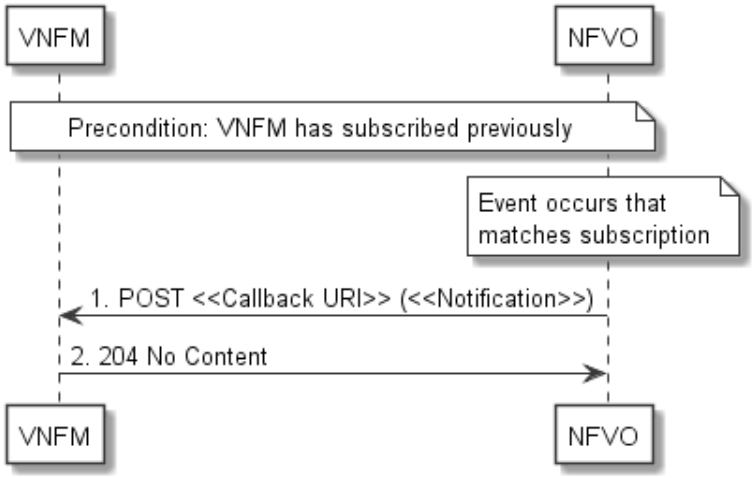

Figure 10.3.6-1: Flow of sending notifications

Precondition: The VNFM has subscribed previously for notifications related to VNF package management.

The procedure consists of the following steps as illustrated in figure 10.3.6-1:

1) If an event occurs that matches the filtering criteria which are part of the subscription, the NFVO generates a notification that includes information about the event and sends it in the body of a POST request to the URI which the VNFM has registered as part of the subscription request. The variable <<Notification>> in the flow is a placeholder for the different types of notifications that can be sent by this API (see clauses 10.5.2.5 and 10.5.2.6).

2) The VNFM acknowledges the successful delivery of the notification by returning a "204 No Content" response.

Error handling: If the NFVO does not receive the "204 No Content" response from the VNFM, it can retry sending the notification.

10.4 Resources

10.4.1 Introduction

This clause defines all the resources and methods provided by the VNF package management interface.

10.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [8] are part of the VNF package management interface.

10.4.2 Resource: VNF packages

10.4.2.1 Description

This resource represents VNF packages. The API consumer can use this resource to query information of the VNF packages.

10.4.2.2 Resource definition

The possible resource URIs are:

{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf_packages

{apiRoot}/vnfpkgm/{apiMajorVersion}/onboarded_vnf_packages

This resource shall support the resource URI variables defined in table 10.4.2.2-1.

Table 10.4.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 10.1a.

10.4.2.3 Resource methods

10.4.2.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.2.3.2 GET

The GET method queries the information of the VNF packages matching the filter.

This method shall follow the provisions specified in tables 10.4.2.3.2-1 and 10.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 10.4.2.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [8]. The NFVO shall support receiving this parameter as part of the URI query string. The VNFM may supply this parameter. All attribute names that appear in the VnfPkgInfo and in data types referenced from it shall be supported by the NFVO in the filter expression.
all_fields	0..1	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The NFVO shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The NFVO should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The NFVO should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The NFVO shall support this parameter. The following attributes shall be excluded from the VnfPkgInfo structure in the response body if this parameter is provided, or none of the parameters "all_fields," "fields", "exclude_fields", "exclude_default" are provided: - softwareImages - additionalArtifacts - userDefinedData - checksum - onboardingFailureDetails.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource.

Table 10.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfPkgInfo	0..N	200 OK	Shall be returned when information about zero or more VNF packages has been queried successfully. The response body shall contain in an array the VNF package info representations that match the attribute filter, i.e. zero or more VNF package info representations as defined in clause 10.5.2.2. If the "filter" URI parameter or one of the "all_fields", "fields" (if supported), "exclude_fields" (if supported) or "exclude_default" URI parameters was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clauses 5.2.2 and 5.3.2 of ETSI GS NFV-SOL 013 [8], respectively. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [8].

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute selector. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

10.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.3 Resource: Individual VNF package

10.4.3.1 Description

This resource represents an individual VNF package. The API consumer can use this resource to read information of the VNF package.

10.4.3.2 Resource definition

The possible resource URIs are:

{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf_packages/{vnfPkgId}

{apiRoot}/vnfpkgm/{apiMajorVersion}/onboarded_vnf_packages/{vnfdId}

This resource shall support the resource URI variables defined in table 10.4.3.2-1.

Table 10.4.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 10.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note 1.
vnfdId	Identifier of the VNFD and the VNF package. The identifier is allocated by the VNF provider. See note 2.
NOTE 1: This identifier can be retrieved from the "vnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification. NOTE 2: This identifier can be retrieved from the "vnfdId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.	

10.4.3.3 Resource methods

10.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.3.3.2 GET

The GET method reads the information of an individual VNF package.

This method shall follow the provisions specified in tables 10.4.3.3.2-1 and 10.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 10.4.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 10.4.3.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfPkgInfo	1	200 OK	Shall be returned when information of the VNF package has been read successfully. The response body shall contain the VNF package info representation defined in clause 10.5.2.2.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

10.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.3.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.3.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.4 Resource: VNFD in an individual VNF package

10.4.4.1 Description

This resource represents the VNFD contained in an on-boarded VNF package. The API consumer can use this resource to obtain the content of the VNFD.

10.4.4.2 Resource definition

The possible resource URIs are:

{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf_packages/{vnfPkgId}/vnfd

{apiRoot}/vnfpkgm/{apiMajorVersion}/onboarded_vnf_packages/{vnfdId}/vnfd

This resource shall support the resource URI variables defined in table 10.4.4.2-1.

Table 10.4.4.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 10.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note 1.
vnfdId	Identifier of the VNFD and the VNF package. The identifier is allocated by the VNF provider. See note 2.
NOTE 1: This identifier can be retrieved from the "vnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification. NOTE 2: This identifier can be retrieved from the "vnfdId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.	

10.4.4.3 Resource methods

10.4.4.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.4.3.2 GET

The GET method reads the content of the VNFD within a VNF package.

The VNFD is implemented as a collection of one or more files. A ZIP archive embedding these files shall be returned when reading this resource.

The default format of the ZIP archive shall comply with the CSAR format as specified in ETSI GS NFV-SOL 004 [2] where only the files representing the VNFD and information needed to navigate the ZIP archive and to identify the file that is the entry point for parsing the VNFD, and, if requested, further security information are included, and software images as well as other artifacts referenced from the YAML files are excluded. This means that the structure of the ZIP archive shall correspond to the directory structure used in the VNF package and that the archive shall contain the following files from the package:

- TOSCA.meta (if available in the package).
- The main TOSCA definitions YAML file (either as referenced by the Entry-Definitions keyword from TOSCA.meta or available as a file with the extension ".yml" or ".yaml" from the root of the archive).

- Other TOSCA YAML files, if any, as referenced by the Other-Definitions keyword from TOSCA.meta.
- Every component of the VNFD referenced (recursively) from the YAML files as mentioned above.

NOTE 1: For a VNFD based on TOSCA, it includes all the imported type definition files as indicated in the top level service template and in any of the lower level service template if it has any as described in ETSI GS NFV-SOL 001 [i.4].

NOTE 2: For a VNFD based on YANG, it includes the file as indicated by the "yang_definitions" keyname in the metadata section of the main yaml file as described in ETSI GS NFV-SOL 004 [2].

- The related security information, if the "include_signatures" URI parameter is provided, as follows:
  - the manifest file;
  - the singleton certificate file in the root of the VNF package (if available in the package);
  - the signing certificates of the individual files included in the ZIP archive (if available in the package);
  - the signatures of the individual files (if available in the package).

Three examples are provided below.

NOTE 3: These examples do not show the security related files.

EXAMPLE 1: Assuming a request is sent for the following VNF package (as described in clause A.1 in ETSI GS NFV-SOL 004 [2]):

```
!------TOSCA-Metadata
        !-----TOSCA.meta (metadata for navigating the ZIP file)
!------Definitions
        !-----MRF.yaml (main VNFD file)
        !-----OtherTemplates (e.g. type definitions referenced by the main VNFD
file or any files indicated in the Other-Definitions keyword in the TOSCA.meta
file if it has any)
!------Files
        !-----ChangeLog.txt
        !-----image(s)
        !-----other artifacts
!------Tests
        !-----file(s)
!------Licenses
        !-----file(s)
!------Scripts
        !-----install.sh
!----- MRF.mf
```

The NFVO will return a ZIP file of the following format:

```
!------TOSCA-Metadata
        !----- TOSCA.meta
!------Definitions
        !----- MRF.yaml
        !----- OtherTemplates
```

EXAMPLE 2: Assuming a request is sent for the following VNF package (a VNF package without a TOSCA-Metadata directory, as described in clause A.2 in ETSI GS NFV-SOL 004 [2]):

```
!------MRF.yaml (main VNFD file)
!------MRF.mf
!------ChangeLog.txt
!------Tests
        !----- file(s)
!------Licenses
        !----- file(s)
!------Artifacts
        !----- install.sh
        !----- start.yang
```

The NFVO will return a ZIP file of the following format:

```
!------MRF.yaml
```

EXAMPLE 3: Assuming a request is sent for the following VNF package (a VNF package with the YANG VNFD without a TOSCA-Metadata directory, as described in clause A.3 in ETSI GS NFV-SOL 004 [2]):

```
!----CompanyVNFD.yaml
!----CompanyVNFD.xml
!----CompanyVNFD.mf
!----ChangeLog.txt
!-----Files
!-----Instance Data Files
        !---- start.xml
!-----Licenses
!-----Scripts
        !----- install.sh
```

The NFVO will return a ZIP file of the following format:

```
!----CompanyVNFD.yaml
!----CompanyVNFD.xml  (indicated in the yang_definitions metadata in
CompanyVNFD.yaml)
```

This method shall follow the provisions specified in tables 10.4.4.3.2-1 and 10.4.4.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 10.4.4.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
include_signatures	0..1	If this parameter is provided, the NFVO shall include in the ZIP archive the security information as specified above. This URI query parameter is a flag, i.e. it shall have no value. The NFVO shall support this parameter.

Table 10.4.4.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a	1	200 OK	Shall be returned when the content of the VNFD has been read successfully. The payload body shall contain a ZIP archive that contains the files representing the VNFD, as specified above. The "Content-Type" HTTP header shall be set to "application/zip".
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that "onboardingState" of the VNF package has a value different from "ONBOARDED". The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

10.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.4a Resource: Manifest in an individual VNF package

10.4.4a.1 Description

This resource represents the manifest contained in an on-boarded VNF package. The API consumer can use this resource to obtain the content of the manifest.

10.4.4a.2 Resource definition

The possible resource URIs are:

{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf_packages/{vnfPkgId}/manifest

{apiRoot}/vnfpkgm/{apiMajorVersion}/onboarded_vnf_packages/{vnfdId}/manifest

This resource shall support the resource URI variables defined in table 10.4.4a.2-1.

Table 10.4.4a.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 10.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note 1.
vnfdId	Identifier of the VNFD and the VNF package. The identifier is allocated by the VNF provider. See note 2.
NOTE 1: This identifier can be retrieved from the "vnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification. NOTE 2: This identifier can be retrieved from the "vnfdId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.	

10.4.4a.3 Resource methods

10.4.4a.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.4a.3.2 GET

The GET method reads the content of the manifest within a VNF package.

This method shall follow the provisions specified in tables 10.4.4a.3.2-1 and 10.4.4a.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 10.4.4a.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
include_signatures	0..1	If this parameter is provided, the NFVO shall return the manifest and related security information (such as certificate) in a ZIP archive. If this parameter is not given, the NFVO shall provide only a copy of the manifest file. This URI query parameter is a flag, i.e. it shall have no value. The NFVO shall support this parameter.

Table 10.4.4a.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a	1	200 OK	Shall be returned when the content of the manifest has been read successfully. If the "include_signatures" URI query parameter was absent in the request, or if the manifest file has all security-related information embedded (i.e. there is no separate certificate file), the payload body shall contain a copy of the manifest file of the VNF package and the "Content-Type" HTTP header shall be set to "text/plain". If the "include_signatures" URI query parameter was present in the related request and the manifest file does not have all the security-related information embedded (i.e. there is a separate certificate file), the "Content-Type" HTTP header shall be set to "application/zip and the payload body shall contain a ZIP archive which includes: • a copy of the manifest file of the VNF package; • a copy of the related individual certificate file.
	ProblemDetails	0..1	406 Not Acceptable	If the related request contained an "Accept" header not compatible with the Content type "application/zip" but the "include_signatures" flag was provided, the NFVO shall respond with this response code. The "ProblemDetails" structure may be included with the "detail" attribute providing more information about the error.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that "onboardingState" of the VNF package has a value different from "ONBOARDED". The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

10.4.4a.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.4a.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.4a.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.5 Resource: VNF package content

10.4.5.1 Description

This resource represents the content of a VNF package identified by the VNF package identifier allocated by the NFVO. The API consumer can use this resource to fetch the content of the VNF package.

10.4.5.2 Resource definition

The possible resource URIs are:

{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf_packages/{vnfPkgId}/package_content

{apiRoot}/vnfpkgm/{apiMajorVersion}/onboarded_vnf_packages/{vnfdId}/package_content

This resource shall support the resource URI variables defined in table 10.4.5.2-1.

Table 10.4.5.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 10.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note 1.
vnfdId	Identifier of the VNFD and the VNF package. The identifier is allocated by the VNF provider. See note 2.
NOTE 1: This identifier can be retrieved from the "vnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification. NOTE 2: This identifier can be retrieved from the "vnfdId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.	

10.4.5.3 Resource methods

10.4.5.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.5.3.2 GET

The GET method fetches the content of a VNF package identified by the VNF package identifier allocated by the NFVO.

The content of the package is provided as onboarded, i.e. depending on the security option used, the CSAR or the CSAR wrapped in a ZIP archive together with an external signature is returned, as defined in clause 5.1 of ETSI GS NFV-SOL 004 [2].

NOTE: Information about the applicable security option can be obtained by evaluating the "packageSecurityOption" attribute in the "VnfPkgInfo" structure.

This method shall follow the provisions specified in tables 10.4.5.3.2-1 and 10.4.5.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 10.4.5.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 10.4.5.3.2-2: Details of the GET request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	n/a		The request may contain a "Range" HTTP header to obtain single range of bytes from the VNF package file. This can be used to continue an aborted transmission. If the "Range" header is present in the request and the NFVO does not support responding to range requests with a 206 response, it shall return a 200 OK response instead as defined below.	
Response body	n/a	1	200 OK	Shall be returned when the whole content of the VNF package file has been read successfully. The response body shall include a copy of the VNF package file. The "Content-Type HTTP" header shall be set according to the type of the file, i.e. to "application/zip" for a VNF Package as defined in ETSI GS NFV-SOL 004 [2].
	n/a	1	206 Partial Content	If the NFVO supports range requests, this response shall be returned when a single consecutive byte range from the content of the VNF package file has been read successfully according to the request. The response body shall contain the requested part of the VNF package file. The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [4]. The "Content-Type" HTTP header shall be set as defined above for the "200 OK" response.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that "onboardingState" of the VNF package has a value different from "ONBOARDED". The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	416 Range Not Satisfiable	Shall be returned upon the following error: The byte range passed in the "Range" header did not match any available byte range in the VNF package file (e.g. "access after end of file"). The response body may contain a ProblemDetails structure.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

10.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.5a Resource: VNF package artifacts

10.4.5a.1 Description

This resource represents the artifacts contained in a VNF package. The API consumer can use this resource to bulk-fetch the artifacts.

Optional filters allow to restrict the set of artifacts included in the resource representation. In the present version of the present document, image artifacts are excluded from the representation of this resource.

10.4.5a.2 Resource definition

The possible resource URIs are:

{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf_packages/{vnfPkgId}/artifacts

{apiRoot}/vnfpkgm/{apiMajorVersion}/onboarded_vnf_packages/{vnfdId}/artifacts

This resource shall support the resource URI variables defined in table 10.4.5a.2-1.

Table 10.4.5a.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 10.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note 1.
vnfdId	Identifier of the VNFD and the VNF package. The identifier is allocated by the VNF provider. See note 2.
NOTE 1: This identifier can be retrieved from the "vnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification. NOTE 2: This identifier can be retrieved from the "vnfdId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.	

10.4.5a.3 Resource methods

10.4.5a.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.5a.3.2 GET

The GET method shall return an archive that contains a set of artifacts according to the provisions for inclusion/exclusion defined below, embedded in a directory structure being the same as in the VNF package.

The criteria for exclusion/inclusion of an artifact in the archive are defined as follows:

- Artifacts that are software images shall be excluded from the archive.
- Artifacts that are not software images and that are external to the VNF package shall be excluded from the archive unless the URI query parameter "include_external_artifacts" has been provided. External artifacts shall be included in the archive using the content of the "artifactPath" attribute as the path.
- All additional artifacts included in the VNF package that are MANO artifacts shall be included in the archive, unless the URI query parameter "exclude_all_mano_artifacts" has been provided, in which case such artifacts shall be excluded.
- All additional artifacts included in the VNF package that are non-MANO artifacts shall be included in the archive, unless:
  - The URI query parameter "exclude_all_non_mano_artifacts" has been provided, in which case such artifacts shall be excluded.
  - The URI query parameter "select_non_mano_artifact_sets" has been provided and is supported by the NFVO, in which case only those non-MANO artifacts shall be included whose non-MANO artifact set identifier matches one of the values of the query parameter.

Package metadata such as manifest file or VNFD shall not be included in the archive.

This method shall follow the provisions specified in tables 10.4.5a.3.2-1 and 10.4.5a.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 10.4.5a.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
exclude_all_mano_artifacts	0..1	Flag (i.e. parameter without value) that instructs the NFVO to exclude the set of additional MANO artifacts (i.e. those that are not images) from the response payload body. The NFVO shall support this parameter. The VNFM may supply this parameter.
exclude_all_non_mano_artifacts	0..1	Flag (i.e. parameter without value) that instructs the NFVO to exclude the set of non-MANO artifacts from the response payload body. The NFVO shall support this parameter. The VNFM may supply this parameter.
include_external_artifacts	0..1	Flag (i.e. parameter without value) that instructs the NFVO to include external artifacts in the response payload body. It shall not be treated as an error if this flag is provided but there is no external artifact to include in the result. If this parameter is missing, no external artifacts shall be included. The NFVO shall support this parameter. The VNFM may supply this parameter.
select_non_mano_artifact_sets	0..1	Comma-separated list of non-MANO artifact set identifiers for which the artifacts are to be included in the response body. The NFVO should support this parameter. If the NFVO does not support this parameter, it shall ignore it, i.e. provide a response as if no parameter was provided. The VNFM may supply this parameter.
include_signatures	0..1	If this parameter is provided, the NFVO shall include in the ZIP archive the individual signatures and, if provided, related certificates for the included artifacts, in the format in which they are provided in the VNF package. If this parameter is not given, the NFVO shall only provide copies of the artifact files. This URI query parameter is a flag, i.e. it shall have no value. The NFVO shall support this parameter.

Table 10.4.5a.3.2-2: Details of the GET request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	n/a			The "Accept" HTTP header shall be set to "application/zip". The request may contain a "Range" HTTP header to obtain single range of bytes from the archive containing the artifacts. This can be used to continue an aborted transmission. If the "Range" header is present in the request and the NFVO does not support responding to range requests with a 206 response, it shall return a 200 OK response instead as defined below.
Response body	n/a	1	200 OK	Shall be returned when the whole content of the archive containing the artifact files has been read successfully. The payload body shall be a ZIP archive containing the requested set of artifacts selected according to the provisions specified above in this clause, and, if the flag "include_signatures" was provided in the related request, the applicable signature files and, if available, the separate certificate files from the VNF package. The "Content-Type" HTTP header shall be set to "application/zip".
	n/a	1	206 Partial Content	If the NFVO supports range requests, this response shall be returned when a single consecutive byte range from the content of the archive that would have been returned in a "200 OK" response has been read successfully according to the request. The response body shall contain the requested part of the archive. The "Content-Type" HTTP header shall be set to "application/zip". The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [4].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid URI query parameters, including the following situations: • "exclude_all_non_mano_artifacts" and "select_non_mano_artifact_sets" are both present in the request • "exclude_all_non_mano_artifacts" and "exclude_all_mano_artifacts" are both present in the request • one or more of the values provided in "select_non_mano_artifact_sets" are not defined in the manifest of the VNF package. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that "onboardingState" of the VNF package has a value different from "ONBOARDED". The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	416 Range Not Satisfiable	Shall be returned upon the following error: The byte range passed in the "Range" header did not match any available byte range in the archive file (e.g. "access after end of file"). The response body may contain a ProblemDetails structure.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

10.4.5a.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.5a.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.5a.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.6 Resource: Individual VNF package artifact

10.4.6.1 Description

This resource represents an individual artifact contained in a VNF package. The API consumer can use this resource to fetch the content of the artifact.

10.4.6.2 Resource definition

The possible resource URIs are:

{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf_packages/{vnfPkgId}/artifacts/{artifactPath}

{apiRoot}/vnfpkgm/{apiMajorVersion}/onboarded_vnf_packages/{vnfdId}/artifacts/{artifactPath}

This resource shall support the resource URI variables defined in table 10.4.6.2-1.

Table 10.4.6.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 10.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note 1.
vnfdId	Identifier of the VNFD and the VNF package. The identifier is allocated by the VNF provider. See note 3.
artifactPath	For an artifact contained as a file in the VNF package, this variable shall contain a sequence of one or more path segments representing the path of the artifact within the VNF package, relative to the root of the package. See note 4. EXAMPLE: foo/bar/m%40ster.sh For an external artifact represented as a URI in the VNF package manifest, this variable shall contain a sequence of one or more path segments as synthesized by the NFVO (see clause 10.5.3.3), representing this artifact. See note 2 and note 4.
NOTE 1: This identifier can be retrieved from the "vnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.	
NOTE 2: This identifier can be retrieved from the "artifactPath" attribute of the applicable "additionalArtifacts" entry in the body of the response to a GET request querying the "Individual VNF package" or the "VNF packages" resource.	
NOTE 3: This identifier can be retrieved from the "vnfdId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.	
NOTE 4: Since multiple path segments are allowed to be contained in this variable, the "/" character that separates these segments is not percent-encoded. Each individual segment is percent-encoded if necessary as defined in clause 4.1 of ETSI GS NFV-SOL 013 [8].	

10.4.6.3 Resource methods

10.4.6.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.6.3.2 GET

The GET method fetches the content of an artifact within a VNF package.

This method shall follow the provisions specified in tables 10.4.6.3.2-1 and 10.4.6.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 10.4.6.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
include_signatures	0..1	If this parameter is provided, the NFVO shall return the artifact and related security information (such as signature and optional certificate) in a ZIP archive. If this parameter is not given, the NFVO shall provide only a copy of the artifact file. This URI query parameter is a flag, i.e. it shall have no value. The NFVO shall support this parameter.

Table 10.4.6.3.2-2: Details of the GET request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	n/a			The request may contain a "Range" HTTP header to obtain single range of bytes from an artifact file. This can be used to continue an aborted transmission. If the "Range" header is present in the request and the NFVO does not support responding to range requests with a 206 response, it shall return a 200 OK response instead as defined below.
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a	1	200 OK	Shall be returned when the whole content of the artifact file has been read successfully. If the "include_signatures" request URI parameter was not provided in the related request, the payload body shall contain a copy of the artifact file from the VNF package, as defined by ETSI GS NFV-SOL 004 [2] and the "Content-Type" HTTP header shall be set according to the content type of the artifact file. If the artifact is encrypted, the header shall be set to the value "application/cms" (IETF RFC 7193 [9]). If the content type cannot be determined, the header shall be set to the value "application/octet-stream". If the "include_signatures" request URI parameter was provided in the related request, the "Content-Type" HTTP header shall be set to "application/zip and the payload body shall contain a ZIP archive which includes: • a copy of the artifact file from the VNF package, as defined by ETSI GS NFV-SOL 004 [2]; • the related security information (individual signature file and optional related individual certificate file).
	n/a	1	206 Partial Content	If the NFVO supports range requests and the "include_signatures" request URI parameter was not present in the related request, this response shall be returned when a single consecutive byte range from the content of the artifact file has been read successfully according to the request. The response body shall contain the requested part of the artifact file from the VNF package, as defined by ETSI GS NFV-SOL 004 [2]. The "Content-Type" HTTP header shall be set according to the content type of the artifact file. If the content type cannot be determined, the header shall be set to the value "application/octet-stream". The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [4].
	ProblemDetails	0..1	406 Not Acceptable	If the related request contained an "Accept" header not compatible with the Content type "application/zip" but the "include_signatures" flag was provided, the NFVO shall respond with this response code. The "ProblemDetails" structure may be included with the "detail" attribute providing more information about the error.

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that "onboardingState" of the VNF package has a value different from "ONBOARDED". The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	416 Range Not Satisfiable	Shall be returned upon the following error: The byte range passed in the "Range" header did not match any available byte range in the artifact file (e.g. "access after end of file"). The response body may contain a ProblemDetails structure.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

10.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.6.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.6.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.7 Resource: Subscriptions

10.4.7.1 Description

This resource represents subscriptions. The API consumer can use this resource to subscribe to notifications related to the VNF package management, and to query its subscriptions.

10.4.7.2 Resource definition

The resource URI is:

{apiRoot}/vnfpkgm/{apiMajorVersion}/subscriptions

This resource shall support the resource URI variables defined in table 10.4.7.2-1.

Table 10.4.7.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 10.1a.

10.4.7.3 Resource methods

10.4.7.3.1 POST

The POST method creates a new subscription.

This method shall follow the provisions specified in tables 10.4.7.3.1-1 and 10.4.7.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual subscription" resource as defined in clause 10.4.8 shall have been created. This method shall not trigger any notification.

Creation of two "Individual subscription" resources with the same callback URI and the same filter can result in performance degradation and will provide duplicates of notifications to the VNFM, and might make sense only in very rare use cases. Consequently, the NFVO may either allow creating a new "Individual subscription" resource if another "Individual subscription" resource with the same filter and callback URI already exists (in which case it shall return the "201 Created" response code), or may decide to not create a duplicate "Individual subscription" resource (in which case it shall return a "303 See Other" response code referencing the existing "Individual subscription" resource with the same filter and callback URI).

Table 10.4.7.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 10.4.7.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	PkgmSubscriptionRequest	1	Details of the subscription to be created.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	PkgmSubscription	1	201 Created	Shall be returned when the subscription has been created successfully. The response body shall contain a representation of the created "Individual subscription" resource. The HTTP response shall include a "Location" HTTP header that points to the created resource.
	n/a		303 See Other	Shall be returned when a subscription with the same callback URI and the same filter already exists and the policy of the NFVO is to not create redundant subscriptions. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the existing "Individual subscription" resource. The response body shall be empty.

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	1	422 Unprocessable Entity	Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 10.4.9.3.2 and the test has failed. In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

10.4.7.3.2 GET

The GET method queries the list of active subscriptions of the functional block that invokes the method. It can be used e.g. for resynchronization after error situations.

This method shall follow the provisions specified in tables 10.4.7.3.2-1 and 10.4.7.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 10.4.7.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [8]. The NFVO shall support receiving this parameter as part of the URI query string. The VNFM may supply this parameter. All attribute names that appear in the PkgmSubscription and in data types referenced from it shall be supported by the NFVO in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource.

Table 10.4.7.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	PkgmSubscription	0..N	200 OK	Shall be returned when the list of subscriptions has been queried successfully. The response body shall contain in an array the representations of all active subscriptions of the functional block that invokes the method, i.e. zero or more representations of VNF package management subscriptions as defined in clause 10.5.2.4. If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [8]. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

10.4.7.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.7.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.7.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.8 Resource: Individual subscription

10.4.8.1 Description

This resource represents an individual subscription. The API consumer can use this resource to read and to terminate a subscription to notifications related to the VNF package management.

10.4.8.2 Resource definition

The resource URI is:

{apiRoot}/vnfpkgm/{apiMajorVersion}/subscriptions/{subscriptionId}

This resource shall support the resource URI variables defined in table 10.4.8.2-1.

Table 10.4.8.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 10.1a.
subscriptionId	Identifier of this subscription. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual subscription" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

10.4.8.3 Resource methods

10.4.8.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.8.3.2 GET

The GET method reads an individual subscription.

This method shall follow the provisions specified in tables 10.4.8.3.2-1 and 10.4.8.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 10.4.8.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 10.4.8.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	PkgmSubscription	1	200 OK	Shall be returned when information about an individual subscription has been read successfully. The response body shall contain a representation of the "Individual subscription" resource.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

10.4.8.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.8.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.8.3.5 DELETE

The DELETE method terminates an individual subscription.

This method shall follow the provisions specified in tables 10.4.8.3.5-1 and 10.4.8.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual subscription" resource shall not exist any longer. This means that no notifications for that subscription shall be sent to the formerly-subscribed API consumer.

NOTE: Due to race conditions, some notifications might still be received by the formerly-subscribed API consumer for a certain time period after the deletion.

Table 10.4.8.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 10.4.8.3.5-2: Details of the DELETE request/response on this resource

Request body	**Data type**	**Cardinality**	**Description**	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned when the "Individual subscription" resource has been deleted successfully.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

10.4.9 Resource: Notification endpoint

10.4.9.1 Description

This resource represents a notification endpoint.

The API producer can use this resource to send notifications related to VNF package management events to a subscribed API consumer, which has provided the URI of this resource during the subscription process.

10.4.9.2 Resource definition

The resource URI is provided by the API consumer when creating the subscription.

This resource shall support the resource URI variables defined in table 10.4.9.2-1.

Table 10.4.9.2-1: Resource URI variables for this resource

Name	Definition
none supported	

10.4.9.3 Resource methods

10.4.9.3.1 POST

The POST method delivers a notification from the API producer to an API consumer. The API consumer shall have previously created an "Individual subscription" resource with a matching filter.

This method shall follow the provisions specified in tables 10.4.9.3.1-1 and 10.4.9.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 10.4.9.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Each notification request body shall include exactly one of the alternatives defined in table 10.4.9.3.1-2.

Table 10.4.9.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	VnfPackageOnboardingNotification	1	A notification about on-boarding of a VNF package.	
	VnfPackageChangeNotification	1	A notification about changes of status in a VNF package.	
Response body	n/a		204 No Content	Shall be returned when the notification has been delivered successfully.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

10.4.9.3.2 GET

The GET method allows the API producer to test the notification endpoint that is provided by the API consumer, e.g. during subscription.

This method shall follow the provisions specified in tables 10.4.9.3.2-1 and 10.4.9.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 10.4.9.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 10.4.9.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned to indicate that the notification endpoint has been tested successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

10.4.9.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.9.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.4.9.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

10.5 Data model

10.5.1 Introduction

This clause defines the request and response data structures of the VNF package management interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error and may choose to ignore them.

10.5.2 Resource and notification data types

10.5.2.1 Introduction

This clause defines data structures to be used in resource representations and notifications.

10.5.2.2 Type: VnfPkgInfo

This type represents the information of a VNF package. It shall comply with the provisions defined in table 10.5.2.2-1.

Table 10.5.2.2-1: Definition of the VnfPkgInfo data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the VNF package. This identifier is allocated by the NFVO.
vnfdId	Identifier	0..1	This identifier, which is managed by the VNF provider, identifies the VNF package and the VNFD in a globally unique way. It is copied from the VNFD of the on-boarded VNF package. It shall be present after the VNF package content has been on-boarded and absent otherwise.
vnfdExtInvariantId	Identifier	0..1	Identifies a VNFD in a version independent manner. This attribute is invariant across versions of the VNFD that fulfil certain conditions related to the external connectivity and management of the VNF. It shall be present after the VNF package content has been on-boarded if it is included in the VNFD and shall be absent otherwise. If present it is copied from the VNFD of the on-boarded VNF package.
vnfProvider	String	0..1	Provider of the VNF package and the VNFD. This information is copied from the VNFD. It shall be present after the VNF package content has been on-boarded and absent otherwise.

Attribute name	Data type	Cardinality	Description
vnfProductName	String	0..1	Name to identify the VNF product. Invariant for the VNF product lifetime. This information is copied from the VNFD. It shall be present after the VNF package content has been on-boarded and absent otherwise.
vnfSoftwareVersion	Version	0..1	Software version of the VNF. This is changed when there is any change to the software included in the VNF package. This information is copied from the VNFD. It shall be present after the VNF package content has been on-boarded and absent otherwise.
vnfdVersion	Version	0..1	The version of the VNFD. This information is copied from the VNFD. It shall be present after the VNF package content has been on-boarded and absent otherwise.
compatibleSpecificationVersions	Version	0..N	Indicates which versions of the ETSI GS NFV-SOL 004 [2] specification the package complies to, as defined in the manifest of the package. Each entry shall be formatted as defined in clause 4.3.2 of ETSI GS NFV-SOL 004 [2].
checksum	Checksum	0..1	Checksum of the on-boarded VNF package. It shall be present after the VNF package content has been on-boarded and absent otherwise. Permitted hash algorithms are defined in ETSI GS NFV-SOL 004 [2].
packageSecurityOption	Enum (inlined)	0..1	Signals the security option used by the package as defined in clause 5.1 of ETSI GS NFV-SOL 004 [2]. It shall be present after the VNF package content has been on-boarded and absent otherwise. Valid values: • OPTION_1 • OPTION_2
signingCertificate	String	0..1	The singleton signing certificate if it is included as a file in the VNF package.
softwareImages	VnfPackageSoftwareImageInfo	0..N	Information about VNF package artifacts that are software images. Every local and external software image referenced from the VNFD shall be included. No other artifacts shall be included. This attribute shall not be present before the VNF package content is on-boarded. Otherwise, this attribute shall be present unless it has been requested to be excluded per attribute selector.

Attribute name	Data type	Cardinality	Description
additionalArtifacts	VnfPackageArtifactInfo	0..N	Information about VNF package artifacts contained in the VNF package that are not software images. Every local and external artifact declared in the manifest shall be included, except the software images and the files that make up the parts of the VNFD (see clause 10.4.4.3.2). Signature files and certificate files are not considered as artifacts, however, the content of the "Licenses" and "Testing" directories in the VNF package is. This attribute shall not be present before the VNF package content is on-boarded. Otherwise, this attribute shall be present if the VNF package contains additional artifacts.
onboardingState	PackageOnboardingStateType	1	On-boarding state of the VNF package.
operationalState	PackageOperationalStateType	1	Operational state of the VNF package. See note 1.
usageState	PackageUsageStateType	1	Usage state of the VNF package. See note 2.
vnfmInfo	String	1..N	Specifies VNFMs compatible with the VNF. This information is copied from the VNFD. See note 3.
userDefinedData	KeyValuePairs	0..1	User defined data for the VNF package.
onboardingFailureDetails	ProblemDetails	0..1	Failure details of current onboarding procedure. See clause 6.3 of ETSI GS NFV-SOL 013 [8] for the details of "ProblemDetails" structure. If "onboardingState" is "ERROR", this attribute shall be present and contain error information (such as failed onboarding or processing operation, affected artifact, etc.), unless it has been requested to be excluded via an attribute selector.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>vnfd	Link	1	Link to the "VNFD in an individual VNF package" resource.
>packageContent	Link	1	Link to the "VNF package content" resource.

NOTE 1: If the value of the onboardingState attribute is not equal to "ONBOARDED", the value of the operationalState attribute shall be equal to "DISABLED".
NOTE 2: If the value of the onboardingState attribute is not equal to "ONBOARDED", the value of the usageState attribute shall be equal to "NOT_IN_USE".
NOTE 3: ETSI GS NFV-SOL 001 [i.4] specifies the structure and format of the VNFD based on TOSCA specifications.

10.5.2.3 Type: PkgmSubscriptionRequest

This type represents a subscription request related to VNF package management notifications about VNF package on-boarding or changes. It shall comply with the provisions defined in table 10.5.2.3-1.

Table 10.5.2.3-1: Definition of the PkgmSubscriptionRequest data type

Attribute name	Data type	Cardinality	Description
filter	PkgmNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this subscription, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [8]. This attribute shall only be present if the subscriber requires authorization of notifications.

10.5.2.4 Type: PkgmSubscription

This type represents a subscription related to notifications about VNF package management. It shall comply with the provisions defined in table 10.5.2.4-1.

Table 10.5.2.4-1: Definition of the PkgmSubscription data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this "Individual subscription" resource.
filter	PkgmNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.

10.5.2.5 Type: VnfPackageOnboardingNotification

This type represents a VNF package management notification, which informs the receiver that the onboarding process of a VNF package is complete and the package is ready for use. It shall comply with the provisions defined in table 10.5.2.5-1. The support of this notification is mandatory.

The notification shall be triggered by the NFVO when the "onboardingState" attribute of a new VNF package has changed to "ONBOARDED".

Table 10.5.2.5-1: Definition of the VnfPackageOnboardingNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "VnfPackageOnboardingNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
vnfPkgId	Identifier	1	Identifier of the VNF package. This identifier is allocated by the NFVO. Its value is the same as the value of the "id" attribute of the related "Individual VNF package" resource.

Attribute name	Data type	Cardinality	Description
vnfdId	Identifier	1	This identifier, which is managed by the VNF provider, identifies the VNF package and the VNFD in a globally unique way. It is copied from the VNFD of the on-boarded VNF package.
vnfmInfo	String	1..N	Specifies VNFMs compatible with the VNF. This information is copied from the VNFD. See table 10.5.2.2-1.
_links	PkgmLinks	1	Links to resources related to this notification.

10.5.2.6 Type: VnfPackageChangeNotification

This type represents a VNF package management notification, which informs the receiver of a change of the status in an on-boarded VNF package. Only changes in the "operationalState" attribute of an on-boarded VNF package and the deletion VNF package will be reported. Changes in the "usageState" and "onboardingState" attributes are not reported. The notification shall comply with the provisions defined in table 10.5.2.6-1. The support of this notification is mandatory.

The notification shall be triggered by the NFVO when there is a change in the status of an onboarded VNF package, as follows:

- The "operationalState" attribute of a VNF package has changed, and the "onboardingState" attribute of the package has the value "ONBOARDED" (i.e. the package has been onboarded previously).
- The on-boarded VNF package has been deleted, and the "onboardingState" attribute of the deleted package had the value "ONBOARDED".

Table 10.5.2.6-1: Definition of the VnfPackageChangeNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "VnfPackageChangeNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
vnfPkgId	Identifier	1	Identifier of the VNF package. This identifier is allocated by the NFVO. Its value is the same as the value of the "id" attribute of the related "Individual VNF package" resource.
vnfdId	Identifier	1	Identifier of the VNFD contained in the VNF package, which also identifies the VNF package. This identifier is allocated by the VNF provider and copied from the VNFD.
changeType	PackageChangeType	1	The type of change of the VNF package.
operationalState	PackageOperationalStateType	0..1	New operational state of the VNF package. Only present when changeType is OP_STATE_CHANGE.
_links	PkgmLinks	1	Links to resources related to this notification.

10.5.3 Referenced structured data types

10.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but are not resource representations.

10.5.3.2 Type: VnfPackageSoftwareImageInfo

This type represents an artifact contained in or external to a VNF package which represents a software image. It shall comply with the provisions defined in table 10.5.3.2-1.

Table 10.5.3.2-1: Definition of the VnfPackageSoftwareImageInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnfd	1	Identifier of the software image.
name	String	1	Name of the software image.
provider	String	1	Provider of the software image.
version	Version	1	Version of the software image.
checksum	Checksum	1	Checksum of the software image file. Permitted hash algorithms are defined in ETSI GS NFV-SOL 004 [2].
isEncrypted	Boolean	1	Reflects whether the image is encrypted (true) or not (false).
containerFormat	Enum (inlined)	1	Container format indicates whether the software image is in a file format that also contains metadata about the actual software. Permitted values: - AKI: a kernel image format - AMI: a machine image format - ARI: a ramdisk image format - BARE: the image does not have a container or metadata envelope - DOCKER: docker container format - OVA: OVF package in a tarfile - OVF: OVF container format See note 1.
diskFormat	Enum (inlined)	0..1	Disk format of a software image is the format of the underlying disk image. Permitted values: - AKI: a kernel image format - AMI: a machine image format - ARI: a ramdisk image format - ISO: an archive format for the data contents of an optical disc, such as CD-ROM - QCOW2: a common disk image format, which can expand dynamically and supports copy on write - RAW: an unstructured disk image format - VDI: a common disk image format - VHD: a common disk image format - VHDX: enhanced version of VHD format - VMDK: a common disk image format See notes 2 and 3.
createdAt	DateTime	1	Time when this software image was created.
minDisk	UnsignedInt	0..1	The minimal disk for this software image in bytes. See note 4.
minRam	UnsignedInt	0..1	The minimal RAM for this software image in bytes. See note 3.
size	UnsignedInt	1	Size of this software image in bytes.

Attribute name	Data type	Cardinality	Description
userMetadata	KeyValuePairs	0..1	User-defined data.
imagePath	String	0..1	Path which identifies the image artifact and also allows to access a copy of the image artifact. For a software image contained as a file in the VNF package, this attribute shall be present, and the value of this attribute shall start with the name of the first segment in the path in the package, i.e. it shall not be prefixed by path separator characters such as "." and "/". EXAMPLE: foo/bar/m%40ster.vhd For an external software image represented as a URI in the VNF descriptor, this attribute shall be present if the image artifact has been downloaded by the NFVO and shall be absent otherwise. If present, it shall contain the artifactPath under which the image artifact can be obtained using the "Individual artifact in a VNF package" resource defined in clause 10.4.6. It is the responsibility of the NFVO to synthesize this path in a manner that avoids any collision of the synthesized artifact path with the paths and names of image artifacts included in the package.
imageUri	Uri	0..1	URI of the image artifact as defined in the VNF package manifest. Shall be present if the image artifact is external to the VNF package and shall be absent otherwise. EXAMPLE: https://example.com/m%40ster.vhd
NOTE 1: The list of permitted values was taken from "Container formats" in [i.5]. NOTE 2: The list of permitted values was adapted from "Disk formats" in [i.5]. NOTE 3: The attribute shall be present for VM-based software images referenced from a Vdu, and shall be absent otherwise. NOTE 4: The attribute shall be present for software images referenced from a VirtualStorageDesc, and shall be absent otherwise.			

10.5.3.3 Type: VnfPackageArtifactInfo

This type represents an artifact other than a software image which is contained in or external to a VNF package. It shall comply with the provisions defined in table 10.5.3.3-1.

Table 10.5.3.3-1: Definition of the VnfPackageArtifactInfo data type

Attribute name	Data type	Cardinality	Description
artifactPath	String	0..1	Path which identifies the artifact and also allows to access a copy of the artifact. For an artifact contained as a file in the VNF package, this attribute shall be present, and the value of this attribute shall start with the name of the first segment in the path in the package, i.e. it shall not be prefixed by path separator characters such as "." and "/". EXAMPLE: foo/bar/m@ster.sh For an external artifact represented as a URI in the VNF descriptor, this attribute shall be present if the artifact has been downloaded by the NFVO and shall be absent otherwise. If present, it shall contain the artifactPath under which the artifact can be obtained using the "Individual artifact in a VNF package" resource defined in clause 10.4.6. It is the responsibility of the NFVO to synthesize this path in a manner that avoids any collision of the synthesized artifact path with the paths and names of artifacts included in the package.

Attribute name	Data type	Cardinality	Description
artifactURI	Uri	0..1	URI of the artifact as defined in the VNF package manifest. Shall be present if the artifact is external to the package and shall be absent otherwise. EXAMPLE: https://example.com/m%40ster.sh
checksum	Checksum	1	Checksum of the artifact file. Permitted hash algorithms are defined in ETSI GS NFV-SOL 004 [2].
isEncrypted	Boolean	1	Reflects whether the artifact is encrypted (true) or not (false).
nonManoArtifactSetId	String	0..1	Non-MANO artifact set identifier of the non-MANO artifact set to which the artifact belongs, as defined in clause 4.3.7 of ETSI GS NFV-SOL 004 [2]. Shall be provided if the artifact is a non-MANO artifact, and shall be omitted otherwise.
artifactClassification	Enum (inlined)	0..1	Marks specific types of artifacts as defined in the VNF package. If none of the specific classes listed below applies, the attribute shall not be present. Valid values: - HISTORY: a history artifact as per clause 4.3.3 in ETSI GS NFV-SOL 004 [2] - TESTING: a testing artifact as per clause 4.3.4 in ETSI GS NFV-SOL 004 [2] - LICENSE: a license artifact as per clause 4.3.5 in ETSI GS NFV-SOL 004 [2]
metadata	KeyValuePairs	0..1	The metadata of the artifact that are available in the VNF package, such as Content type, size, creation date, etc.

10.5.3.4 Type: PkgmNotificationsFilter

This type represents a subscription filter related to notifications related to VNF package management. It shall comply with the provisions defined in table 10.5.3.4-1.

At a particular nesting level in the filter structure, the following applies: All attributes shall match in order for the filter to match (logical "and" between different filter attributes). If an attribute is an array, the attribute shall match if at least one of the values in the array matches (logical "or" between the values of one filter attribute).

Table 10.5.3.4-1: Definition of the PkgmNotificationsFilter data type

Attribute name	Data type	Cardinality	Description
notificationTypes	Enum (inlined)	0..N	Match particular notification types. Permitted values: • VnfPackageOnboardingNotification • VnfPackageChangeNotification See note 1.
vnfProductsFromProviders	Structure (inlined)	0..N	If present, match VNF packages that contain VNF products from certain providers. See note 2.
>vnfProvider	String	1	Name of the VNF provider to match.
>vnfProducts	Structure (inlined)	0..N	If present, match VNF packages that contain VNF products with certain product names, from one particular provider.
>>vnfProductName	String	1	Name of the VNF product to match.
>>versions	Structure (inlined)	0..N	If present, match VNF packages that contain VNF products with certain versions and a certain product name, from one particular provider.
>>>vnfSoftwareVersion	Version	1	VNF software version to match.

Attribute name	Data type	Cardinality	Description
>>>vnfdVersions	Version	0..N	If present, match VNF packages that contain VNF products with certain VNFD versions, a certain software version and a certain product name, from one particular provider.
vnfdId	Identifier	0..N	Match VNF packages with a VNFD identifier listed in the attribute. See note 2.
vnfPkgId	Identifier	0..N	Match VNF packages with a package identifier listed in the attribute. May be present if the "notificationTypes" attribute contains the value "VnfPackageChangeNotification" and shall be absent otherwise. See note 2.
operationalState	PackageOperationalStateType	0..N	Match particular operational states of the VNF package. May be present if the "notificationTypes" attribute contains the value "VnfPackageChangeNotification" and shall be absent otherwise.
usageState	PackageUsageStateType	0..N	Match particular usage states of the VNF package. May be present if the "notificationTypes" attribute contains the value "VnfPackageChangeNotification" and shall be absent otherwise.
vnfmInfo	String	0..N	Match strings that specify VNFMs compatible with the VNF. See table 10.5.2.2-1.
NOTE 1: The permitted values of the "notificationTypes" attribute are spelled exactly as the names of the notification types to facilitate automated code generation systems. NOTE 2: The attributes "vnfProductsFromProviders", "vnfdId" and "vnfPkgId" are alternatives to reference to particular VNF packages in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.			

10.5.3.5 Type: PkgmLinks

This type represents the links to resources that a VNF package management notification can contain. It shall comply with the provisions defined in table 10.5.3.5-1.

Table 10.5.3.5-1: Definition of the PkgmLinks data type

Attribute name	Data type	Cardinality	Description
vnfPackage	NotificationLink	1	Link to the resource representing the VNF package to which the notified change applies, i.e. the "Individual VNF package" resource that represents the VNF package, identified by the "vnfPkgId" identifier which is managed by the NFVO. This attribute shall be provided by the NFVO but is deprecated and can be removed in future versions of the present document.
vnfPackageByVnfdId	NotificationLink	1	Link to the resource representing the VNF package to which the notified change applies, i.e. the "Individual VNF package" resource that represents the VNF package, identified by the "vnfdId" identifier which is assigned by the VNF vendor.
subscription	NotificationLink	1	Link to the related subscription.

10.5.3.6 Void

10.5.4 Referenced simple data types and enumerations

10.5.4.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

10.5.4.2 Simple data types

No particular simple data types are defined for this interface, in addition to those defined in clause 4.4.

10.5.4.3 Enumeration: PackageOperationalStateType

The enumeration PackageOperationalStateType shall comply with the provisions defined in table 10.5.4.3-1.

Table 10.5.4.3-1: Enumeration PackageOperationalStateType

Enumeration value	Description
ENABLED	The VNF package is enabled, i.e. it can be used for the creation of new "Individual VNF instance" resources.
DISABLED	The VNF package is disabled, i.e. it shall not be used for the creation of further "Individual VNF instance" resources (unless and until the VNF package is re-enabled).

10.5.4.4 Enumeration: PackageUsageStateType

The enumeration PackageUsageStateType shall comply with the provisions defined in table 10.5.4.4-1.

Table 10.5.4.4-1: Enumeration PackageUsageStateType

Enumeration value	Description
IN_USE	"Individual VNF instance" resources created from this VNF package exist.
NOT_IN_USE	No "Individual VNF instance" resource created from this VNF package exists.

10.5.4.5 Enumeration: PackageChangeType

The enumeration PackageChangeType shall comply with the provisions defined in table 10.5.4.5-1.

Table 10.5.4.5-1: Enumeration PackageChangeType

Enumeration value	Description
OP_STATE_CHANGE	The "operationalState" attribute has been changed.
PKG_DELETE	The VNF package has been deleted.

10.5.4.6 Enumeration: PackageOnboardingStateType

The enumeration PackageOnboardingStateType shall comply with the provisions defined in table 10.5.4.6-1.

Table 10.5.4.6-1: Enumeration PackageOnboardingStateType

Enumeration value	Description
CREATED	The "Individual VNF package" resource has been created.
UPLOADING	The associated VNF package content is being uploaded.
PROCESSING	The associated VNF package content is being processed, e.g. validation.
ONBOARDED	The associated VNF package content has been on-boarded successfully.
ERROR	There was an error during upload of the VNF package content or external artifacts, or during VNF package processing.

11 Virtualised Resources Quota Available Notification interface

11.1 Description

This interface allows the VNFM to subscribe to notifications on the availability of the virtualised resources quotas and allows the NFVO to provide such notification to the subscriber. Further, this interface allows API version information retrieval.

Support for this interface is optional.

The operations provided through this interface are:

- Subscribe
- Query Subscription Information
- Terminate Subscription
- Notify

11.1a API version

For the virtualised resources quota available notification interface version as specified in the present document, the MAJOR version field shall be 1, the MINOR version field shall be 10 and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [8] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v1".

11.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [8]. The string "vrqan" shall be used to represent {apiName}. All resource URIs in the clauses below are defined relative to the above base URI.

Figure 11.2-1 shows the overall resource URI structure defined for the Virtualised Resources Quota Available Notification interface.

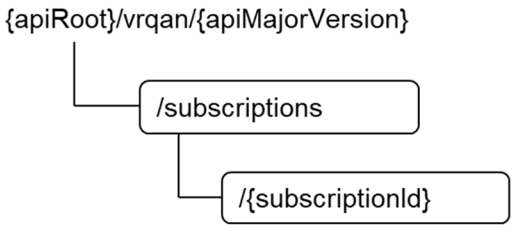

Figure 11.2-1: Resource URI structure of Virtualised Resources Quota Available Notification Interface

Table 11.2-1 lists the individual resources defined, and the applicable HTTP methods.

If the NFVO supports the Virtualised Resources Quota Available Notification interface, the NFVO shall support responding to requests for all HTTP methods on the resources in table 11.2-1 that are marked as "M" (mandatory) in the "Cat" column. The NFVO shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [8].

Table 11.2-1: Resources and methods overview of the Virtualised Resources Quota Available Notification interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
Subscriptions	/subscriptions	POST	M	Subscribe to the notifications related to the availability of the virtualised resources quotas
		GET	M	Query subscriptions
Individual subscription	/subscriptions/{subscriptionId}	GET	M	Read individual subscription
		DELETE	M	Terminate subscription
Notification endpoint	(provided by API consumer)	POST	See note	Notify about the availability of the virtualised resources quota
		GET	See note	Test the notification endpoint
NOTE: If the NFVO supports the Virtualised Resources Quota Available Notification interface, the NFVO shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the VNFM. If the VNFM supports the Virtualised Resources Quota Available Notification interface, it shall support responding to the HTTP requests defined for the "Notification endpoint" resource.				

11.3 Sequence diagrams (informative)

11.3.1 Flow of managing subscriptions

This clause describes the procedure for creating, reading and terminating subscriptions to notifications related to the availability of the virtualised resources quotas.

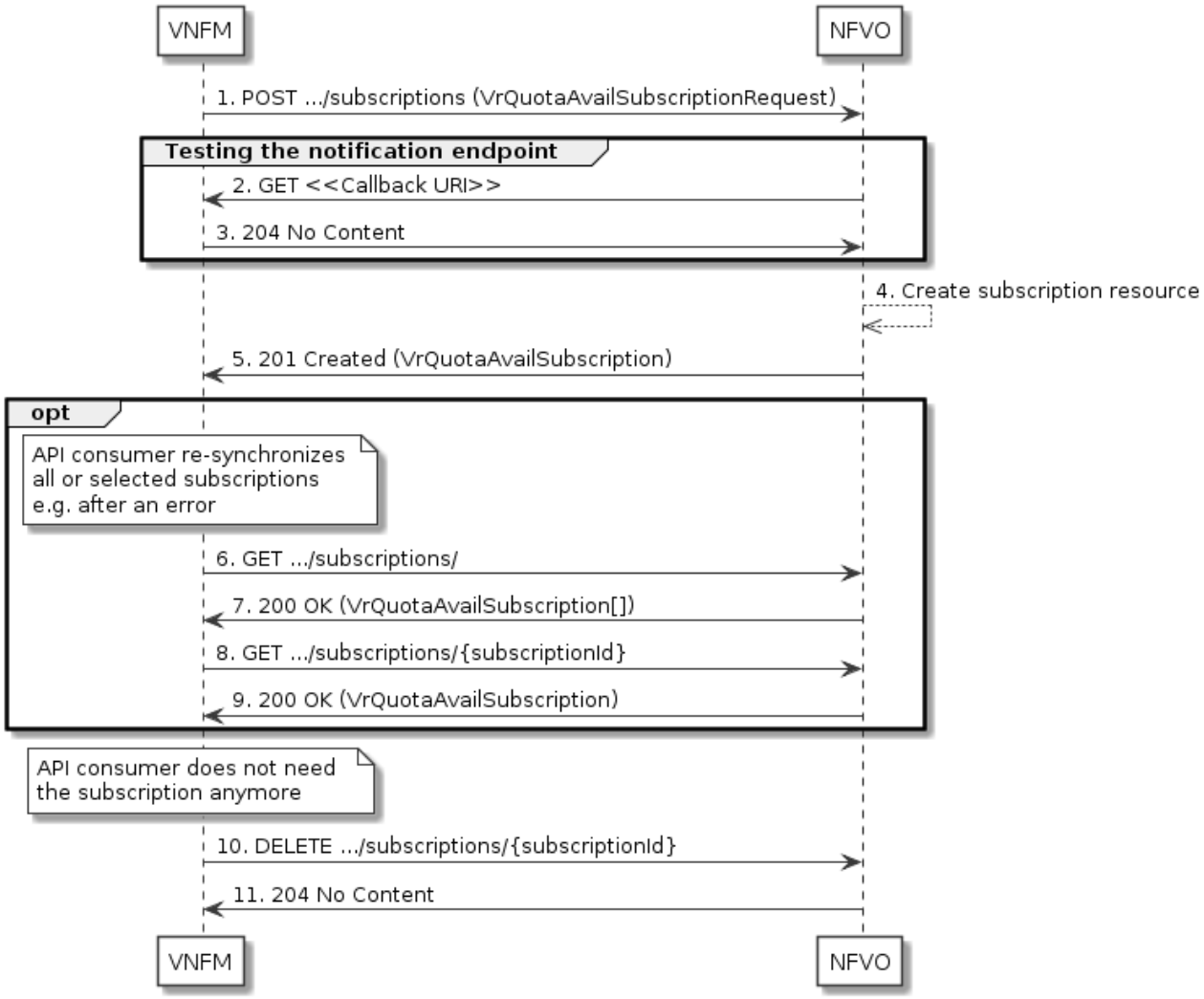

Figure 11.3.1-1: Flow of managing subscriptions

The procedure consists of the following steps as illustrated in figure 11.3.1-1:

1) The VNFM sends a POST request to the "Subscriptions" resource including in the payload body a data structure of type "VrQuotaAvailSubscriptionRequest". That data structure contains filtering criteria and a callback URI to which the NFVO will subsequently send notifications about events that match the filter.

2) To test the notification endpoint that has been registered by the VNFM as part of the subscription, the NFVO sends a GET request to the notification endpoint URI.

3) The VNFM returns a "204 No Content" response to indicate success.

4) The NFVO creates a new subscription to notifications related to the availability of the virtualised resources quotas, and a resource that represents this subscription.

5) The NFVO returns a "201 Created" response containing a data structure of type "VrQuotaAvailSubscription" representing the "Individual subscription" resource just created by the NFVO and provides the URI of the newly-created resource in the "Location" HTTP header.

6) If desired, e.g. to recover from an error situation, the VNFM can obtain information about its subscriptions by sending a GET request to the resource representing the subscriptions.

7) In that case, the NFVO returns a "200 OK" response that contains the list of representations of all existing subscriptions that were created by the VNFM.

8) If desired, the VNFM can obtain information about a particular subscription by sending a GET request to the resource representing that individual subscription.

9) In that case, the NFVO returns a "200 OK" response that contains a representation of that individual subscription.

10) If the VNFM does not need the subscription anymore, it terminates the subscription by sending a DELETE request to the resource that represents the individual subscription to remove.

11) The NFVO acknowledges the successful termination of the subscription by returning a "204 No Content" response.

Error handling: The NFVO rejects a subscription if the subscription information is not valid: endpoint cannot be reached, subscription information is malformed, etc.

11.3.2 Flow of sending notifications

This clause describes the procedure of sending notifications related to the availability of virtualised resources quota.

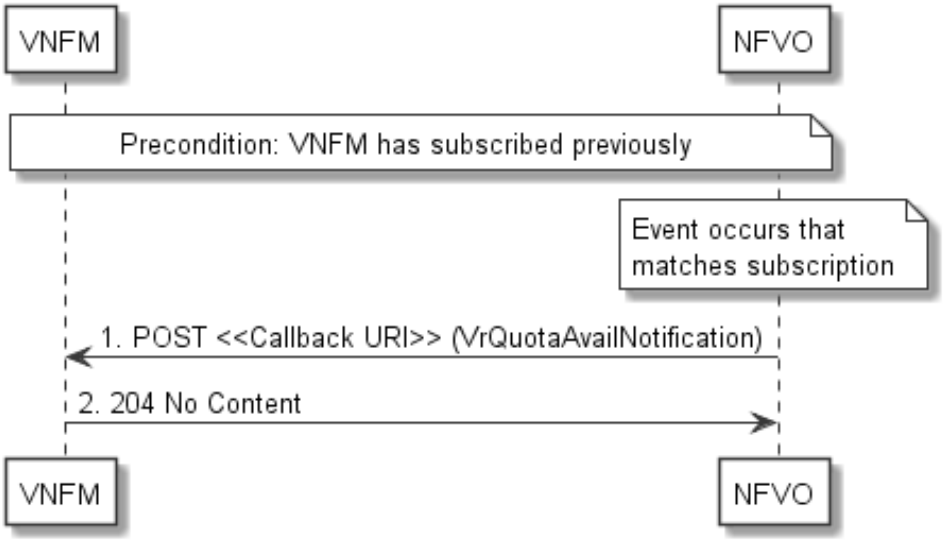

Figure 11.3.2-1: Flow of sending notifications

The procedure consists of the following steps as illustrated in figure 11.3.2-1:

Precondition: The VNFM has subscribed previously to notifications related to the availability of virtualised resources quotas.

1) If an event occurs that matches the filtering criteria which are part of the subscription, the NFVO generates a VrQuotaAvailNotification that includes information about the event and sends it in the body of a POST request to the URI which the VNFM has registered as part of the subscription request.

2) The VNFM acknowledges the successful delivery of the notification by returning a "204 No Content" response.

Error handling: If the NFVO does not receive the "204 No Content" response from the VNFM, it can retry sending the notification.

11.4 Resources

11.4.1 Introduction

This clause defines all the resources and methods provided by the virtualised resources quota available notification interface.

11.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [8] are part of the virtualised resources quota available Notification interface.

11.4.2 Resource: Subscriptions

11.4.2.1 Description

This resource represents subscriptions. The API consumer can use this resource to subscribe to notifications related to the availability of the virtualised resources quotas, and to query its subscriptions.

11.4.2.2 Resource definition

The resource URI is:

{apiRoot}/vrqan/{apiMajorVersion}/subscriptions

This resource shall support the resource URI variables defined in table 11.4.2.2-1.

Table 11.4.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 11.1a.

11.4.2.3 Resource methods

11.4.2.3.1 POST

The POST method creates a new subscription.

This method shall follow the provisions specified in tables 11.4.2.3.1-1 and 11.4.2.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual subscription" resource as defined in clause 11.4.3 shall have been created. This method shall not trigger any notification.

Creation of two "Individual subscription" resources with the same callback URI and the same filter can result in performance degradation and will provide duplicates of notifications to the VNFM, and might make sense only in very rare use cases. Consequently, the NFVO may either allow creating a new "Individual subscription" resource if another "Individual subscription" resource with the same filter and callback URI already exists (in which case it shall return the "201 Created" response code), or may decide to not create a duplicate "Individual subscription" resource (in which case it shall return a "303 See Other" response code referencing the existing "Individual subscription" resource with the same filter and callback URI).

Table 11.4.2.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 11.4.2.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	VrQuotaAvailSubscriptionRequest	1	Details of the subscription to be created.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VrQuotaAvailSubscription	1	201 Created	Shall be returned when the subscription has been created successfully. The response body shall contain a representation of the created "Individual subscription" resource. The HTTP response shall include a "Location" HTTP header that points to the created resource.
	n/a		303 See Other	Shall be returned when a subscription with the same callback URI and the same filter already exists and the policy of the NFVO is to not create redundant subscriptions. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the existing "Individual subscription" resource. The response body shall be empty.
	ProblemDetails	1	422 Unprocessable Entity	Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [8], including rules for the presence of the response body. Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 11.4.4.3.2 and the test has failed. In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

11.4.2.3.2 GET

The GET method queries the list of active subscriptions of the functional block that invokes the method. It can be used e.g. for resynchronization after error situations.

This method shall follow the provisions specified in tables 11.4.2.3.2-1 and 11.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 11.4.2.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [8]. The NFVO shall support receiving this parameter as part of the URI query string. The VNFM may supply this parameter. All attribute names that appear in the VrQuotaAvailSubscription and in data types referenced from it shall be supported by the NFVO in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the VNFM if the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource.

Table 11.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Response Codes	Description
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VrQuotaAvailSubscription	0..N	200 OK	Shall be returned when the list of subscriptions has been queried successfully. The response body shall contain in an array the representations of all active subscriptions of the functional block that invokes the method, i.e. zero or more representations of virtualised resource quota available subscriptions as defined in clause 11.5.2.3. If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [8]. If the VNFM supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the VNFM supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

11.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

11.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

11.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

11.4.3 Resource: Individual subscription

11.4.3.1 Description

This resource represents an individual subscription. The API consumer can use this resource to read and to terminate a subscription to notifications related to the availability of the virtualised resources quotas.

11.4.3.2 Resource definition

The resource URI is:

{apiRoot}/vrqan/{apiMajorVersion}/subscriptions/{subscriptionId}

This resource shall support the resource URI variables defined in table 11.4.3.2-1.

Table 11.4.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 11.1a.
subscriptionId	Identifier of this subscription. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual subscription" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

11.4.3.3 Resource methods

11.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

11.4.3.3.2 GET

The GET method reads an individual subscription.

This method shall follow the provisions specified in tables 11.4.3.3.2-1 and 11.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 11.4.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 11.4.3.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VrQuotaAvailSubscription	1	200 OK	Shall be returned when information about an individual subscription has been read successfully. The response body shall contain a representation of the "Individual subscription" resource.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

11.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

11.4.3.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

11.4.3.3.5 DELETE

The DELETE method terminates an individual subscription.

This method shall follow the provisions specified in tables 11.4.3.3.5-1 and 11.4.3.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual subscription" resource shall not exist any longer. This means that no notifications for that subscription shall be sent to the formerly-subscribed API consumer.

NOTE: Due to race conditions, some notifications might still be received by the formerly-subscribed API consumer for a certain time period after the deletion.

Table 11.4.3.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 11.4.3.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned when the "Individual subscription" resource has been deleted successfully.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

11.4.4 Resource: Notification endpoint

11.4.4.1 Description

This resource represents a notification endpoint.

The API producer can use this resource to send notifications related to virtualised resources quota availability to a subscribed API consumer, which has provided the URI of this resource during the subscription process.

11.4.4.2 Resource definition

The resource URI is provided by the API consumer when creating the subscription.

This resource shall support the resource URI variables defined in table 11.4.4.2-1.

Table 11.4.4.2-1: Resource URI variables for this resource

Name	Definition
none suported	

11.4.4.3 Resource methods

11.4.4.3.1 POST

The POST method delivers a notification from the API producer to an API consumer. The API consumer shall have previously created an "Individual subscription" resource with a matching filter.

This method shall follow the provisions specified in tables 11.4.4.3.1-1 and 11.4.4.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 11.4.4.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 11.4.4.3.1-2: Details of the POST request/response on this resource

	Data type	Cardinality	Description	
Request body	VrQuotaAvailNotification	1	A notification related to the availability of the virtualised resources quota.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned when the notification has been delivered successfully.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

11.4.4.3.2 GET

The GET method allows the API producer to test the notification endpoint that is provided by the API consumer, e.g. during subscription.

This method shall follow the provisions specified in tables 11.4.4.3.2-1 and 11.4.4.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 11.4.4.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 11.4.4.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	n/a		204 No Content	Shall be returned to indicate that the notification endpoint has been tested successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

11.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

11.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

11.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the VNFM shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

11.5 Data model

11.5.1 Introduction

This clause defines the request and response data structures of the Virtualised Resources Quota Available Notification interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error and may choose to ignore them.

11.5.2 Resource and notification data types

11.5.2.1 Introduction

This clause defines data structures to be used in resource representations and notifications.

11.5.2.2 Type: VrQuotaAvailSubscriptionRequest

This type represents a subscription request related to notifications related to the availability of the virtualised resources quotas. It shall comply with the provisions defined in table 11.5.2.2-1.

Table 11.5.2.2-1: Definition of the VrQuotaAvailSubscriptionRequest data type

Attribute name	Data type	Cardinality	Description
filter	VrQuotaAvailNotificationsFilter	0..1	Input filter for selecting notifications to subscribe to. This filter can contain information about specific attributes of the virtualised resources quota.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this subscription, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [8]. This attribute shall only be present if the subscriber requires authorization of notifications.

11.5.2.3 Type: VrQuotaAvailSubscription

This type represents a subscription related to notifications related to the availability of the virtualised resources quotas. It shall comply with the provisions defined in table 11.5.2.3-1.

Table 11.5.2.3-1: Definition of the VrQuotaAvailSubscription data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this "Individual subscription" resource.
filter	VrQuotaAvailNotificationsFilter	0..1	Input filter for selecting notifications to subscribe to. This filter can contain information about specific attributes of the virtualised resources quota.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
_links	Structure (inlined)	1	Links for this resource.
>self	Link	1	URI of this resource.

11.5.2.4 Type: VrQuotaAvailNotification

This type represents a notification which indicates the availability of a quota applicable to the consumer. It shall comply with the provisions defined in table 11.5.2.4-1. Support of this notification is mandatory if the Virtualised Resources Quota Available Notification interface is supported.

The notification shall be triggered by the NFVO when a virtualised resource quota applicable to the consumer has been set.

Table 11.5.2.4-1: Definition of the VrQuotaAvailNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "VrQuotaAvailNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
resourceGroupId	IdentifierInVim	1	Identifier of the "infrastructure resource group", logical grouping of virtual resources assigned to a tenant within an Infrastructure Domain.
vimConnectionInfo	VimConnectionInfo	0..1	Information about the VIM connection to manage the virtualised resources quota. This attribute shall only be supported and present when VNF-related Resource Management in direct mode is applicable.

Attribute name	Data type	Cardinality	Description
resourceProviderId	Identifier	0..1	Identifies the entity responsible for the management of the virtualised resources quota. This attribute shall only be supported and present when VNF-related Resource Management in indirect mode is applicable. The identification scheme is outside the scope of the present document.
_links	QuotaAvailLinks	1	Links to resources related to this notification.

11.5.3 Referenced structured data types

11.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but can neither be resource representations nor bound to any subscribe/notify mechanism.

11.5.3.2 Type: VrQuotaAvailNotificationsFilter

This type represents a subscription filter related to notifications about the availability of the virtualised resources quotas. It shall comply with the provisions defined in table 11.5.3.2-1.

At a particular nesting level in the filter structure, the following applies: All attributes shall match in order for the filter to match (logical "and" between different filter attributes). If an attribute is an array, the attribute shall match if at least one of the values in the array matches (logical "or" between the values of one filter attribute).

Table 11.5.3.2-1: Definition of the VrQuotaAvailNotificationsFilter data type

Attribute name	Data type	Cardinality	Description
vimIds	Identifier	0..N	Match VIMs that were created the quota for a consumer of the virtualised resources. This attribute shall only be supported when VNF-related Resource Management in direct mode is applicable.
resourceProviderIds	Identifier	0..N	Match the entities responsible for the management of the virtualised resources that were allocated by the NFVO. This attribute shall only be supported when VNF-related Resource Management in indirect mode is applicable. The identification scheme is outside the scope of the present document.
resourceTypes	Enum (inlined)	0..N	Match particular resource types. Permitted values: - COMPUTE - STORAGE - NETWORK
resourceGroupIds	IdentifierInVim	0..N	Match the "infrastructure resource groups" that are logical groupings of the virtualised resources assigned to a tenant within an infrastructure Domain.

11.5.3.3 Type: QuotaAvailLinks

This type represents the links to resources that a notification of type "VrQuotaAvailNotification" can contain. It shall comply with the provisions defined in table 11.5.3.3-1.

Table 11.5.3.3-1: Definition of the QuotaAvailLinks data type

Attribute name	Data type	Cardinality	Description
subscription	NotificationLink	1	Link to the related subscription.

12 VNF Snapshot Package Management interface

12.1 Description

This interface allows the VNFM to access VNF snapshot package information and to fetch VNF snapshot packages from/to the NFVO.

The operations provided through this interface are:

- Fetch VNF Snapshot Package.
- Fetch VNF Snapshot Package Artifacts.
- Query VNF Snapshot Package Information.

This interface also enables API version information retrieval.

12.1a API version

For the VNF snapshot package management interface as specified in the present document, the MAJOR version field shall be 1, the MINOR version field shall be 10 and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [8] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v1".

12.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [8]. The string "vnfsnapshotpkgm" shall be used to represent {apiName}. All resource URIs in clauses below are defined relative to the above base URI.

Figure 12.2-1 shows the overall resource URI structure defined for the VNF snapshot package management interface.

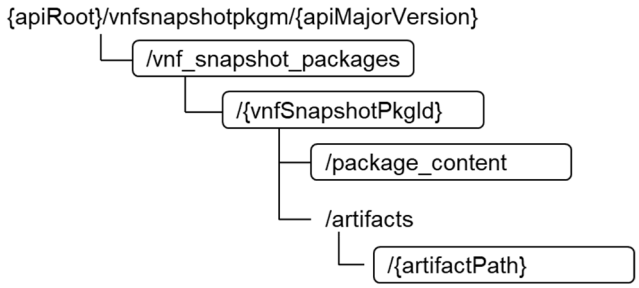

Figure 12.2-1: Resource URI structure of the VNF snapshot package management interface

Table 12.2-1 lists the individual resources defined, and the applicable HTTP methods.

The NFVO shall support responding to requests for all HTTP methods on the resources in table 12.2-1 that are marked as "M" (mandatory) in the "Cat" column. The NFVO shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [8].

Table 12.2-1: Resources and methods overview of the VNF snapshot package management interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
VNF snapshot packages	/vnf_snapshot_packages	GET	M	Query multiple VNF snapshot packages information.
Individual VNF snapshot package	/vnf_snapshot_packages/{vnfSnapshotPkgId}	GET	M	Read an "Individual VNF snapshot package" resource.
VNF snapshot package content	/vnf_snapshot_packages/ {vnfSnapshotPkgId}/package_content	GET	M	Fetch VNF snapshot package.
Individual VNF snapshot package artifact	/vnf_snapshot_packages/ {vnfSnapshotPkgId}/artifacts/{artifactPath}	GET	M	Fetch individual VNF snapshot package artifact.

12.3 Sequence diagrams (informative)

12.3.1 Flow of querying/reading VNF snapshot package information

This clause describes the procedure for querying information about one or multiple VNF snapshot packages.

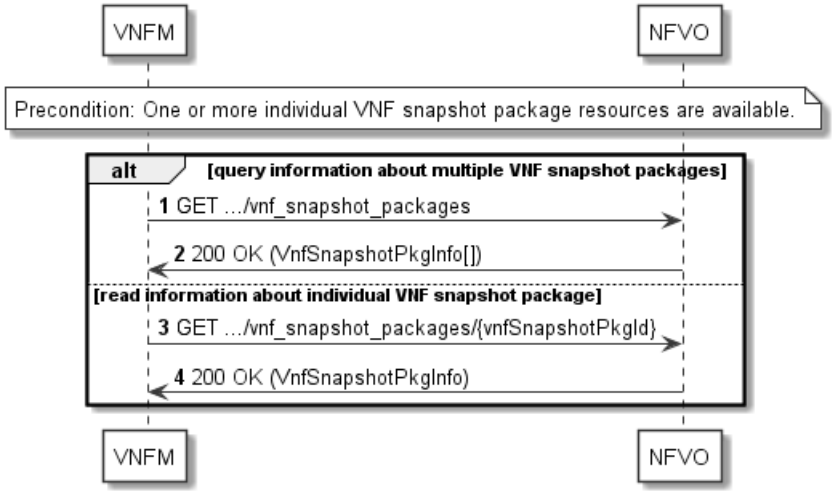

Figure 12.3.1-1: Flow of querying/reading VNF snapshot package information

Precondition: One or more "Individual VNF package" resources are created.

VNF snapshot package information query, as illustrated in figure 12.3.1-1, consists of the following steps:

1) If the VNFM intends to query information about multiple VNF snapshot packages, it sends a GET request to the "VNF snapshot packages" resource.

2) The NFVO returns a "200 OK" response and includes in the payload body zero or more data structures of type "VnfSnapshotPkgInfo".

3) If the VNFM intends to read information about a particular VNF snapshot package, the VNFM sends a GET request to the "Individual VNF snapshot package" resource, addressed by the appropriate VNF snapshot package identifier in its resource URI.

4) The NFVO returns a "200 OK" response and includes in the payload body a data structure of type "VnfSnapshotPkgInfo".

Postcondition: Upon successful completion, the VNFM gets the information about one or more VNF snapshot packages.

Error handling: In case of failure, appropriate error information is provided in the response.

12.3.2 Flow of fetching a VNF snapshot package

This clause describes the procedure for fetching the content of an available VNF snapshot package.

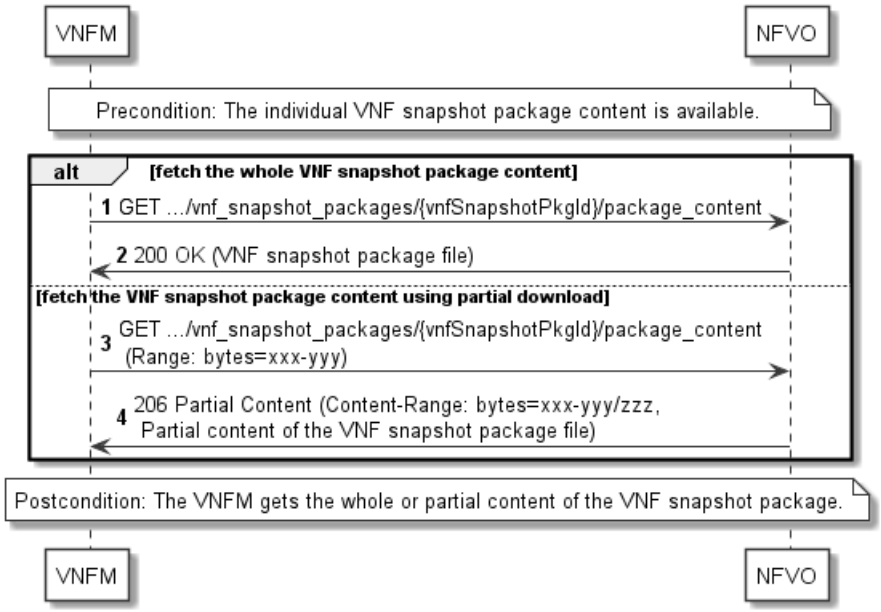

Figure 12.3.2-1: Flow of fetching a VNF snapshot package

Precondition: The VNF snapshot package is available on the NFVO.

Fetching an available VNF snapshot package, as illustrated in figure 12.3.2-1, consists of the following steps:

1) If fetching the whole VNF snapshot package content, the VNFM sends a GET request to the "VNF snapshot package content" resource.

2) The NFVO returns a "200 OK" response and includes a copy of the VNF snapshot package file in the payload body.

3) If fetching the VNF snapshot package content using partial download, the VNFM sends a GET request to the "VNF snapshot package content" resource and includes a "Range" HTTP header indicating the partition of the VNF snapshot package content needs to be transferred.

4) The NFVO returns a "206 Partial Content" response with a payload body containing the partial content of the VNF snapshot package, and a "Content-Range" HTTP header indicating the byte range enclosed in the payload and the complete length of the VNF snapshot package content.

Postcondition: Upon successful completion, the VNFM gets the whole or partial content of the VNF snapshot package.

Error handling: In case of failure, appropriate error information is provided in the response.

12.3.3 Flow of fetching a VNF snapshot package artifact

This clause describes the procedure for fetching an individual artifact contained in an available VNF snapshot package.

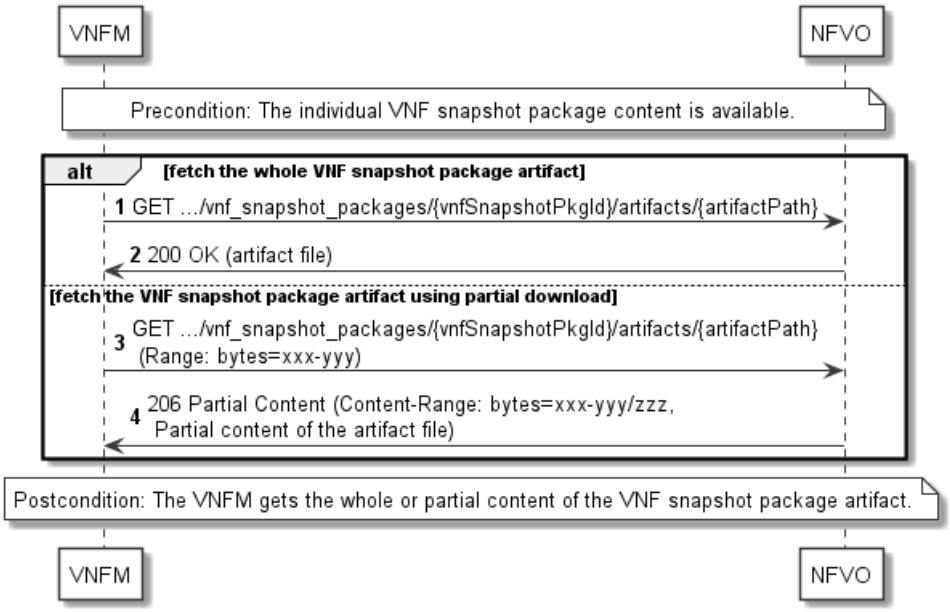

Figure 12.3.3-1: Flow of fetching a VNF snapshot package artifact

Precondition: The VNF snapshot package is available on the NFVO.

Fetching an individual artifact contained in an available VNF snapshot package, as illustrated in figure 12.3.3-1, consists of the following steps:

1) If fetching the whole content of the artifact, the VNFM sends a GET request to the "Individual VNF snapshot package artifact" resource.

2) The NFVO returns a "200 OK" response and includes a copy of the applicable artifact file from the VNF snapshot package in the payload body.

3) If fetching the artifact using partial download, the VNFM sends a GET request to the "Individual VNF snapshot package artifact" resource and includes a "Range" HTTP header indicating the partition of the artifact needs to be transferred.

4) The NFVO returns a "206 Partial Content" response with a payload body containing the partial content of the artifact file, and a "Content-Range" HTTP header indicating the byte range enclosed in the payload and the complete length of the artifact file.

Postcondition: The VNFM gets the whole or partial content of the VNF snapshot package artifact.

Error handling: In case of failure, appropriate error information is provided in the response.

12.4 Resources

12.4.1 Introduction

This clause defines all the resources and methods provided by the VNF snapshot package management interface.

12.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [8] are part of the VNF snapshot package management interface.

12.4.2 Resource: VNF snapshot packages

12.4.2.1 Description

This resource represents VNF snapshot packages. The API consumer can use this resource to create "Individual VNF snapshot package" resources, and to query information of the VNF snapshot packages.

12.4.2.2 Resource definition

The resource URI is:

{apiRoot}/vnfsnapshotpkgm/{apiMajorVersion}/vnf_snapshot_packages

This resource shall support the resource URI variables defined in table 12.4.2.2-1.

Table 12.4.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 12.1a.

12.4.2.3 Resource methods

12.4.2.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

12.4.2.3.2 GET

The GET method queries the information of the VNF packages matching the filter.

This method shall follow the provisions specified in tables 12.4.2.3.2-1 and 12.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 12.4.2.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [8]. The NFVO shall support receiving this parameter as part of the URI query string. The VNFM may supply this parameter. All attribute names that appear in the VnfSnapshotPkgInfo and in data types referenced from it shall be supported by the NFVO in the filter expression.
all_fields	0..1	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The NFVO shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The NFVO should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The NFVO should support this parameter.

Name	Cardinality	Description
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [8] for details. The NFVO shall support this parameter. The following attributes shall be excluded from the VnfSnapshotPkgInfo structure in the response body if this parameter is provided, or none of the parameters "all_fields," "fields", "exclude_fields", "exclude_default" are provided: - vnfcSnapshotImages - additionalArtifacts - userDefinedData - checksum
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource.

Table 12.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfSnapshotPkgInfo	0..N	200 OK	Shall be returned when information about zero or more VNF snapshot packages has been queried successfully. The response body shall contain in an array the VNF snapshot package info representations that match the attribute filter, i.e. zero or more VNF snapshot package info representations as defined in clause 12.5.2.2. If the "filter" URI parameter or one of the "all_fields", "fields", "exclude_fields" or "exclude_default" URI parameters was supplied in the request and is supported, the data in the response body shall have been transformed according to the rules specified in clauses 5.2.2 and 5.3.2 of ETSI GS NFV-SOL 013 [8], respectively. If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute selector. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [8] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [8].
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

12.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

12.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

12.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

12.4.3 Resource: Individual VNF snapshot package

12.4.3.1 Description

This resource represents an individual VNF snapshot package. The API consumer can use this resource to read information of the VNF snapshot package, update information of the VNF snapshot package, or delete a VNF snapshot package.

12.4.3.2 Resource definition

The resource URI is:

{apiRoot}/vnfsnapshotpkgm/{apiMajorVersion}/vnf_snapshot_packages/{vnfSnapshotPkgId}

This resource shall support the resource URI variables defined in table 12.4.3.2-1.

Table 12.4.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 12.1a.
vnfSnapshotPkgId	Identifier of the VNF snapshot package. The identifier is allocated by the NFVO. See note.
NOTE: This identifier can be retrieved from the "id" attribute of the applicable "VnfSnapshotPkgInfo" in the body of the response to requesting the creation of a new "Individual VNF snapshot package" resource or in a response to a GET request querying the "Individual VNF snapshot package" or the "VNF snapshot packages" resource.	

12.4.3.3 Resource methods

12.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

12.4.3.3.2 GET

The GET method reads the information of an individual VNF snapshot package.

This method shall follow the provisions specified in tables 12.4.3.3.2-1 and 12.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 12.4.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 12.4.3.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Description**
	VnfSnapshotPkgInfo	1	200 OK	Shall be returned when information of the VNF snapshot package has been read successfully. The response body shall contain the VNF snapshot package info representation defined in clause 12.5.2.2.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

12.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

12.4.3.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

12.4.3.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

12.4.4 Resource: VNF snapshot package content

12.4.4.1 Description

This resource represents a VNF snapshot package identified by the VNF snapshot package identifier allocated by the NFVO. The API consumer can use this resource to upload and fetch the content of the VNF snapshot package.

12.4.4.2 Resource definition

The resource URI is:

{apiRoot}/vnfsnapshotpkgm/{apiMajorVersion}/vnf_snapshot_packages/{vnfSnapshotPkgId}/package_content

This resource shall support the resource URI variables defined in table 12.4.4.2-1.

Table 12.4.4.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 12.1a.
vnfSnapshotPkgId	Identifier of the VNF snapshot package. The identifier is allocated by the NFVO. See note.
NOTE: This identifier can be retrieved from the "id" attribute of the applicable "VnfSnapshotPkgInfo" in the body of the response to requesting the creation of a new "Individual VNF snapshot package" resource or in a response to a GET request querying the "Individual VNF snapshot package" or the "VNF snapshot packages" resource.	

12.4.4.3 Resource methods

12.4.4.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

12.4.4.3.2 GET

The GET method fetches the content of a VNF snapshot package.

This method shall follow the provisions specified in tables 12.4.4.3.2-1 and 12.4.4.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 12.4.4.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 12.4.4.3.2-2: Details of the GET request/response on this resource

	Data type	Cardinality	Description
Request body	n/a		The request may contain a "Range" HTTP header to obtain single range of bytes from the VNF snapshot package file. This can be used to continue an aborted transmission. If the Range header is present in the request and the NFVO does not support range requests with a 206 response, it shall return the whole file with a 200 OK response instead as defined below.

	Data type	Cardinality	Response Codes	Description
Response body	n/a	1	200 OK	Shall be returned when the whole content of the VNF snapshot package file has been read successfully. The response body shall include a copy of the VNF snapshot package file. The "Content-Type" HTTP header shall be set according to the type of the file, i.e. to "application/zip" for a VNF snapshot package. The VNF snapshot package format is defined in ETSI GS NFV-SOL 010 [i.14].
	n/a	1	206 Partial Content	If the NFVO supports range requests, this response shall be returned when a single consecutive byte range from the content of the VNF snapshot package file has been read successfully according to the request. The response body shall contain the requested part of the VNF snapshot package file. The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [4]. The "Content-Type" HTTP header shall be set as defined above for the "200 OK" response.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact the "state" of the VNF snapshot package has a value different from "AVAILABLE". The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	416 Range Not Satisfiable	Shall be returned upon the following error: The byte range passed in the "Range" header did not match any available byte range in the VNF snapshot package file (e.g. "access after end of file"). The response body may contain a ProblemDetails structure.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

12.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

12.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

12.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

12.4.5 Resource: Individual VNF snapshot package artifact

12.4.5.1 Description

This resource represents an individual artifact contained in a VNF snapshot package. The API consumer can use this resource to fetch the content of the artifact.

12.4.5.2 Resource definition

The resource URI is:

{apiRoot}/vnfsnapshotpkgm/{apiMajorVersion}/vnf_snapshot_packages/{vnfSnapshotPkgId}/artifacts/{artifactPath}

This resource shall support the resource URI variables defined in table 12.4.5.2-1.

Table 12.4.5.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [8].
apiMajorVersion	See clause 12.1a.
vnfSnapshotPkgId	Identifier of the VNF snapshot package. The identifier is allocated by the NFVO. See note 1.
artifactPath	For an artifact contained as a file in the VNF snapshot package, this variable shall contain a sequence of one or path segments representing the path of the artifact within the VNF snapshot package, relative to the root of the package. EXAMPLE: foo/bar/m%40ster.sh For an external artifact represented as a URI in the VNF snapshot package manifest, this variable shall contain a sequence of one or more path segments as synthesized by the NFVO (see clause 12.5.3.3) representing this artifact. See notes 2 and 3.
NOTE 1: This identifier can be retrieved from the "id" attribute of the applicable "VnfSnapshotPkgInfo" in the body of the response to requesting the creation of a new "Individual VNF snapshot package" resource or in a response to a GET request querying the "Individual VNF snapshot package" or the "VNF snapshot packages" resource.	
NOTE 2: This identifier can be retrieved from the "artifactPath" attribute of the applicable "additionalArtifacts" entry in the body of the response to a GET request querying the "Individual VNF snapshot package" or the "VNF snapshot packages" resource.	
NOTE 3: Since multiple path segments are allowed to be contained in this variable, the "/" character that separates these segments is not percent-encoded. Each individual segment is percent-encoded if necessary as defined in clause 4.1 of ETSI GS NFV-SOL 013 [8].	

12.4.5.3 Resource methods

12.4.5.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

12.4.5.3.2 GET

The GET method fetches the content of an artifact within the VNF snapshot package.

This method shall follow the provisions specified in tables 12.4.5.3.2-1 and 12.4.5.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 12.4.5.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 12.4.5.3.2-2: Details of the GET request/response on this resource

	Data type	Cardinality	Response Codes	Description
Request body	n/a			The request may contain a "Range" HTTP header to obtain single range of bytes from the artifact file. This can be used to continue an aborted transmission. If the Range header is present in the request and the NFVO does not support range requests with a 206 response, it shall return the whole file with a 200 OK response instead as defined below.
Response body	n/a	1	200 OK	Shall be returned when the whole content of the artifact file has been read successfully. The response body shall include a copy of the artifact file from the VNF snapshot package. The VNF snapshot package format is defined in ETSI GS NFV-SOL 010 [i.14]. The "Content-Type" HTTP header shall be set according to the content type of the artifact file. If the content type cannot be determined, the header shall be set to the value "application/octet-stream".
	n/a	1	206 Partial Content	If the NFVO supports range requests, this response shall be returned when a single consecutive byte range from the content of the artifact file has been read successfully according to the request. The response body shall contain the requested part of the artifact file from the VNF snapshot package. The VNF snapshot package is defined in ETSI GS NFV-SOL 010 [i.14]. The "Content-Type" HTTP header shall be set according to the content type of the artifact file. If the content type cannot be determined, the header shall be set to the value "application/octet-stream". The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [4].
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact the "state" of the VNF snapshot package has a value different from "AVAILABLE". The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.

	Data type	Cardinality	Response Codes	Description
Response body	ProblemDetails	0..1	416 Range Not Satisfiable	Shall be returned upon the following error: The byte range passed in the "Range" header did not match any available byte range in the artifact file (e.g. "access after end of file"). The response body may contain a ProblemDetails structure.
	ProblemDetails	See clause 6.4 of [8]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8] may be returned.

12.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

12.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

12.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [8].

12.5 Data model

12.5.1 Introduction

This clause defines the request and response data structures of the VNF snapshot package management interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error, and may choose to ignore them.

12.5.2 Resource and notification data types

12.5.2.1 Introduction

This clause defines data structures to be used in resource representations and notifications.

12.5.2.2 Type: VnfSnapshotPkgInfo

This type represents the information of a VNF snapshot package. It shall comply with the provisions defined in table 12.5.2.2-1.

Table 12.5.2.2-1: Definition of the VnfSnapshotPkgInfo data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the VNF snapshot package information held by the NFVO. This identifier is allocated by the NFVO.
vnfSnapshotPkgUniqueId	Identifier	0..1	Identifier of the VNF snapshot package, which identifies the VNF snapshot package in a globally unique way. It is created during the "build VNF snapshot package operation". Multiples instances of the same VNF snapshot package share the same vnfSnapshotPkgUniqueId. See note.
name	String	1	Human-readable name of the VNF snapshot package.
checksum	Checksum	0..1	Checksum of the stored VNF snapshot package. Hash algorithms applicable to VNF snapshot packages are defined in ETSI GS NFV-SOL 010 [i.14]. See note.
createdAt	DateTime	0..1	Timestamp indicating when the VNF snapshot package creation has been completed. See note.
vnfSnapshotId	Identifier	0..1	Identifier of the specific VNF snapshot in the VNF snapshot package. This identifier is allocated by the VNFM during the VNF snapshot creation. See note.
vnfcSnapshotInfoIds	IdentifierLocal	0..N	Identifiers of information held by the VNFM about specific VNFC snapshots part of the VNF snapshot and contained in the VNF snapshot package. This identifier is allocated by the VNFM during the VNF snapshot creation. See note.
isFullSnapshot	Boolean	1	Value is TRUE in case of a "full" VNF snapshot package, i.e. containing all snapshotted VNFC instances; otherwise the value is FALSE.
vnfdInfo	VnfdInfo	0..1	VNFD of the snapshotted VNF instance that is contained in the stored VNF snapshot package. See note.
vnfsr	VnfSnapshotRecord	0..1	VNF snapshot record with the information as present in the representation of the "Individual VNF snapshot" resource.
vnfcSnapshotImages	VnfcSnapshotImageInfo	0..N	Information about VNF snapshot artifacts that are VNFC snapshot images. Every local and external snapshot image shall be included. No other artifacts shall be included. See note.
additionalArtifacts	SnapshotPkgArtifactInfo	0..N	Information about VNF snapshot artifacts that are not VNFC snapshot images. See note.

Attribute name	Data type	Cardinality	Description
state	Enum (inlined)	1	State of the VNF snapshot package. Permitted values: • CREATED: the VNF snapshot package information has been created. • BUILDING: the VNF snapshot package is being built. • UPLOADING: the VNF snapshot package is being uploaded. • EXTRACTING: the VNF snapshot package's content is being extracted. • AVAILABLE: the VNF snapshot package is available (i.e. build or upload is completed). • PROCESSING: the VNF snapshot is being processed. • ERROR: failure during the VNF snapshot package building, uploading or processing. • ERROR_EXTRACTING: failure during the VNF snapshot package extraction task.
isCancelPending	Boolean	1	Indicates if an ongoing operation with the content of the VNF snapshot package is being cancelled. If the value of the "state" attribute is "BUILDING", "UPLOADING", "PROCESSING" or "EXTRACTING" and the operation is being cancelled, this attribute shall be set to true. Otherwise, it shall be set to false.
failureDetails	Structure (inlined)	0..1	Failure details associated to current error state of the VNF snapshot package state. If "state" is "ERROR" or "ERROR_EXTRACTING", this attribute shall be present unless it has been requested to be excluded via an attribute selector.
>errorType	Enum (inlined)	1	Type of error, when the failure happened (building, upload, processing, extracting). Permitted values: • BUILD_ERROR • UPLOAD_ERROR • PROCESS_ERROR • CANCELLED • EXTRACTION_ERROR
>details	ProblemDetails	1	Failure details containing error information (such as failed uploading or processing operation, affected artifact, reason for cancellation, etc.). See clause 6.3 of ETSI GS NFV-SOL 013 [8] for the details of "ProblemDetails" structure.
userDefinedData	KeyValuePairs	0..1	User defined data for the VNF snapshot package to be built/uploaded.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>packageContent	Link	1	Link to the "VNF snapshot package content" resource.
NOTE: The attribute shall not be present before the VNF snapshot package content has been uploaded or built. Otherwise, this attribute shall be present unless it has been requested to be excluded per attribute selector.			

12.5.3 Referenced structured data types

12.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but are not resource representations.

12.5.3.2 Type: VnfcSnapshotImageInfo

This type represents an artifact contained in a VNF snapshot package which represents a snapshot image. It shall comply with the provisions defined in table 12.5.3.2-1.

Table 12.5.3.2-1: Definition of the VnfcSnapshotImageInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierLocal	1	Identifier of the VNFC snapshot image. When building the VNF snapshot package, the NFVO shall set the value of this attribute as follows: • for an image artifact corresponding to a compute snapshot resource, the value is copied from the "id" attribute of the "VnfcSnapshotInfo" • for an image artifact corresponding to a storage snapshot resource, the value is copied from the "storageResourceId" attribute in the "VnfcSnapshotInfo" of the corresponding storage snapshot resource. When onboarding an existing VNF snapshot package, the NFVO shall set the value of this attribute as provided in the manifest file in the VNF snapshot package (refer to ETSI GS NFV-SOL 010 [i.14]).
name	String	1	Name of the VNFC snapshot image.
checksum	Checksum	1	Checksum of the snapshot image file. Hash algorithms applicable to VNF snapshot package artifacts are defined in ETSI GS NFV-SOL 010 [i.14].
isEncrypted	Boolean	1	Reflects whether the artifact is encrypted (true) or not (false).
vnfcInstanceId	IdentifierInVnf	1	Identifier of the snapshotted VNFC instance that this snapshot image belongs to.

Attribute name	Data type	Cardinality	Description
containerFormat	Enum (inlined)	1	Container format indicates whether the snapshot image is in a file format that also contains metadata about the actual snapshot. Permitted values: • AKI: a kernel image format • AMI: a machine image format • ARI: a ramdisk image format • BARE: the image does not have a container or metadata envelope • DOCKER: docker container format • OVA: OVF package in a tarfile • OVF: OVF container format See note 1.
diskFormat	Enum (inlined)	1	Disk format of a snapshot image is the format of the underlying disk image. Permitted values: • AKI: a kernel image format • AMI: a machine image format • ARI: a ramdisk image format • ISO: an archive format for the data contents of an optical disc, such as CD-ROM • QCOW2: a common disk image format, which can expand dynamically and supports copy on write • RAW: an unstructured disk image format • VDI: a common disk image format • VHD: a common disk image format • VHDX: enhanced version of VHD format • VMDK: a common disk image format See note 2.
createdAt	DateTime	1	Timestamp indicating when the VNFC snapshot image was created.
minDisk	UnsignedInt	1	The minimal disk for this VNFC snapshot image (in bytes).
minRam	UnsignedInt	1	The minimal RAM for this VNFC snapshot image (in bytes).
size	UnsignedInt	1	Size of this VNFC snapshot image (in bytes).
userMetadata	KeyValuePairs	0..1	User-defined metadata.
imagePath	String	0..1	Path which identifies the image artifact and also allows to access a copy of the image artifact. For an image artifact contained as a file in the VNF snapshot package, this attribute shall be present, and the value of this attribute shall start with the name of the first segment in the path in the package, i.e. it shall not be prefixed by path separator characters such as "." and "/". EXAMPLE: foo/bar/m%40ster.vhd For an external image artifact represented as a URI in the manifest file, this attribute shall be present if the artifact has been downloaded by the NFVO or the artifact has been processed after building the VNF snapshot package and shall be absent otherwise. If present, it shall contain the artifactPath under which the image artifact can be obtained using the "Individual artifact in a VNF snapshot package" resource defined in clause 12.4.5. It is the responsibility of the NFVO to synthesize this path in a manner that avoids any collision of the synthesized artifact path with the paths and names of artifacts included in the snapshot package.

Attribute name	Data type	Cardinality	Description
imageUri	Uri	0..1	URI of the image artifact as defined in the VNF snapshot package manifest. Shall be present if the image artifact is external to the snapshot package and shall be absent otherwise. EXAMPLE: https://example.com/m%40ster.vhd
NOTE 1: The list of permitted values was taken from "Container formats" in [i.5]. NOTE 2: The list of permitted values was adapted from "Disk formats" in [i.5].			

12.5.3.3 Type: SnapshotPkgArtifactInfo

This type represents an artifact other than a software image which is contained in a VNF snapshot package. It shall comply with the provisions defined in table 12.5.3.3-1.

Table 12.5.3.3-1: Definition of the SnapshotPkgArtifactInfo data type

Attribute name	Data type	Cardinality	Description
artifactPath	String	0..1	Path which identifies the artifact and also allows to access a copy of the artifact. For an artifact contained as a file in the VNF snapshot package, this attribute shall be present, and the value of this attribute shall start with the name of the first segment in the path in the package, i.e. it shall not be prefixed by path separator characters such as "." and "/". EXAMPLE: foo/bar/m%40ster.sh For an external artifact represented as a URI in the manifest file, this attribute shall be present if the artifact has been downloaded by the NFVO or the artifact has been processed after building the VNF snapshot package and shall be absent otherwise. If present, it shall contain the artifactPath under which the artifact can be obtained using the "Individual artifact in a VNF snapshot package" resource defined in clause 12.4.5. It is the responsibility of the NFVO to synthesize this path in a manner that avoids any collision of the synthesized artifact path with the paths and names of artifacts included in the snapshot package.
artifactUri	Uri	0..1	URI of the artifact as defined in the VNF snapshot package manifest. Shall be present if the artifact is external to the snapshot package and shall be absent otherwise. EXAMPLE: https://example.com/m%40ster.sh
checksum	Checksum	1	Checksum of the artifact file. Hash algorithms applicable to VNF snapshot package artifacts are defined in ETSI GS NFV-SOL 010 [i.14].
isEncrypted	Boolean	1	Reflects whether the artifact is encrypted (true) or not (false).
metadata	KeyValuePairs	0..1	The metadata of the artifact that are available in the VNF package, such as Content type, size, creation date, etc.

12.5.3.4 Type: VnfdInfo

This type represents the VNFD which is contained in a VNF snapshot package. It shall comply with the provisions defined in Table 12.5.3.4-1.

Table 12.5.3.4-1: Definition of the VnfdInfo data type

Attribute name	Data type	Cardinality	Description
vnfdId	Identifier	1	VNFD identifier of the snapshotted VNF instance.
vnfdPath	String	1	Path which allows to access a copy of the VNFD. The VNFD is implemented as a collection of one or more files, and the path refers to the ZIP archive file embedding these files. The VNF snapshot package format is defined in the ETSI GS NFV-SOL 010 [i.14]. The value of this attribute shall start with the name of the first segment of the path in the package, i.e. it shall not be prefixed by path separator characters such as "." and "/". EXAMPLE: foo/bar/m@ster
checksum	Checksum	1	Checksum of the VNFD archive file. Hash algorithms applicable to VNF snapshot package artifacts are defined in ETSI GS NFV-SOL 010 [i.14].
isEncrypted	Boolean	1	Reflects whether the VNFD archive file is encrypted (true) or not (false).

12.5.3.5 Type: VnfSnapshotRecord

This type represents the VNF snapshot record which is contained in a VNF snapshot package. It shall comply with the provisions defined in table 12.5.3.5-1.

Table 12.5.3.5-1: Definition of the VnfSnapshotRecord data type

Attribute name	Data type	Cardinality	Description
recordPath	String	1	Path which identifies the VNF snapshot record and allows to access a copy of the VNF snapshot record for the extraction. The value of this attribute shall start with the name of the first segment of the path in the package, i.e. it shall not be prefixed by path separator characters such as "." and "/". EXAMPLE: foo/bar/m@ster
checksum	Checksum	1	Checksum of the VNF snapshot record file. Hash algorithms applicable to VNF snapshot package artifacts are defined in ETSI GS NFV-SOL 010 [i.14].
isEncrypted	Boolean	1	Reflects whether the VNF snapshot record file is encrypted (true) or not (false).

12.5.4 Referenced simple data types and enumerations

No particular simple data types and enumerations are defined for this interface, in addition to those defined in clause 4.4.

12.6 VNF snapshot package state model and error handling

The VNF snapshot package state model and error handling is specified in clause 11.6 of ETSI GS NFV-SOL 005 [i.13].

Annex A (informative): Mapping operations to protocol elements

A.1 Overview

This annex provides the mapping between operations as defined in ETSI GS NFV-IFA 007 [1] and the corresponding resources and HTTP methods defined in the present document.

A.2 VNF Package Management interface

Table A.2-1

ETSI GS NFV-IFA 007 [1] operation	HTTP method	Resource	Direction
Query VNF Package Info	GET	/vnf_packages /onboarded_vnf_packages	VNFM → NFVO
	GET	/vnf_packages/{vnfPkgId} /onboarded_vnf_packages/{vnfdId}	VNFM → NFVO
	GET	/vnf_packages/{vnfPkgId}/vnfd /onboarded_vnf_packages/{vnfdId}/vnfd	VNFM → NFVO
	GET	/vnf_packages/{vnfPkgId}/manifest /onboarded_vnf_packages/{vnfdId}/manifest	VNFM → NFVO
Fetch VNF Package	GET	/vnf_packages/{vnfPkgId}/package_content /onboarded_vnf_packages/{vnfdId}/package_content	VNFM → NFVO
Fetch VNF Package Artifacts	GET	/vnf_packages/{vnfPkgId}/artifacts /onboarded_vnf_packages/{vnfdId}/artifacts	VNFM → NFVO
	GET	/vnf_packages/{vnfPkgId}/artifacts/{artifactPath} /onboarded_vnf_packages/{vnfdId}/artifacts/{artifactPath}	VNFM → NFVO
Subscribe	POST	/subscriptions	VNFM → NFVO
Query Subscription Information	GET	/subscriptions	VNFM → NFVO
	GET	/subscriptions/{subscriptionId}	VNFM → NFVO
Terminate subscription	DELETE	/subscriptions/{subscriptionId}	VNFM → NFVO
Notify	POST	(provided by API consumer)	NFVO → VNFM

A.3 VNF Lifecycle Operation Granting interface

Table A.3-1

ETSI GS NFV-IFA 007 [1] operation	HTTP method	Resource	Direction
Grant Lifecycle Operation	POST	/grants	VNFM → NFVO
	GET	/grants/{grantId}	VNFM → NFVO

A.4 Virtualised Resources Management interfaces in indirect mode

This group of interfaces is outside the scope of the present document.

A.5 Virtualised Resources Quota Available Notification interface

Table A.5-1

ETSI GS NFV-IFA 007 [1] operation	HTTP method	Resource	Direction
Subscribe	POST	/subscriptions	VNFM → NFVO
Query Subscription Information	GET	/subscriptions	NFVO → VNFM
	GET	/subscriptions/{subscriptionId}	NFVO → VNFM
Terminate subscription	DELETE	/subscriptions/{subscriptionId}	NFVO → VNFM
Notify	POST	(provided by API consumer)	NFVO → VNFM

A.6 VNF Lifecycle Management interface

Table A.6-1

ETSI GS NFV-IFA 007 [1] operation	HTTP method	Resource	Direction
Create VNF Identifier	POST	/vnf_instances	NFVO → VNFM
Instantiate VNF	POST	/vnf_instances/{vnfInstanceId}/instantiate	NFVO → VNFM
Scale VNF	POST	/vnf_instances/{vnfInstanceId}/scale	NFVO → VNFM
Scale VNF to Level	POST	/vnf_instances/{vnfInstanceId}/scale_to_level	NFVO → VNFM
Change VNF Flavour	POST	/vnf_instances/{vnfInstanceId}/change_flavour	NFVO → VNFM
Terminate VNF	POST	/vnf_instances/{vnfInstanceId}/terminate	NFVO → VNFM
Delete VNF Identifier	DELETE	/vnf_instances/{vnfInstanceId}	NFVO → VNFM
Query VNF	GET	/vnf_instances/{vnfInstanceId}	NFVO → VNFM
	GET	/vnf_instances	NFVO → VNFM
Heal VNF	POST	/vnf_instances/{vnfInstanceId}/heal	NFVO → VNFM
Operate VNF	POST	/vnf_instances/{vnfInstanceId}/operate	NFVO → VNFM
Change External VNF Connectivity	POST	/vnf_instances/{vnfInstanceId}/change_ext_conn	NFVO → VNFM
Change current VNF package	POST	/vnf_instances/{vnfInstanceId}/change_vnfpkg	NFVO → VNFM
Modify VNF Information	PATCH	/vnf_instances/{vnfInstanceId}	NFVO → VNFM
Get Operation Status	GET	/vnf_lcm_op_occs	NFVO → VNFM
	GET	/vnf_lcm_op_occs/{vnfLcmOpOccId}	NFVO → VNFM
Subscribe	POST	/subscriptions	NFVO → VNFM
Query Subscription Information	GET	/subscriptions	NFVO → VNFM
	GET	/subscriptions/{subscriptionId}	NFVO → VNFM
Terminate Subscription	DELETE	/subscriptions/{subscriptionId}	NFVO → VNFM
Notify	POST	(provided by API consumer)	VNFM → NFVO
Create Snapshot	POST	/vnf_snapshots	NFVO → VNFM
	POST	/vnf_instances/{vnfInstanceId}/create_snapshot	NFVO → VNFM
(not available)	PATCH	/vnf_snapshots/{vnfSnapshotInfoId}	NFVO → VNFM
Query Snapshot Information	GET	/vnf_snapshots	NFVO → VNFM
	GET	/vnf_snapshots/{vnfSnapshotInfoId}	NFVO → VNFM
Revert-to Snapshot	POST	/vnf_instances/{vnfInstanceId}/revert_to_snapshot	NFVO → VNFM
Fetch VNF state snapshot	GET	/vnf_snapshots/{vnfSnapshotInfoId}/vnf_state_snapshot	NFVO → VNFM
Delete Snapshot Information	DELETE	/vnf_snapshots/{vnfSnapshotInfoId}	NFVO → VNFM

A.7 VNF Performance Management interface

Table A.7-1

ETSI GS NFV-IFA 007 [1] operation	HTTP method	Resource	Direction
Create PM Job	POST	/pm_jobs	NFVO → VNFM
Delete PM Job	DELETE	/pm_jobs/{pmJobId}	NFVO → VNFM
Query PM Job	GET	/pm_jobs	NFVO → VNFM
	GET	/pm_jobs/{pmJobId}	NFVO → VNFM
Create Threshold	POST	/thresholds	NFVO → VNFM
Delete Threshold	DELETE	/thresholds/{thresholdId}	NFVO → VNFM
Query Threshold	GET	/thresholds	NFVO → VNFM
	GET	/thresholds/{thresholdId}	NFVO → VNFM
Subscribe	n/a	see note	n/a
Query Subscription Information	n/a	see note	n/a
	n/a	see note	n/a
Terminate Subscription	n/a	see note	n/a
Notify	POST	(provided by API consumer)	VNFM → NFVO
NOTE: In the VNF Performance Management interface, support for subscriptions has been dropped in version 2.7.1 of the present document in favour of controlling the delivery of notifications directly by the "Thresholds" and "PM jobs" resources.			

A.8 VNF Fault Management interface

Table A.8-1

ETSI GS NFV-IFA 007 [1] operation	HTTP method	Resource	Direction
Get Alarm List	GET	/alarms	NFVO → VNFM
Acknowledge Alarm	PATCH	/alarms/{alarmId}	NFVO → VNFM
Subscribe	POST	/subscriptions	NFVO → VNFM
Query Subscription Information	GET	/subscriptions	NFVO → VNFM
	GET	/subscriptions/{subscriptionId}	NFVO → VNFM
Terminate Subscription	DELETE	/subscriptions/{subscriptionId}	NFVO → VNFM
Notify	POST	(provided by API consumer)	VNFM → NFVO

A.9 VNF Indicator interface

Table A.9-1

ETSI GS NFV-IFA 007 [1] operation	HTTP method	Resource	Direction
Get Indicator Value	GET	/indicators	NFVO → VNFM
	GET	/indicators/{vnfInstanceId}	NFVO → VNFM
	GET	/indicators/{vnfInstanceId}/{indicatorId}	NFVO → VNFM
Subscribe	POST	/subscriptions	NFVO → VNFM
Query Subscription Information	GET	/subscriptions	NFVO → VNFM
	GET	/subscriptions/{subscriptionId}	NFVO → VNFM
Terminate Subscription	DELETE	/subscriptions/{subscriptionId}	NFVO → VNFM
Notify	POST	(provided by API consumer)	VNFM → NFVO

A.10 VNF snapshot package management interface

Table A.10-1

ETSI GS NFV-IFA 007 [1] operation	HTTP method	Resource	Direction
Fetch VNF Snapshot Package	GET	/vnf_snapshot_packages/{vnfSnapshotPkgId}/package_content	VNFM → NFVO
Fetch VNF Snapshot Package Artifacts	GET	/vnf_snapshot_packages/{vnfSnapshotPkgId}/artifacts/{artifactPath}	VNFM → NFVO
Query VNF Snapshot Package Information	GET	/vnf_snapshot_packages	VNFM → NFVO
	GET	/vnf_snapshot_packages/{vnfSnapshotPkgId}	VNFM → NFVO

Annex B (informative): Explanations

B.1 Introduction

This annex provides explanations of certain concepts introduced in the present document.

In clause B.2, the underlying concepts of scaling a VNF instance are explained.

In clause B.3, examples of VNF connectivity patterns, and change of VNF external connectivity, are provided.

B.2 Scaling of a VNF instance

A VNF instance can be scaled in the following ways:

- scale out: adding additional VNFC instances to the VNF to increase capacity
- scale in: removing VNFC instances from the VNF, in order to release unused capacity

This mechanism is called "horizontal scaling".

NOTE: Besides that, there is also "vertical scaling" which is not supported in the present document, and which includes scale up (adding further resources to existing VNFC instances) and scale down (removing resources from existing VNFC instances).

Potentially, different *aspects* of a VNF can be scaled independently. For example, a VNF could be designed to provide static capacity such as database nodes and dynamic capacity such as query processing nodes. Such a VNF might be scaled with reference to two separate aspects: the 'static capacity' aspect can be scaled by adding VNFCs from VNF Deployment Units (VDUs) defining database nodes, and the 'dynamic capacity' aspect can be scaled by adding VNFCs from VDUs defining query processing nodes. In complex VNF designs, scaling a VNF often requires adding/removing a number of related VNFC instances of several different types, possibly based on multiple VDUs. For example, in a high availability configuration, it might be required to add in each scaling step a pair of VNFC instances, one in active and one in standby configuration. The scaling aspects valid for a particular VNF are declared in the VNFD.

Each scaling aspect can only be scaled in discrete steps, the so-called "*scaling steps*". Each scaling step corresponds to adding or removing an *increment* (set of VNFCs based on one or more VDUs, and the related virtualised storage/virtualised network resources) to or from the VNF instance, and (re)configuring the virtualised resources. Per increment, the VNFM will figure out the necessary set of VNFCs and the related set of resources based on VNF-specific rules, for instance using the lifecycle management script associated to the Scale VNF or Scale VNF to Level event.

When scaling a VNF for a particular aspect, the number of scaling steps to apply to that aspect can be provided as a parameter. A scaling step is the smallest unit by which a particular aspect of a VNF can be scaled, and is mapped by the VNFM to the addition (or removal) of a certain set of resources. For each scaling aspect, the minimum scale level is assumed as zero, and the maximum scale level is defined in the VNFD. The maximum scale level corresponds to the maximum number of scaling steps that can be performed for this aspect, starting from the minimum scale level (i.e. zero). The maximum scale level represents the maximum configuration of that aspect of the VNF in a given deployment flavour. The minimum scale level represents the minimum configuration of that aspect of the VNF in a given deployment flavour. It usually corresponds to some deployed resources, but it is also possible to define in the VNFD that certain VDUs may not always have a corresponding VNFC instance, i.e. for certain aspects the minimum configuration may indeed be empty.

At each point in time between the completed VNF instantiation and the VNF termination, the current "size" of a particular scaling aspect of the VNF can be expressed by the current scale level with reference to that aspect. When the VNF is instantiated, the current scale level is initialized with values that are defined as part of the instantiation level in the VNFD for the associated aspect. Figure B.2-1 illustrates the concepts described above.

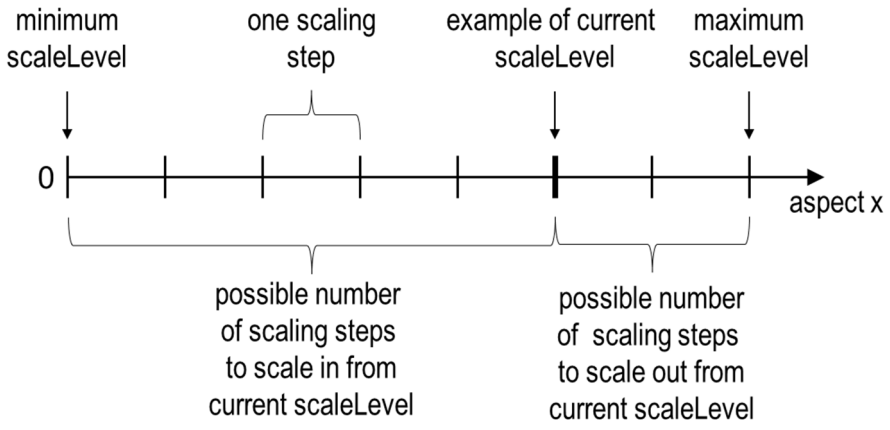

Figure B.2-1: Illustrating the concepts of scale level and scaling steps for a particular scaling aspect

As indicated above, a VNF can have one or more scaling aspects. Each individual aspect has a current scale level. All pairs of (aspect, scaleLevel) together are called the *scale status* of the VNF instance and can be obtained from the "scaleStatus" attribute of the VnfInstance structure which is returned when reading the "Individual VNF instance" resource or when querying the "VNF instances" resource. Example 1 illustrates a possible scale status.

EXAMPLE 1:

```
"scaleStatus": [
    {"aspectId": "processing", "scaleLevel": "2"},
    {"aspectId": "database", "scaleLevel": "3"}
]
```

When requesting scaling of a VNF instance, there are two methods: Scale VNF (see clause 5.4.5) and Scale VNF to Level (see clause 5.4.6). When using "Scale VNF", the scaling request defines how many increments (scaling steps) are requested to be added to or removed from the current "size" (scale level) *for a single aspect*. Depending on the VNF capabilities, single-step scaling or multiple-step scaling can be supported in a single scale request. When using "Scale VNF to Level", the scale request defines a target size of the VNF instance by defining the requested target size *for all aspects at once*, independent from the current scale status (current size) of the VNF instance. The target size can be expressed by referencing pre-defined sizes (called *instantiation levels*) declared in the VNFD, or by explicitly providing the target scale level for each scaling aspect, as illustrated in example 2.

EXAMPLE 2:

```
"scaleInfo": [
    {"aspectId": "processing", "scaleLevel": "4"},
    {"aspectId": "database", "scaleLevel": "2"}
]
```

These combinations allow four sub-modes of scaling:

- Scale VNF with a single step.
- Scale VNF with multiple steps.
- Scale VNF to Level based on pre-defined sizes (instantiation levels) only.
- Scale VNF to Level with arbitrary sizes.

B.3 Examples of VNF connectivity patterns

B.3.1 Introduction

Clause B.3.2 illustrates examples of possible connectivity patterns for a VNF. The purpose is to illustrate the relationship among the different information elements specified in clause 8.5 that are used to describe the connectivity of and within a VNF instance.

NOTE: The information related to connectivity as shown in clause B.3.2 is to be understood in the context of the present document, i.e. availability of certain information on the Or-Vnfm reference point follows the conditions that are detailed in the respective attribute descriptions and notes in the present document.

Clause B.3.3 illustrates the use of the "Change external VNF connectivity task" resource to re-connect external CPs of a VNF instance to a different external VL.

B.3.2 Example of a VNF instance with two different types of external connection points

The present example shows a regular connectivity pattern of a VNF where the two external CPs of the VNF use different connectivity patterns. Figure B.3.2-1 illustrates the example, from which it is highlighted the following:

- An external CP of the VNF instance (see VnfExtCp #1) that maps to an internal CP, i.e. a CP of a specific VNFC.
- An external CP of the VNF instance (see VnfExtCp #2) that refers to a link port of an internal VL of the VNF (see VnfLinkPort #2.2).
- An internal VL of the VNF instance (see VnfVirtualLink #1) that is only used for connectivity of VNFCs within the VNF.
- An internal VL of the VNF instance (see VnfVirtualLink #2) that is used as provider of a link port for connectivity of external CPs of the VNF.
- Link ports of internal VLs of the VNF instance (see VnfLinkPort #1.1 to #1.3 and VnfLinkPort #2.1) that are optionally exposed on Or-Vnfm reference point.
- Internal CPs, i.e. CPs of specific VNFCs (see grey VNFC CPs) that are optionally exposed on the Or-Vnfm reference point.

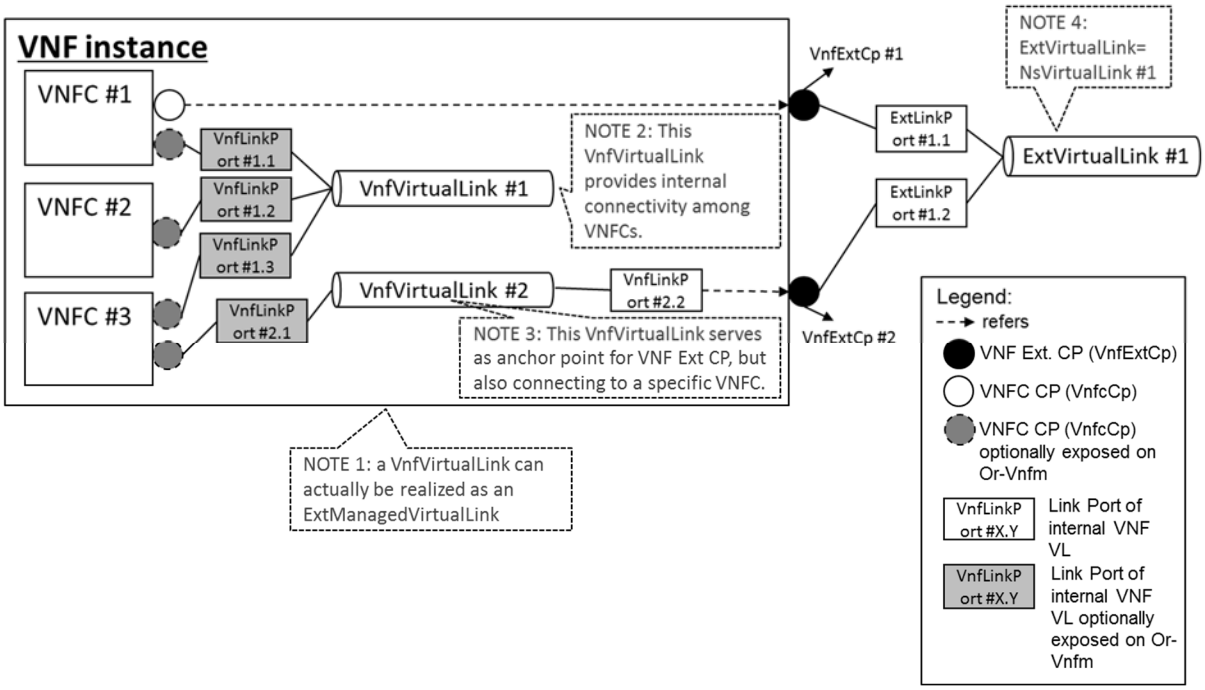

Figure B.3.2-1: Example of a VNF instance with two different types of external connection points

B.3.3 Example of changing VNF connectivity

This example illustrates changing the external connectivity of a VNF instance using the "Change external VNF connectivity task" resource (clause 5.4.11). The scenario depicted disconnects from a "source" external VL all those external CP instances that were created based on a particular CPD, and connects them to a "target" external VL.

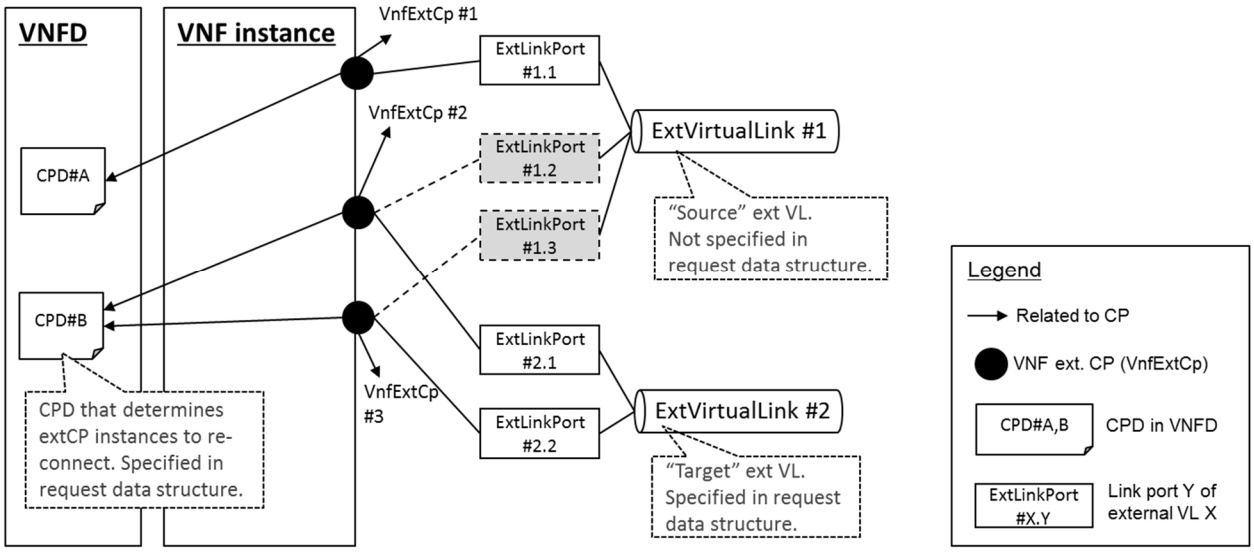

Figure B.3.3-1: Illustration of disconnecting external CPs from one external VL and connecting them to another external VL

Annex C (normative): VimConnectionInfo registry

C.1 Purpose

This annex defines the basic structure of the entries of a registry for VimConnectionInfo parameters, the structure of the identifiers, and the registration template. The registry contains a reference to the present document to indicate where the structure of the VimConnectionInfo data type is defined.

C.2 Registry content

The primary elements of the registry are:

Registered identifier

- vimType: Identifier of a set of VimConnectionInfo parameters (mandatory)

NOTE 1: The registration authority is required to ensure global uniqueness of registered identifiers.

Registered interface information for a particular vimType

- interfaceInfo: Interface information as a list of key names with data type, permissible values and description (mandatory)
- accessInfo: Access information as a list of key names with data type, permissible values and description (mandatory)
- extra: Additional specific information as a list of key names with data type, permissible values and description (optional)

Registrant information

- Registrant Name: Name of the company or organization registering the vimType (mandatory)
- Previous Registrant Name(s): Name or names of the company or organization to whom the registered identifier has belonged previously, e.g. due to buyout, merger, acquisition (optional)

NOTE 2: It is assumed that the registration authority will manage further information related to the identity of the registrant (e.g. contact information).

Additional information

- Solution Name: Name of the VIM for which the VimConnectionInfo parameter set is being registered (e.g. "OpenStack Release xyz with Keystone") (mandatory)
- Description: General description of the VIM for which the VimConnectionInfo parameter set is being registered (e.g. "ETSI-registered VIM Connection Info to enable the use of Openstack rel xyz with ETSI GS NFV-SOL 003. Interface is using keystone as the gateway. Valid for releases starting from xyz"). (mandatory)
- Specification URI: Publically reachable URI of the specification that defines further details of the particular VIM Connection Info registered. Needs to be long-lived. (recommended)
- Registration Date: Date of the registration (mandatory)

C.3 Structure of the vimType identifier

A registered vimType identifier shall comply with the following syntax:

- It shall have the following structure:
  `<registrant> "." <vimName> ["." <version>].`
- <registrant> and <vimName> shall be strings that contain only uppercase letters, digits, "_" and "-".
- <registrant> shall be a concise string that represents the registrant (e.g. the company or organization name) chosen at registration time.
- The <registrant> value of "PRIVATE" is reserved for private use by implementations and shall not be used in registered identifiers.
- <vimName> shall be a string that represents the VIM Type.
- <version> shall be a string with the following structure:
  `"V_" <x> [ "_" <y>].`
- Providing a <version> is optional. In the <version> string, "x" and "y" represent a sequence of digits that denote the major (x) and minor (y) part of the version number of the VIM. Providing <y> is optional.
- If there are no changes of the interface in subsequent versions of the VIM, i.e. the registration of the previous version can still be used with the new versions, a registration of these subsequent versions is not required.

C.4 Initial registration

C.4.1 Instructions for data structure definition

Data structure definitions shall be submitted using JSON schema according to [i.6], separately for the three objects "interfaceInfo" (mandatory), "accessInfo" (mandatory) and "extra" (optional). The "description" keyword in the JSON schema shall be used to appropriately document each attribute in the data structure. The JSON schema shall include information about allowed values where applicable, formulated as JSON schema constraints (such as "minimumInclusive") or documented in text form using the "description" keyword.

In the registration template, items that are mandatory to be provided are marked with [M]; optional items are marked with [O].

C.4.2 Template

1 Solution information

Solution Name [M]	<Name of the VIM for which the VimConnectionInfo parameter set is being registered.> *EXAMPLE: "OpenStack Release xyz with Keystone"*
Description [M]	General description of the VIM for which the VimConnectionInfo parameter set is being registered *EXAMPLE: "ETSI-registered VIM Connection Info to enable the use of Openstack rel xyz with ETSI GS NFV-SOL 003. Interface is using keystone as the gateway. Valid for releases starting from xyz".*
Specification URI [O]	<Publically reachable, long-lived URI of a specification that defines further details of the particular VIM Connection Info registered. Optional, but recommended.>

2 Registration information

Registrant name [M]	<Name of the legal entity requesting registration>
Registrant address [M]	<Address of the legal entity requesting registration>
Registrant contact [M]	<Name and email address of the contact person or the function of the legal entity requesting registration>
Registration date [M]	<The date when the registration request has been sent>

3 Requested vimType identifier

Registrant [M]		VIM Name [M]		Version [O]
	.		.	

4 JSON schema definition of "interfaceInfo"

Purpose: Provides information about the interface or interfaces to the VIM, such as the URI of an interface endpoint to communicate with the VIM.

interfaceInfo [M]

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "interfaceInfo",
    "type": "object",
    "properties": {
       <include list of properties here >
    },
    "required": [<define properties that are required>]
}
```

5 JSON schema definition of "accessInfo"

Purpose: Provides authentication credentials for accessing the VIM, and other access-related information such as tenants or infrastructure resource groups.

accessInfo [M]

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "accessInfo",
    "type": "object",
    "properties": {
       <include list of properties here >
    },
    "required": [<define properties that are required>]
}
```

6 JSON schema definition of "extra"

Purpose: Provides optional additional VIM type specific information.

extra [O]

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "extra",
    "type": "object",
    "properties": {
       <include list of properties here >
    },
    "required": [<define properties that are required>]
}
```

C.5 Registration update

Only limited parts of the registration information are allowed to be updated:

- Registrant name (in case of mergers, etc.). In this case, the registrant information will be updated and the previous registrant information will be preserved in a special "previous registrants" section.
- Registrant contact data (in case of change of contact person).
- Specification URI (in case of update of URI).

C.6 Initial registry content

C.6.1 Registration for ETSINFV.OPENSTACK_KEYSTONE.V_2

1 Solution information

Solution Name [M]	OpenStack with Keystone V2
Description [M]	ETSI-registered VIM Connection Info defining the interface and access parameters to use an OpenStack-based VIM with Keystone V2, to be signalled via the APIs specified in ETSI GS NFV-SOL 003. Keystone is used for access control to the VIM interfaces.
Specification URI [O]	https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/003/

2 Registration information

Registrant name [M]	ETSI ISG NFV

3 Requested vimType identifier

Registrant [M]		VIM Name [M]		Version [O]
ETSINFV	.	OPENSTACK_KEYSTONE	.	V_2

4 JSON schema definition of "interfaceInfo"

Purpose: Provides information about the interface or interfaces to the VIM, such as the URI of an interface endpoint to communicate with the VIM.

interfaceInfo [M]

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "interfaceInfo",
    "additionalProperties": false,
    "required": [
        "endpoint"
    ],
    "type": "object",
    "properties": {
        "endpoint": {
            "type": "string",
            "format": "url",
            "description": "The url representing the interface endpoint."
        },
        "trustedCertificates": {
            "items": {
                "type": "string",
                "format": "byte"
            },
            "type": "array",
            "description": "A collection of base64 encoded certificates to be trusted in relation
to the endpoint."
```

```
        },
        "skipCertificateHostnameCheck": {
            "default": false,
            "type": "boolean",
            "description": "Certificate hostname check for the endpoint can be skipped by setting
this field to true."
        },
        "skipCertificateVerification": {
            "default": false,
            "type": "boolean",
            "description": "Certificate verification for the endpoint can be skipped by setting
this field to true."
        }
    }
}
```

5 JSON schema definition of "accessInfo"

Purpose: Provides authentication credentials for accessing the VIM, and other access-related information such as tenants or infrastructure resource groups.

accessInfo [M]

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "accessInfo",
    "additionalProperties": false,
    "required": [
        "username",
        "password",
        "region",
        "tenant"
    ],
    "type": "object",
    "properties": {
        "username": {
            "type": "string",
            "description": "The username to use for access."
        },
        "region": {
            "type": "string",
            "description": "The OpenStack region to use for the VIM connection."
        },
        "password": {
            "writeOnly": true,
            "type": "string",
            "format": "password",
            "description": "The password to use for access. Required for input, not returned on
output."
        },
        "tenant": {
            "type": "string",
            "description": "The OpenStack tenant to use for the VIM connection."
        }
    }
}
```

6 JSON schema definition of "extra"

Purpose: Provides optional additional VIM type specific information.

extra [O]

```
not specified
```

C.6.2 Registration for ETSINFV.OPENSTACK_KEYSTONE.V_3

1 Solution information

Solution Name [M]	OpenStack with Keystone V3
Description [M]	ETSI-registered VIM Connection Info defining the interface and access parameters to use an OpenStack-based VIM with Keystone V2, to be signalled via the APIs specified in ETSI GS NFV-SOL 003. Keystone is used for access control to the VIM interfaces.
Specification URI [O]	https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/003/

2 Registration information

Registrant name [M]	ETSI ISG NFV

3 Requested vimType identifier

Registrant [M]		VIM Name [M]		Version [O]
ETSINFV	.	OPENSTACK_KEYSTONE	.	V_3

4 JSON schema definition of "interfaceInfo"

Purpose: Provides information about the interface or interfaces to the VIM, such as the URI of an interface endpoint to communicate with the VIM.

interfaceInfo [M]

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "interfaceInfo",
    "additionalProperties": false,
    "required": [
        "endpoint"
    ],
    "type": "object",
    "properties": {
        "endpoint": {
            "type": "string",
            "format": "url",
            "description": "The url representing the interface endpoint."
        },
        "trustedCertificates": {
            "items": {
                "type": "string",
                "format": "byte"
            },
            "type": "array",
            "description": "A collection of base64 encoded certificates to be trusted in relation
to the endpoint."
        },
        "skipCertificateHostnameCheck": {
            "default": false,
            "type": "boolean",
            "description": "Certificate hostname check for the endpoint can be skipped by setting
this field to true."
        },
        "skipCertificateVerification": {
            "default": false,
            "type": "boolean",
            "description": "Certificate verification for the endpoint can be skipped by setting
this field to true."
        }
    }
}
```

5 JSON schema definition of "accessInfo"

Purpose: Provides authentication credentials for accessing the VIM, and other access-related information such as tenants or infrastructure resource groups.

accessInfo [M]

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "accessInfo",
    "additionalProperties": false,
    "required": [
         "username",
         "password",
         "region",
         "project",
        "projectDomain",
        "userDomain"
    ],
    "type": "object",
    "properties": {
         "username": {
             "type": "string",
             "description": "The username to use for access."
         },
         "userDomain": {
         "type": "string",
             "description": "The OpenStack user domain to use for the VIM connection."
        },
         "region": {
             "type": "string",
             "description": "The OpenStack region to use for the VIM connection."
         },
         "password": {
             "writeOnly": true,
             "type": "string",
             "format": "password",
             "description": "The password to use for access. Required for input, not returned on
output."
         },
         "project": {
             "type": "string",
             "description": "The OpenStack project to use for the VIM connection."
         },
         "projectDomain": {
             "type": "string",
             "description": "The OpenStack project domain to use for the VIM connection"
         }
    }
}
```

6 JSON schema definition of "extra"

Purpose: Provides optional additional VIM type specific information.

extra [O]

```
not specified
```

Annex D (informative): Complementary material for API utilization

To complement the definitions of each method, resource, and data type defined in the main body of the present document, the ETSI NFV ISG is providing supplementary description files, compliant to the OpenAPI Specification [**Error! Reference source not found.**], for the Or-Vnfm reference point. These supplementary description files, containing the OpenAPI specification for each API defined in the present document, are located at https://forge.etsi.org/rep/nfv/NFV-SOL003.

In case of discrepancies between the supplementary files and the related data structure definitions in the main body of the present document, the data structure definitions take precedence.

The OpenAPI representations referenced above:

1) use the MAJOR.MINOR.PATCH version fields to signal the version of the API as defined in the present document; and

2) use the "impl" version parameter (see clause 9.1.2 of ETSI GS NFV-SOL 013 [8]) to represent changes to the OpenAPI representation without changing the present document).

It is specified in clause 6 of ETSI GS NFV-SOL 015 [i.11] how the OpenAPI specification references the present document and signals the version information.

Annex E (informative): Void

Annex F (informative): History of features added to the present document

F.1 Overview

The present document has been first released as part of ETSI NFV Release 2 and went through multiple cycles of maintenance.

In ETSI NFV Release 3, features were added. The branching has occurred after version 2.8.1 of the present document.

This annex lists the features that were added on top of Release 2 in Release 3. To help implementers to determine which changes make up together a particular feature, these are documented below per feature.

F.2 Features added in Release 3

F.2.1 FEAT02: VNF Software modification

This feature addresses the initiation and the coordination of the software modification process related to VNFs. Goal is to minimize the impact of software modification on service availability.

Table F.2.1-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
5.4.11a	vnflcm	/vnf_instances/{vnfInstanceId}/change_vnfpkg	New resource
5.5.2.2.	vnflcm	VnfInstance._links.changeCurrentVnfPkg	New attribute
5.5.2.11a	vnflcm	Data type: ChangeCurrentVnfPkgRequest	New resource data type
5.5.2.13	vnflcm	VnfLcmOpOcc.operationParams: new structure "ChangeCurrentVnfPkgRequest"	Modified permitted attribute values
5.5.2.13	vnflcm	VnfLcmOpOcc.modificationsTriggeredByVnfPkgChange	New attribute
5.5.2.17	vnflcm	VnfLcmOperationOccurrenceNotification Trigger "Change of current VNF package"	New trigger condition
5.5.2.17	vnflcm	VnfLcmOperationOccurrenceNotification.modificationsTriggeredByVnfPkgChange	New attribute
5.5.3.3	vnflcm	ExtManagedVirtualLinkInfo.vnfdId	New attribute
4.4.1.15	multiple	ScaleInfo.vnfdId	New attribute
5.5.3.5	vnflcm	VnfcResourceInfo.vnfdId	New attribute
5.5.3.6	vnflcm	VnfVirtualLinkResourceInfo.vnfdId	New attribute
5.5.3.7	vnflcm	VirtualStorageResourceInfo.vnfdId	New attribute
5.5.3.11	vnflcm	MonitoringParameter.vnfdId	New attribute
5.5.3.13	vnflcm	AffectedVnfc.vnfdId	New attribute
5.5.3.14	vnflcm	AffectedVirtualLink.vnfdId	New attribute
5.5.3.15	vnflcm	AffectedVirtualStorage.vnfdId	New attribute
5.5.3.17	vnflcm	VnfExtCpInfo.vnfdId	New attribute
5.5.3.21	vnflcm	ModificationsTriggeredByVnfPkgChange.vimConnectionInfo	New attribute
5.5.4.7	vnflcm	LcmOperationType: added value CHANGE_VNFPKG	New enum value
8.4.7.3	vnfind	<<Callback URI>>(SupportedIndicatorsChangeNotification)	New notification
8.5.2.6	vnfind	SupportedIndicatorsChangeNotification	New notification
8.5.3.2	vnfind	VnfIndicatorNotificationsFilter.notificationTypes	New attribute
9.5.2.2	grant	GrantRequest.dstVnfdId	New attribute
9.5.3.2	grant	ResourceDefinition.vnfdId	New attribute
9.5.4.3	grant	GrantedLcmOperationType: added value CHANGE_VNFPKG	New enum value

F.2.2 FEAT04: Host reservation

The present enhancement proposes adding the capability to the NFV-MANO architectural framework to support the reservation of compute hosts.

Table F.2.2-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
9.5.2.3	grant	Grant.computeReservationId	Removed attribute
9.5.2.3	grant	Grant.networkReservationId	Removed attribute
9.5.2.3	grant	Grant.storageReservationId	Removed attribute
9.5.3.3	grant	GrantInfo.reservationId has now VIM scope and can also carry the identifier of a reserved host	Modified attribute semantics
9.1 and 9.5.3.3	grant	Support for the policies GRANT_RESERVE_SINGLE and GRANT_RESERVE_MULTI has been dropped. A new policy GRANT_RESERVE has been introduced instead	Other change

F.2.3 FEAT10: Multi-site connectivity services

This feature specifies management requirements, interfaces and information models to support connectivity for multi-site services.

Table F.2.3-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
4.4.1.12	multiple	ExtManagedVirtualLinkData.vnfLinkPort	New attribute
4.4.1.12	multiple	ExtManagedVirtualLinkData.extManagedMultisiteVirtualLinkId	New attribute
5.5.2.2	vnflcm	VnfInstance.extManagedVirtualLinkInfo	Changed attribute semantics
5.5.2.4	vnflcm	InstantiateVnfRequest.extManagedVirtualLinkData	Changed attribute semantics
5.5.3.3	vnflcm	ExtManagedVirtualLinkInfo.extManagedMultisiteVirtualLinkId	New attribute
9.5.2.3	grant	Grant.vimConnectionInfo	Changed attribute semantics
9.5.2.3	grant	Grant.extManagedVirtualLinks	Changed attribute semantics
9.5.3.3	grant	GrantInfo.reservationId	Changed attribute semantics

F.2.4 FEAT15: VNF snapshotting

VNF Snapshot is a replication of a VNF instance at a specific point in time with a corresponding VNF Snapshot Package which is collection of files representing a VNF Snapshot. The feature implementation enables operations on and management of VNF Snapshots and their corresponding packages.

Table F.2.4-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
5.4.21	vnflcm	/vnf_instances/{vnfInstanceId}/create_snapshot	New resource
5.4.22	vnflcm	/vnf_instances/{vnfInstanceId}/revert_to_snapshot	New resource
5.4.23	vnflcm	/vnf_snapshots	New resource
5.4.24	vnflcm	/vnf_snapshots/{vnfSnapshotInfoId}	New resource
5.4.25	vnflcm	/vnf_snapshots/{vnfSnapshotInfoId}/vnf_state_snapshot	New resource
5.5.2.2	vnflcm	VnfInstance._links.createSnapshot	New attribute
5.5.2.2	vnflcm	VnfInstance._links.revertToSnapshot	New attribute
5.5.2.13	vnflcm	VnfLcmOpOcc.operationParams: new structures "CreateVnfSnapshotRequest" and "RevertToVnfSnapshotRequest"	Modified permitted attribute values
5.5.2.13	vnflcm	VnfLcmOpOcc.vnfSnapshotInfoId	New attribute
5.5.2.13	vnflcm	VnfLcmOpOcc._links.vnfSnapshot	New attribute
5.5.2.20	vnflcm	CreateVnfSnapshotInfoRequest	New resource data type
5.5.2.21	vnflcm	CreateVnfSnapshotRequest	New resource data type
5.5.2.22	vnflcm	VnfSnapshotInfo	New resource data type
5.5.2.23	vnflcm	VnfSnapshot	New resource data type
5.5.2.24	vnflcm	VnfSnapshotInfoModificationRequest	New resource data type
5.5.2.25	vnflcm	VnfSnapshotInfoModifications	New resource data type
5.5.2.26	vnflcm	RevertToVnfSnapshotRequest	New resource data type
5.5.4.7	vnflcm	LcmOperationType: added values CREATE_SNAPSHOT and REVERT_TO_SNAPSHOT	Additional enumeration values
9.5.2.3	grant	Grant.vimAssets.snapshotResources	New attribute
9.5.3.2	grant	ResourceDefinition.snapshotResDef	New attribute

Clause	Interface	Content of the change	Type of change
9.5.4.3	grant	GrantedLcmOperationType: added values CREATE_SNAPSHOT and REVERT_TO_SNAPSHOT	New enum value
12	vnfsnapshotpkgm	VNF snapshot package management API	New API

F.2.5 Additional new functionality outside the "NFV features" scheme

F.2.5.1 Trunking support

The parameters that provide external CP data have been modified to support trunking and to allow easier modification.

Table F.2.5.1-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
4.4.1.10	multiple	VnfExtCpData.cpConfig	Other change: turn this attribute into a map, Modified attribute semantics
4.4.1.10a	multiple	VnfExtCpConfig.parentCpConfigId	New attribute
4.4.1.10a	multiple	VnfExtCpConfig.id	Removed attribute
4.4.1.10a	multiple	VnfExtCpConfig.cpInstanceId	Removed attribute
4.4.1.10c	multiple	IpOverEthernetAddressData.segmentationId	New attribute
4.4.1.10c	multiple	IpOverEthernetAddressData.segmentationType	New attribute
4.4.1.14	multiple	ExtLinkPortData.trunkResourceId	New attribute
5.5.3.2	vnflcm	ExtVirtualLinkInfo.currentVnfExtCpData	New attribute
5.5.3.5	vnflcm	VnfcResourceInfo.vnfcCpInfo.parentCpId	New attribute
5.5.3.8	vnflcm	VnfLinkPortInfo.trunkResourceId	New attribute
5.5.3.9	vnflcm	ExtLinkPortInfo.trunkResourceId	New attribute
5.5.3.10	vnflcm	IpOverEthernetAddressInfo.segmentationId	New attribute
5.5.3.17	vnflcm	VnfExtCpInfo.cpConfigId	New attribute

F.2.5.2 Refactored patching scheme for VIM connection information

The scheme to patch VIM connection information has been refactored to be fully compliant with JSON Merge Patch.

Table F.2.5.2-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
4.4.1.6	multiple	VimConnectionInfo.id	Removed attribute
5.5.2.2	vnflcm	VnfInstance.vimConnectionInfo	Other change: turn this attribute into a map
5.5.2.4	vnflcm	InstantiateVnfRequest.vimConnectionInfo	Other change: turn this attribute into a map
5.5.2.7	vnflcm	ChangeVnfFlavourRequest.vimConnectionInfo	Other change: turn this attribute into a map
5.5.2.11	vnflcm	ChangeExtVnfConnectivityRequest.vimConnectionInfo	Other change: turn this attribute into a map
5.5.2.12	vnflcm	VnfInfoModificationRequest.vimConnectionInfo	Other change: turn this attribute into a map, Modified attribute semantics
5.5.2.12	vnflcm	VnfInfoModificationRequest.vimConnectionInfoDeleteIds	Removed attribute
5.5.2.12a	vnflcm	VnfInfoModifications.vimConnectionInfo	Other change: turn this attribute into a map, Modified attribute semantics
5.5.2.12a	vnflcm	VnfInfoModifications.vimConnectionInfoDeleteIds	Removed attribute
9.5.2.3	grant	Grant.vimConnectionInfo	Renamed attribute, Other change: turn this attribute into a map, Modified attribute semantics

F.2.5.3 Verbosity of VNF LCM operation occurrence notifications

This change enables to control the verbosity of VNF LCM operation occurrence notifications.

Table F.2.5.3-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
5.5.2.15	vnflcm	LccnSubscriptionRequest.verbosity	New attribute
5.5.2.16	vnflcm	LccnSubscription.verbosity	New attribute
5.5.2.17	vnflcm	VnfLcmOperationOccurrenceNotification.verbosity	New attribute
5.5.2.17	vnflcm	VnfLcmOperationOccurrenceNotification	Other change: Allow to omit certain attributes depending on the value of the "verbosity" attribute

F.2.5.4 LCM coordination

LCM coordination allows an ongoing LCM operation occurrence to trigger a related management operation by the EM or VNF, to wait for its result, and to coordinate that management operation with the LCM operation occurrence. This functionality is used e.g. by FEAT02 and FEAT15 if such coordination is needed.

Table F.2.5.4-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
5.5.2.13	vnflcm	VnfLcmOpOcc.lcmCoordinations	New attribute
5.5.2.13	vnflcm	VnfLcmOpOcc.rejectedLcmCoordinations	New attribute
5.5.2.13	vnflcm	VnfLcmOpOcc.warnings	New attribute

F.2.5.5 Support for virtual IP connection points

The VNF connectivity model has been updated to support virtual IP connection points (VIP CPs). Refer to clause A.4 in ETSI GS NFV-IFA 007 [1] for the supported use cases.

Table F.2.5.5-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
4.4.1.10a	multiple	VnfExtCpConfig.createExtLinkPort	New attribute
4.4.1.11	multiple	ExtVirtualLinkData.extLinkPorts	Modified attribute semantics (added support for ports related to a VIP CP)
5.5.2.2	vnflcm	VnfInstance.instantiatedVnfInfo.vipCpInfo	New attribute
5.5.2.13	vnflcm	VnfLcmOpOcc.affectedVipCps	New attribute
5.5.2.17	vnflcm	VnfLcmOperationOccurrenceNotification. affectedVipCps	New attribute
5.5.3.8	vnflcm	VnfLinkPortInfo.vipCpInstanceId	New attribute
5.5.3.9	vnflcm	ExtLinkPortInfo.secondaryCpInstanceId	New attribute
5.5.3.17	vnflcm	VnfExtCpInfo.associatedVipCpId	New attribute
9.5.3.2	grant	ResourceDefinition. secondaryResourceTemplateId	New attribute

F.2.6 FEAT03: NFVI software modification

This feature addresses the support for the coordination of the NFVI software modification process with the VNFs hosted on the NFVI in order to minimize impact on service availability. FEAT03 specifies the fault type "NFVI_OAM_VIRTUALISED_RESOURCE_STATE_CHANGE" which enables the NFVO to observe information on changes in the state of the virtualised resources due to upcoming NFVI operation and maintenance.

Table F.2.6-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
7.5.2.4	vnffm	Alarm.faultType	Modified permitted attribute values
7.5.2.4	vnffm	Alarm.probableCause	Modified permitted attribute values
7.5.2.4	vnffm	Alarm.rootCauseFaultyResource	Modified attribute semantics
7.5.2.4	vnffm	Alarm.eventTime	Modified attribute semantics
7.5.2.4	vnffm	Alarm.faultDetails	Modified attribute semantics

F.3 Features added in Release 4

F.3.1 FEAT17: Cloud-Native VNFs and Container Infrastructure management

The scope of the feature covers the following areas:

- NFV Architecture support for VNFs which follow "cloud-native" design principles.
- Enhance NFV-MANO capabilities to support container technologies based on ETSI GR NFV-IFA 029 [i.4].
- Enhance NFV-MANO capabilities for container management and orchestration.
- Enhance information model for containerized VNFs both using bare metal or nested virtualization technologies.

Table F.3.1-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
4.4.1.6	multiple	VimConnectionInfo.vimId: added the support of CISM, CIR and MCIOP repository	Modified attribute semantics
4.4.1.6	multiple	VimConnectionInfo.vimType: added the support of CISM, CIR and MCIOP repository	Modified attribute semantics
4.4.1.6	multiple	VimConnectionInfo.interfaceInfo: added the support of CISM, CIR and MCIOP repository	Modified attribute semantics
4.4.1.6	multiple	VimConnectionInfo.accessInfo: added the support of CISM, CIR and MCIOP repository	Modified attribute semantics
4.4.1.6	multiple	VimConnectionInfo.extra: added the support of CISM, CIR and MCIOP repository	Modified attribute semantics
4.4.1.7	multiple	ResourceHandle.vimConnectionId: extended to include the support of CISM connections	Modified attribute semantics
4.4.1.7	multiple	ResourceHandle.resourceId: extended to include the support of CISM	Modified attribute semantics
4.4.1.7	multiple	ResourceHandle.vimLevelResourceType: extended to include the support of CISM	Modified attribute semantics
4.4.1.7	multiple	ResourceHandle.vimLevelAdditionalResourceInfo	New attribute
4.4.1.7	multiple	ResourceHandle.containerNamespace	New attribute
4.4.1.10a	multiple	VnfExtCpConfig.linkPortId	Modified attribute semantics
4.4.1.10a	multiple	VnfExtCpConfig.netAttDefResourceId	New attribute
4.4.1.10a	multiple	VnfExtCpConfig.cpProtocolData	Modified attribute semantics
4.4.1.10b	multiple	CpProtocolData.layerProtocol: added value IP_FOR_VIRTUAL_CP	New enum value
4.4.1.10b	multiple	CpProtocolData.virtualCpAddress	New attribute
4.4.1.11	multiple	ExtVirtualLinkData.extLinkPorts	Modified attribute semantics
4.4.1.11	multiple	ExtVirtualLinkData.extNetAttDefResourceData	New attribute
4.4.1.12	multiple	ExtManagedVirtualLinkData.netAttDefResourceData	New attribute
4.4.1.12	multiple	ExtManagedVirtualLinkData.intCp	New attribute
4.4.1.12	multiple	ExtManagedVirtualLinkData.vnfLinkPort	Modified attribute semantics
4.4.2.2	multiple	IdentifierInVim: extended to include the support of CISM	Modified attribute semantics
5.5.2.2	vnflcm	VnfInstance.vimConnectionInfo: extended to include the support of CISM connections	Modified attribute semantics
5.5.2.2	vnflcm	VnfInstance.cirConnectionInfo	New attribute

Clause	Interface	Content of the change	Type of change
5.5.2.2	vnflcm	VnfInstance.mciopRepositoryInfo	New attribute
5.5.2.2	vnflcm	VnfInstance.instantiatedVnfInfo.virtualCpInfo	New attribute
5.5.2.2	vnflcm	VnfInstance.instantiatedVnfInfo.mcioInfo	New attribute
5.5.2.4	vnflcm	InstantiateVnfRequest.vimConnectionInfo: extended to include the support of CISM connections	Modified attribute semantics
5.5.2.7	vnflcm	ChangeVnfFlavourRequest.vimConnectionInfo: extended to include the support of CISM connections	Modified attribute semantics
5.5.2.11	vnflcm	ChangeExtVnfConnectivityRequest.vimConnectionInfo: extended to include the support of CISM connections	Modified attribute semantics
5.5.2.11	vnflcm	ChangeCurrentVnfPkgRequest.vimConnectionInfo: extended to include the support of CISM connections	Modified attribute semantics
5.5.2.17	vnflcm	VnfLcmOperationOccurrenceNotification.affectedVirtualCps	New attribute
5.5.3.2	vnflcm	ExtVirtualLinkInfo.extNetAttDefResource	New attribute
5.5.3.3	vnflcm	ExtManagedVirtualLinkInfo.vnfNetAttDefResource	New attribute
5.5.3.5	vnflcm	VnfcResourceInfo.computeResource: added the support of Compute MCIO	Modified attribute semantics
5.5.3.5	vnflcm	VnfcResourceInfo.storageResourceIds: added the support of Storage MCIO	Modified attribute semantics
5.5.3.5	vnflcm	VnfcResourceInfo.vnfcCpInfo.netAttDefResourceId	New attribute
5.5.3.6	vnflcm	VnfVirtualLinkResourceInfo.networkResource: added the support of Network MCIO	Modified attribute semantics
5.5.3.7	vnflcm	VirtualStorageResourceInfo.storageResource: added the support of Storage MCIO	Modified attribute semantics
5.5.3.9b	vnflcm	CpProtocolInfo.layerProtocol: added value IP_FOR_VIRTUAL_CP	New enum value
5.5.3.9b	vnflcm	CpProtocolInfo.virtualCpAddress	New attribute
5.5.3.17	vnflcm	VnfExtCpInfo.associatedVirtualCpId	New attribute
5.5.3.17	vnflcm	VnfExtCpInfo.netAttDefResourceId	New attribute
5.5.4.9	vnflcm	McioTypeName	New string type
9.5.2.3	grant	Grant.vimConnectionInfo: extended to include the support of CISM connections	Modified attribute semantics
9.5.2.3	grant	Grant.cirConnectionInfo	New attribute
9.5.2.3	grant	Grant.mciopRepositoryInfo	New attribute
9.5.2.3	grant	Grant.vimAssets: extended to support the information in the NFVI	Modified attribute semantics
9.5.2.3	grant	Grant.vimAssets.storageAssets	New attribute
9.5.3.2	grant	ResourceDefinition.type: added values OSCONTAINER and VIRTUALCP	New enum values
9.5.3.2	grant	ResourceDefinition.resourceTemplateId: added the support for the new type OSCONTAINER	Modified attribute cardinality and semantics
9.5.3.3	grant	GrantInfo.vimConnectionInfo: extended to include the use case of CISM connections	Modified attribute semantics
9.5.3.3	grant	GrantInfo.containerNamespace	New attribute
9.5.3.3	grant	GrantInfo.mcioConstraints	New attribute
9.5.3.6	grant	PlacementConstraint.scope: added values CIS_NODE and CONTAINER_NAMESPACE	New enum values
9.5.3.10	grant	VimSoftwareImage.vimConnectionInfo: extended to include the use case of CISM connections	Modified attribute semantics
9.5.3.10	grant	VimSoftwareImage.vimSoftwareImageId	Modified attribute semantics
10.5.3.2	vnfpkgm	VnfPackageSoftwareImageInfo.diskFormat	Modified attribute cardinality
10.5.3.2	vnfpkgm	VnfPackageSoftwareImageInfo.minDisk	Modified attribute cardinality
10.5.3.2	vnfpkgm	VnfPackageSoftwareImageInfo.minRam	Modified attribute cardinality

F.3.2 ENH02: Special technical enhancements

F.3.2.1 ENH02.04: Invariant identification of NSD constituents

This enhancement proposes to add the capability to identify the VNFDs, nested NSDs and PNFDs of an NSD by invariant identities, as an alternative option to the descriptor identifiers. The enhancement will avoid having to change and create a new NSD when its components (VNFDs, PNFDs or nested NSDs) are replaced by another version and this replacement does not require changes in the rest of the NSD.

Table F.3.2.1-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
10.5.2.2	vnfpkgm	VnfPkgInfo.vnfdExtInvariantId	New attribute

F.3.2.2 ENH02.05: Flexibility with scalable VNF/NS instantiation

The enhancement feature proposes to add scaleLevels as input during instantiation to support flexibility with scalable VNF/NS instantiation. The VNFs/NSs supporting flexible instantiations are identified with VNFD/NSD level attribute(s). This enhancement provides flexibility for the service providers to adjust instantiation level when instantiating a VNF and supports instantiate a VNF with required size in one single operation.

Table F.3.2.2-1: Changes that make up the feature

Clause	Interface	Content of the change	Type of change
4.4.1.15	multiple	ScaleInfo	Other change: new clause moved from vnflcm interface due to being used by multiple interfaces
5.5.2.4	vnflcm	InstantiateVnfRequest.instantiationLevelId	Modified attribute semantics
5.5.2.4	vnflcm	InstantiateVnfRequest.targetScaleLevelInfo	New attribute
5.5.2.7	vnflcm	ChangeVnfFlavourRequest.instantiationLevelId	Modified attribute semantics
5.5.2.7	vnflcm	ChangeVnfFlavourRequest.targetScaleLevelInfo	New attribute
5.5.3.4	vnflcm	ScaleInfo	Other change: void clause due to being movedto the multiple interface
9.5.2.2	grant	GrantRequest.instantiationLevelId	Modified attribute semantics
9.5.2.2	grant	GrantRequest.targetScaleLevelInfo	New attribute

Annex G (informative): Change History

Date	Version	Information about changes
May 2016	0.0.1	Initial version based on - NFVSOL(16)000008r1 SOL003 ToC Proposal - NFVSOL(16)000009_SOL003_basic_skeleton
September 2016	0.1.0	Contributions incorporated - NFVSOL(16)000092_SOL003_structure_update - NFVSOL(16)000085r2_SOL003_InstantiateVnf_flow - NFVSOL(16)000076r2_SOL002___SOL003_-_4_2_Specification_Methodology - NFVSOL(16)000081r3_SOL003_-_Interface_design_for_CreateVNFId Editorials - Removed the old clause 5 and 6 structure as this is not in line anymore with the template - Added reference to RFC7159 as instructed in the Editor's note in document 76r2, removed the Editor's note, and changed the style of the reference to comply with ETSI Drafting Rules - Added applicable abbreviations - Added changes of GS title page requested by Technical Steering Committee (NFVTSC(16)000052) - Added editorials requested by Ericsson
October 2016	0.1.5	Contributions incorporated that were agreed at SOL#11 call - NFVSOL(16)000105r1_SOL003_align_GS_with_template_changes_in_104 - NFVSOL(16)000096_SOL003_4_2_RFC3986 - NFVSOL(16)000101_SOL003__Removal_of_Editor_s_Note_in_Section_5_2_1_Creation - NFVSOL(16)000107r1_SOL003_Scope
November 2016	0.2.0	Contributions incorporated that were agreed at SOL#12 meeting in Bundang - NFVSOL(16)000097r3_SOL003_4_2_Normative_status_of_https - NFVSOL(16)000100r3_SOL003_Add_JSON_schema_Annex_skeleton - NFVSOL(16)000102r3_SOL003_VNF_Lifecycle_Management_interface___Terminate_VNF_fl - NFVSOL(16)000108r1_SOL003_Interface_overview_annex - NFVSOL(16)000109_SOL003_operations_mapping_annex - NFVSOL(16)000110r3_SOL002_SOL003_Delete_VNF_Identifier_Interface - NFVSOL(16)000113r3_SOL003_ScaleVnf_flow - NFVSOL(16)000115r2_SOL003_QueryVnf_flow - NFVSOL(16)000116r3_SOL003_GetOperationStatus_flow - NFVSOL(16)000118r2_SOL002_SOL003_instantiate_VNF_resource_description - NFVSOL(16)000119r3_SOL002_SOL003_terminate_VNF_resource_description - NFVSOL(16)000120r4_SOL003_LCM-Operate_Interface - NFVSOL(16)000122r1_SOL003_data_type_of_CreateVnfRequest - NFVSOL(16)000128_SOL003_small_change - NFVSOL(16)000129r2_SOL003_Instantiate_flow_changes - NFVSOL(16)000132r1_SOL003_LCCN_notification_interface_resources_and_methods - NFVSOL(16)000134r1_SOL003_Resource_of_VNF_lifecycle_operation_occurrences - NFVSOL(16)000137_SOL003_Apply_convention_change_from_document_136

Date	Version	Information about changes
November 2016	0.3.0	Contributions incorporated that were agreed at SOL#13 and SOL#14 - NFVSOL(16)000145r2_Conventions_simplifying_the_table - NFVSOL(16)000131r2_SOL003_Granting_interface_resources_and_methods - NFVSOL(16)000140R1_SOL003_Create_Delete_VNF_Identifier_-_make_consistent_with_REST - NFVSOL(16)000156R1 Change Flavour Inerface
December 2016	0.4.0	Contributions incorporated that were agreed in SOL#15 call and at SOL#16Shenzhen F2F - NFVSOL(16)000144r3_SOL003_refactoring_the_LCM_flows (aligned style of headlines of flows afterwards as an editorial action, and moved the Delete VNF Instance clause to a more appropriate place) - NFVSOL(16)000159R1_SOL003_LCCN_interface_flows - NFVSOL(16)000160r2_SOL003_LCCN_interface_subscriptions - NFVSOL(16)000161r1_SOL003_LCCN_interface_notification_resources - NFVSOL(16)000167_SOL003_Editors_note_on_granting_OperateVnf - NFVSOL(16)000172r1_SOL003_Attribute_filters (added clause "4.3.1 Introduction" as an editorial action on top of the text introduced by this contribution in order to improve the reading flow). - NFVSOL(16)000174r2_SOL003_LCM_error_handling - NFVSOL(16)000179r3_SOL003_Heal_VNF_Interface - NFVSOL(16)000180r2_SOL003_Scale_VNF_task - NFVSOL(16)000181r2_SOL003_Scale_VNF_to_Level_task - NFVSOL(16)000191r3_SOL003_VNF_Pkg_resources - NFVSOL(16)000193r2_SOL003_QueryVnfPackage_flow - NFVSOL(16)000196r1_SOL003_Query_VNF_resource_description - NFVSOL(16)000197r2_SOL003_General_Aspects__HTTP_headers - NFVSOL(16)000200r1_SOL003_align_resource_description_clauses - NFVSOL(16)000201r2_SOL003_Modify_VNF_Information_Interface - NFVSOL(16)000202r3_SOL003_VNF_Fault_Management_interface - NFVSOL(16)000208_SOL003_nicer_tables (aligned the table in the FM clause as an editorial action) Editorials: - Aligned the format of the references in clause 2.1 - Fixed a copy&paste error in the Granting interface (VNF lifecycle management interface → VNF lifecycle operation granting interface) - Added "Release 2" to title
January 2017	0.5.0	Contributions incorporated that were agreed in NFVSOL#17 call: - NFVSOL(17)000001r1_SOL003_merging_LCM_and_LCCN_interfaces - NFVSOL(16)000173r3_SOL003_Attribute_selectors - NFVSOL(16)000194r4_SOL003_FetchVnfPackage_flow Editorials: - Added "Draft" to page header
January 2017	0.6.0	Contributions incorporated that were agreed at NFVSOL#18 F2F in Munich: - NFVSOL(16)000139r1_SOL003_error_codes_to_signal_application_errors - NFVSOL(16)000195r4_SOL003_FetchVnfPackageArtifacts_flow - NFVSOL(16)000203r2_SOL003_LCCN_interface_notification_data_types (Rapporteur's addition when implementing this contribution: Removed the inline definitions of DateTime as this is now defined as a global type.) - NFVSOL(17)000005r3_SOL003_VnfLcOpOcc_data_structure - NFVSOL(17)000006r1_SOL003_LCM_error_handling_resources - NFVSOL(17)000007r2_SOL003_Indicator_interface_structure - NFVSOL(17)000008_SOL003_Replace_URL_by_URI - NFVSOL(17)000022r1_SOL003_ModifyVnfInformatin_flow - NFVSOL(17)000023r1_SOL003_VR_Quota_Available_resources - NFVSOL(17)000024r1_SOL003_VR_Quota_Available_notification_flow - NFVSOL(17)000026_SOL003_VNF_Pkgm_structure_update - NFVSOL(17)000027r1_SOL003_VnfConfig_implementing_IFA_verdict_part_1_-_PATCH - NFVSOL(17)000028r2_SOL003_VnfConfig_implementing_IFA_verdict_part_2_-_Change_ex - NFVSOL(17)000032r4_SOL003_Granting_resources - NFVSOL(17)000033r2_SOL003_Granting_data_types - NFVSOL(17)000035r1_SOL003_LCM_flow_update - NFVSOL(17)000046_SOL003__Change_to_the_Resource_Representation_of_an_Individu

Date	Version	Information about changes
		Editorials: - Fixed some typos - s/DF/deployment flavour/ - removed the mentioning of boldfaced text in the text referring to figure 5.5.2.1-1 - s/encodeddata/encoded data/ - s/and includes in the entity body binary-encoded data/and includes binary-encoded data in the entity body/ - Fixed leftovers of "information element" in Granting interface (copy&paste error from IFA007)
February 2017	0.7.0	Contributions incorporated that were agreed at NFVSOL#19 call and NFV#20 in Bilbao: - NFVSOL(17)000038r2_SOL003_Add_resource_and_enumeration_for_alarm - NFVSOL(17)000047r1_SOL003_PATCH_entity_body - NFVSOL(17)000079r1_SOL003_General_Data_types - NFVSOL(17)000048r2_SOL003_Indirect_RM - NFVSOL(17)000049_SOL003_Error_handling_Granting_interface - NFVSOL(17)000056r1_SOL003_PM_interface_structure (as agreed in Bilbao, clauses related to subscription resources have been moved to the end of the resources clauses to align with other APIs) - NFVSOL(17)000059_SOL003_Adding_criterion_to_LccnSubscriptionFilter - NFVSOL(17)000064r1_SOL002_SOL003_and_SOL005_Labelling_of_API_Names - NFVSOL(17)000066r2_SOL003_VimInfo_fixes_plus_InterfaceInfo_and_AccessInfo_defin - NFVSOL(17)000074_SOL003_VNF_package_management_notification_flows - NFVSOL(17)000080r1_SOL003_VNF_package_management_resources - NFVSOL(17)000083_SOL003_Error_handling_LCM_interface (with the following modification by the rapporteur: replaced multiple occurrences of the copy-paste error "corresponding to the instantiation operation" by "corresponding to the operation") - NFVSOL(17)000085r1_SOL003_Flows_for_Indicator_interface - NFVSOL(17)000086r1_SOL003_Resources_of_Indicator_interface - NFVSOL(17)000087r1_SOL003_Filters_and_Selectors_for_the_LCM_interface - NFVSOL(17)000090r2_SOL003_Links_for_LCM_and_Granting_interfaces - NFVSOL(17)000091_SOL003_LifecycleChangeNotification_terminology (with the following modifications by the rapporteur: (1) use lowercase in "lifecycle management operation occurrence" if it appears in flowing text. (2) change "represented by a VNF Lifecycle Operation Occurrence" → "represented by a VNF Lifecycle Operation Occurrence resource") - NFVSOL(17)000092r2_SOL003_VNF_Fault_Management_interface_data_model (with the following modifications by the rapporteur: as the contribution proposes two alternatives for the request body, added the sentence "Each notification request body shall include exactly one of the alternatives defined in table 7.3.6.3.1-2" that we use in other occurrences of this pattern. - NFVSOL(17)000096_SOL003_VNF_fault_management_subscription_and_notification_fl - NFVSOL(17)000097r2_SOL003_VNF_package_management_data_models - NFVSOL(17)000098r2_SOL003_notification_and_filter_design_for_vr_quota_avail_not - NFVSOL(17)000099r1_SOL003_error_codes_design_for_vr_quota_avail_notification_in - NFVSOL(17)000100r1_SOL003_Data_structures_of_the_Indicator_interface - Aligned Annex A with Conventions change in NFVSOL(17)000106_Conventions_Document_NFVSOL_17_000050__Swagger_Representatio - NFVSOL(17)000111_SOL003_Conventions_move_Resource_structure_up_in_the_TOC Editorials: - Sorted the references. Removed the two informative references to JSON schema as they are not referenced any longer. - Various typo and numbering corrections - In FM interface, moved subscriptions resources to the end of the resources clauses to align with the structure of the other APIs - Applied those changes in NFVSOL(17)000084r1_Template_changes_for_error_handling that are editorial (note that technical changes regarding error hand ling etc. require a contribution) - In a number of NOTEs, replaced "softwareVersion" by "vnfSoftwareVersion". In the table, the name of the attribute to which the note refers was changed to "vnfSoftwarVersion" but that was forgotten in the NOTE.

Date	Version	Information about changes
		- Moved the datatype definition for VnfLcOpOcc in front of the subscription data types in order to align with the sequence of the resources
March 2017	0.8.0	Stable Draft, with a small number of noted gaps as defined in contribution NFVSOL(17)000009r9. Contributions incorporated that were agreed at NFVSOL#21 call and NFV#22 in Piscataway: - NFVSOL(17)000078r3_SOL002_SOL003_VimId_fixes (Additionally, rapporteur deleted one related editor's note in table 9.5.3.3-1 that was forgotten to be deleted by contribution 78r3) - NFVSOL(17)000118r1_SOL003_VNF_performance_management_pmJobs_flows - NFVSOL(17)000119r3_SOL003_VNF_performance_management_thresholds_flows - NFVSOL(17)000120_SOL003_VNF_performance_management_subscription_and_notification - NFVSOL(17)000124r1_SOL003_Authorization - NFVSOL(17)000125r1_SOL002_SOL003_MonitoringParameters_data_structure - NFVSOL(17)000126r1_SOL002_SOL003_NetworkAddress_data_structure - NFVSOL(17)000127r2_SOL003_Fixes_related_to_IFA_decisions_in_Bilbao_and_beyond (Additional change was performed by the rapporteur to Annex B.2 to align with the removal of "Fetch VNF Package" which corresponds to the spirit of this contribution but was forgotten) - NFVSOL(17)000128r2_SOL002_SOL003_Addressing_Rapporteur_s_and_Editor_s_notes_bat - NFVSOL(17)000139_SOL003_apply_conventions_to_vr_quota_avail_notification_flow - NFVSOL(17)000141_SOL002_SOL003_Rename_ind_to_vnfind - NFVSOL(17)000147r1_SOL002_SOL003_Definitions__symbols_and_abbreviations - NFVSOL(17)000148r1_SOL003_Clause_4_1_Overview - NFVSOL(17)000149_SOL002_SOL003_LCM_ed_note_error_handling_bugfix - NFVSOL(17)000150r1_SOL002_SOL003_Adding_description_of_rollback_retry_cancel_fa - NFVSOL(17)000151r1_SOL003_SOL002_clause_5_6_1_Basic_concepts__for_LCM_errors_ - NFVSOL(17)000152_SOL003_Move_LcmOpType_to_global_types_list - NFVSOL(17)000153r2_SOL003_PM_ThresholdCriteria - NFVSOL(17)000154r2_SOL003_PM_interface_data_model - NFVSOL(17)000155r1_SOL003_SOL002_resolve_Auto-X_editor_s_note - NFVSOL(17)000157_SOL003_SOL002_5_2_1_Fixing_Vnf_Instance_Creation_flow - NFVSOL(17)000169_SOL003_Adding_VNFD_resource_in_VNF_package_management_interf - NFVSOL(17)000170r1_SOL003_Read_VNFD_flow - NFVSOL(17)000171r1_SOL003_Add_selectors_for_VNF_package_management_interface - NFVSOL(17)000172_SOL003_Add_filters_for_VNF_package_management_interface - NFVSOL(17)000173_SOL003_Add_filters_for_vr_quota_available_notification_inter - NFVSOL(17)000174r1_SOL003_modify_data_type_of_the_artifacts - NFVSOL(17)000175r1_SOL003_define_checksum_data_type_for_the_VNF_package_managem - NFVSOL(17)000176r1_SOL003_Add_alarm_resource_to_FM_and_fix_links - NFVSOL(17)000177r2_SOL003_VNF_fault_management_related_data_types - NFVSOL(17)000178_SOL003__Correction_of_vnfdId_Data_Type_in_VnfInstance__Claus - NFVSOL(17)000180r2_SOL003_VNF_Fault_Management_Get_Alarm_List_sequence_flow - NFVSOL(17)000181r3_ SOL003_Update_FM_Resources - NFVSOL(17)000182_SOL002_SOL003_Indicators clean up - NFVSOL(17)000187r1_SOL002_SOL003_Conventions_global_fix_for_normative_statement - NFVSOL(17)000188_SOL002_SOL003_Notification_id - NFVSOL(17)000189r1_SOL002_SOL003_VnfInstanceSubscriptionFilter_general_data_typ - NFVSOL(17)000190_SOL002_SOL003_VnfLcOpOcc_fixes_for_ModifyVnfInfo - NFVSOL(17)000191r1_SOL002_SOL003_state_change_timestamp_and_affected_resources (but "attributeChanges was not added to the exclude-default for VnfLcOpOcc in the last change, as this attribute has been removed from the contribution in r1, but removal from that particular change was missed) - NFVSOL(17)000192_SOL002_SOL003_Remove_editor_s_note_in_5_4_3_3_4

Date	Version	Information about changes
		- NFVSOL(17)000193r1_SOL003_Resumed___partial_download_of_artifacts_and_VNF_packa (and applied the change we agreed in another document for the 200 OK response for fetching a VNF package artifact also to the 206 response) - NFVSOL(17)000199_SOL002_SOL003_Renaming_attribute_selectors - NFVSOL(17)000200_SOL002_SOL003_Attribute_filter_equality - NFVSOL(17)000201_SOL003_FM_align_fault_type_and_event_type_with_IFA_contribut - NFVSOL(17)000202r1_SOL003_Annex_B_Mapping - NFVSOL(17)000203_SOL003__VNF_FM_Interface_Terminology_Clean-Ups - NFVSOL(17)000204_SOL003_error_handling__filters_and_selectors_for_VNF_perform Editorials: - A number of small typos fixed - Removed ":" after "Location" when the "Location" HTTP header is mentioned. - Reworded two awkwardly-formulated sentences about the NFVO trying to recover from an error situation, in the subscription management flows of FM and PM i/fs) - Aligned the intro text of the clauses "Referenced simple types and enumerations", and replaced empty "Referenced simple types and enumerations" clauses by references to clause 4.4. - Captions of tables that define an enumeration have been aligned with the template - Changed name of the "Remarks" column to "Description" globally - Removed template leftovers from the front matter.
April 2017	V 0.8.1	Editorial changes to prepare for clean-up by editHelp!
April 2017	V 0.8.2	Clean-up done by ***editHelp!***
May 2017	V 0.9.0	Contributions incorporated that were agreed at NFVSOL#23 and NFVSOL#25 calls and in email approval round #1: - NFVSOL(17)000209r1_SOL002_SOL003_SOL005_all_fields__and_defaults_for_selectors - NFVSOL(17)000225r2_SOL003_clause_6_Fetching_of_Performance_reports - NFVSOL(17)000226r1_SOL003_global_Harmonize_Query_vs_Read - NFVSOL(17)000237r1_SOL003_6_5_2_4_EN_add__id_to_ThresholdCrossedNotification - NFVSOL(17)000238_SOL003_6_5_2_5_EN_add__id_to_PerformanceInformationAvailable - NFVSOL(17)000242_SOL003_10_4_5_3_2_and_10_4_6_3_2_EN_Non-support_of_range_req - NFVSOL(17)000245r2_SOL002_SOL003_5_x_harmonization_of_VnfLcOpOcc_related_identi - NFVSOL(17)000247_SOL003_B_7_small_bugfix - NFVSOL(17)000258r1_SOL003_5_6_Better_name_for_unresolvable_fail - NFVSOL(17)000257_SOL003_5_3_vnfInstanceId_in_LCM_flow_diagrams (applies to ModifyVnfInfo) - NFVSOL(17)000260r4_SOL003_global_editorial_and_minor_technical_changes - NFVSOL(17)000294_SOL003_4_5_2_3_Clarification_of_two_access_token_usage (Rapporteur: and added "#2" in the text in a few additional places to align with the figure) - NFVSOL(17)000295_SOL003_4_5_2_3_flow_and_step_consistency - NFVSOL(17)000297_SOL003_5_3_12_Missing_error_handling_description - NFVSOL(17)000304_SOL003_removing_status_from_SoftwareImageInformation - NFVSOL(17)000305_SOL003_4_1_Overview_Consistency_Fix - NFVSOL(17)000309_SOL003_4_5_2_Authorization_flow_description_inconsistency - NFVSOL(17)000315_SOL003_4_5_2_fixing_HTTP_method_in_OAuth_2_0_flows - NFVSOL(17)000317_SOL003_5_5_2_13_operationType_in_VnfLcOpOcc (Rapporteur: In table 5.5.2.13-1, it was missed to delete "Type" from "operationType"; all other occurrences use "operation" not "operationType") - NFVSOL(17)000321_SOL003__9_5_2_3_Change_VimInfo_data_type_to_VimConnectionInf - NFVSOL(17)000323_SOL003_Clause_7_5_4_3_Change_the_description_of_PerceivedSev - NFVSOL(17)000329_SOL003_global_ignore_unknown_attributes - NFVSOL(17)000336r1_SOL003_many_Fixing_some_conditions (Rapporteur: and aligned the new text with 245r2, i.e. "VNF LCM operation occurrence" instead of "operation")

		Contributions incorporated that were agreed at NFVSOL#26 and NFVSOL#27 calls, in email approval rounds #2 and #3, and in SOL#28 F2F: - NFVSOL(17)000217r4_SOL003_VNF_FM_Acknowledge_Alarm_operation - NFVSOL(17)000218r2 - NFVSOL(17)000227r1_SOL003_4_3_2_2_EN_Attribute_filters_point_in_time - NFVSOL(17)000228r1_SOL003_4_4_1_3_EN_for_KeyValuePair___Object - NFVSOL(17)000229r1_SOL003_10_5_3_2_VnfPackageSoftwareImageInfo_fixes - NFVSOL(17)000230r3_SOL003_global_Apply_conventions_for_identifiers - NFVSOL(17)000231r1_SOL003_5_5_3_5_ENs_VimConnectionInfo_registry - NFVSOL(17)000232_SOL003_5_5_3_7_rapp_note__information_element - NFVSOL(17)000233r2_SOL003_5_5_3_7_to_5_5_3_9_resource_metadata - NFVSOL(17)000234r2_SOL003_5_6_2_3_EN_regarding_forceful_vs_graceful_cancel - NFVSOL(17)000235_SOL003_6_3_3_rapporteur_s_notes_regarding_PM_flow - NFVSOL(17)000236_SOL003_6_3_6_rapporteur_s_note_regarding_PM_flow_bugs - NFVSOL(17)000240_SOL003_9_5_3_3_EN_in_table_for_GrantInfo - NFVSOL(17)000241_SOL003_9_5_2_2_and_9_5_2_3_EN_and_rapporteurs_note - NFVSOL(17)000244_SOL003_5_x_Problem_with_Error_handling_ModifyVnfInfo - NFVSOL(17)000246r1_SOL003_5_4_16_Making_the_Fail_operation_synchronous - NFVSOL(17)000256r1_SOL003_4_4_1_3_VnfInstanceSubscriptionFilter_fixes - NFVSOL(17)000261r1_SOL003_10_3_3_and_10_4_5_3_2_ZIP_as_content_type_for_VNF_Pac - NFVSOL(17)000262_SOL003_9_4_3_Deletion_of_grants - NFVSOL(17)000280r2_SOL003_-_Clause_4_4_2_-_5_5_3_-_MAC_and_IP_address_represent (Rapporteur: renamed IPAddress to IpAddress to follow conventions) - NFVSOL(17)000282_SOL003_-_Removing_normative_dependencies_on_SOL001 - NFVSOL(17)000291_SOL003_Clause_4_4_2_simple_data_type_editor_note - NFVSOL(17)000293r1_SOL003_4_2_Consistency_of_URI_and_OAuth - NFVSOL(17)000298r2_SOL003_5_4_3_2_Improvement_of_resource_definition_descriptio - NFVSOL(17)000299r1_SOL003_5_4_13_2_Improvement_of_resource_definition_descripti - NFVSOL(17)000300_SOL003_4_4_2_Clear_meaning_of_IdentifierLocal_type - NFVSOL(17)000301r1_SOL003_5_4_16_Add_the_supplement_to_Finally_Failed - NFVSOL(17)000302r1_SOL003_4_3_5_5_Consistency_between_4_3_5_4_and_4_3_5_5 - NFVSOL(17)000303r3_SOL003_-_Editorial_changes - NFVSOL(17)000306r1_SOL003_4_3_2_2_Filter_Spec_Fix - NFVSOL(17)000307r2_SOL003_4_3_4_3_Adding_retry-after_header_field - NFVSOL(17)000308r1_SOL003_4_3_5_Bug_fix_and_modify_title - NFVSOL(17)000310r1_SOL003_4_5_3_Small_bug_fixes (rapporteur also accordingly changed other places that mention that " Link to the subscription that triggered the notification" to "Link to the related subscription") - NFVSOL(17)000311r1_SOL003_5_3_9_Retry_flow_error_handling_description_modificat - NFVSOL(17)000313_SOL003_5_4_12_3_2_Small_bug_fix - NFVSOL(17)000314_SOL003_5_5_2_2_modify_the_description_for_VnfInstance - NFVSOL(17)000316_SOL003_5_5_2_8_Graceful_Termination_clarification - NFVSOL(17)000318r2_SOL003_5_5_2_16_LCM_Notifications_race_condition - NFVSOL(17)000322r1_SOL003__Various_Clauses__Consistency_in_the_naming_of_the_ad - NFVSOL(17)000326_SOL003_6_5_3_2_PmFilter - NFVSOL(17)000328r4_SOL003_5_5_Making_the_relationship_between_vnflcOpOcc_and_no - NFVSOL(17)000330_SOL003_9_5_2_3_and_9_5_3_3_Grant_fixes - NFVSOL(17)000332r2_SOL003_5_5_3_17_Problem_with_storage_resources_in_AffectedVn - NFVSOL(17)000333r3_SOL003_5_5_3_15_wrong_use_of_Object_in_MonitoringParameter - NFVSOL(17)000334r1_SOL003_global_Subscribe_Notify_error_handling - NFVSOL(17)000335r1_SOL003_4_3_3_2_1_Fixes_to_attribute_selector - NFVSOL(17)000338_SOL003_5_5_3_5_Protecting_credentials_in_VimConnectionInfo - NFVSOL(17)000339_SOL003_5_4_2_3_2_Adding_vimConnections_to_exclude_default - NFVSOL(17)000341_SOL003_7_5_2_5_Refactoring_links_in_AlarmNotification

Date	Version	Information about changes
		- NFVSOL(17)000342_SOL002_SOL003_4_4_2_and_10_5_3_2_EN_on_Version_and_unsignedI - NFVSOL(17)000343_SOL002_SOL003_global_Renaming_of_attribute_filters_to_attrib. - NFVSOL(17)000347r4_SOL003_Annex__X___VIM_Registry - NFVSOL(17)000349r2_SOL003_SOL002_5_x_Error_code_404_if_task_resource_not_suppor - NFVSOL(17)000352r1_SOL003_7_5_3_4_Update_data_type_FaultyResourceInfo - NFVSOL(17)000354r3_SOL003_SOL002_autoscale_autohel_description - NFVSOL(17)000355_SOL003_SOL002_Replace_entity_body_by_payload_body - NFVSOL(17)000363_SOL002_SOL003_global_consistency_of_enum_type_names - NFVSOL(17)000368_SOL002_SOL003_4_3_3_1_adding_informative_to_overview_and_exa - NFVSOL(17)000374_SOL003_SOL002_Notification_Authorization_future_proofing - NFVSOL(17)000375r1_SOL003_SOL002_Explanation_of_OperateVnf - NFVSOL(17)000384_SOL003_SOL002_Refactoring_VNF_Instance_link_in_AlarmNotifica (This is a strict subset of 341) - NFVSOL(17)000389r2_SOL003_10_4_4_Fetch_the_VNFD - NFVSOL(17)000393_SOL002_SOL003_Move_VimConnectionInfo_to_the_correct_clause - NFVSOL(17)000394_SOL002_SOL003_address_comments_from_Procera_Networks - NFVSOL(17)000396r2_SOL002_SOL003_6_4_7_4_8_4_9_4_10_4_11_4_Addition_of_the_note - NFVSOL(17)000397_SOL002_SOL003_5_3_3_Clarification_of_notification_flow_in_th - NFVSOL(17)000399_SOL003_SOL002_changedExtVLs_in_LcmOpOccNotif Editorials: - Put "Notify" always as last row in the tables in all clauses of Annex B - Enforced naming convention of using plural for attributes of cardinality 0..N
June 2017	V 0.10.0	Contributions incorporated that were approved at NFVSOL#29 call: - NFVSOL(17)000337r1_SOL003_5_5_2_x_additionalParameters_missing_from_TerminateVn - NFVSOL(17)000387r2_SOL003_SOL002_EN_Scaling_explanation - NFVSOL(17)000403r3_SOL003 ExtCP ExtVL fixes related to IFA discussion - NFVSOL(17)000404_SOL003_SOL002_Align_with_outcome_of_IFA_document_468 - NFVSOL(17)000405r2_SOL003_SOL002_one_change_pulled_out_from_349r1 - NFVSOL(17)000408r3_SOL003_rapporteur_s_cleanup_of_V_0_9_0 - NFVSOL(17)000414r3_SOL003_issue__resourceGroupId_in_Grant_interface - NFVSOL(17)000416_SOL003__Remove_Swagger_Annex_A - NFVSOL(17)000417_SOL003_SOL002_copying_over_a_note_from_IFA007 Editorials: - Fixing awkward "(s)" plural where applicable: (1) use plural where appropriate e.g. for arrays resource(s) → resources, (2) use singular where appropriate, e.g. if there's only a single entry such as notification about quota(s) → quota, (3) use alternatives where there are two distinct possibilities, e.g. VIM interface or VIM interfaces - Removed editor's notes not addressed by contributions, apart from the one in the registry
July 2017	V 0.10.1	- Editorial changes - Fixing the implementation of document NFVSOL(17)000199_SOL002_SOL003_Renaming_attribute_selectors
November 2017	V 2.3.2	Contributions incorporated that were approved at NFVSOL#40 call, NFVSOL#41 F2F and subsequent email approval: - NFVSOL(17)000565_SOL002_SOL003_Fixing_actors_in_authorization_flows - NFVSOL(17)000592r4_SOL003_miscellaneous_bugfixes - NFVSOL(17)000593r1_SOL002_SOL003_miscellaneous_small_bugfixes - NFVSOL(17)000594r1_SOL002_SOL003_Adding_405_response - NFVSOL(17)000596r1_SOL003_Adding_statement_for_mandatory_and_optional_HTTP_meth - NFVSOL(17)000600_SOL003_IFA027_reference - NFVSOL(17)000602_SOL003_Get_Alarm_List_query_fix - NFVSOL(17)000629_SOL003_VR_Quota_Avail_Notification_Trigger_Condition Editorials: - Added draft disclaimer box

Date	Version	Information about changes
November 2017	V 2.3.3	Contributions incorporated that were approved at NFVSOL#42 - NFVSOL#44 calls: - NFVSOL(17)000634_SOL003_VNFC_CP_changes_in_AffectedVnfc - NFVSOL(17)000667r2_SOL002_SOL003_Add_description_to_VNF_fault_management_interf - NFVSOL(17)000668_SOL002_SOL003_complement_the_description_of_CancelModeType - NFVSOL(17)000670r2_SOL002_SOL003_Fixing_statement_for_mandatory_and_conditional - NFVSOL(17)000691r1_SOL002_SOL003__Use_of_verbal_forms_for_the_expression_of_pr
December 2017	V 2.3.4	Contributions incorporated that were approved at NFVSOL#45 F2F, the subsequent email approval and the NFVSOL#46 and NFVSOL#47 calls: - NFVSOL(17)000388r6_SOL002_ed2_4_1_and_SOL003_ed_2_4_1__Authorization_of_API_Req - NFVSOL(17)000635r1_SOL002_SOL003_implicit_changes_in_VnfInfoModifications - NFVSOL(17)000646_SOL002_SOL003_Add_resource_metadata_to_AffectedVnfc_VirtualL - NFVSOL(17)000654r1_SOL003_Align_PkgmNotificationsFilter_with_VnfInstanceSubscri - NFVSOL(17)000666r1_SOL003_Refactoring_VNF_package_management_interface_to_align - NFVSOL(17)000669r3_SOL002_SOL003_Add_note_to_clarify_how_timeout_of_the_cancel_ - NFVSOL(17)000671r2_SOL002_SOL003_ExtCpData_changes_from_IFA1114r1_and_IFA1037r4 (in clause 5.5.3.5, rapporteur has changed in the Description column "vnfLinkPortInfo" to "vnfLinkPorts" to reflect the correct attribute name ("VnfLinkPortInfo" is the type, "vnfLinkPorts" is the attribute) - NFVSOL(17)000674r4_SOL002_SOL003_Authorization_method_negotiation - NFVSOL(17)000690r1_SOL003_5_5_2_10_Add_note_for_the_presence_condition_of_attri - NFVSOL(17)000695r1_SOL002_SOL003_Fixing_normative_status_of_notification_endpoi - NFVSOL(17)000698_SOL002_SOL003_allow_Fail_operation_in_FAILED_TEMP - NFVSOL(17)000706r1_SOL003_Solve_the_inconsistency_of_password_transmission - NFVSOL(17)000715r2_SOL002_SOL003_-_Double_Subscriptions_for_Notifications - NFVSOL(17)000718r2_SOL002_SOL003_sequence_of_responses_and_notifications - NFVSOL(17)000723r1_SOL003_fixing_VNF_connectivity_figure - NFVSOL(17)000730r1_SOL003_Fix_description_of_unsupported_method_for_notification - NFVSOL(17)000733_SOL003_align_normative_statements_in_resource_tables - NFVSOL(17)000734r1_SOL003_align_normative_statements_in_trigger_conditions - NFVSOL(17)000739_SOL003_VimConnectionInfo_fix_the_word_needed - NFVSOL(17)000749_SOL003_Add_error_code_for_fetching_package_content_or_artifa - NFVSOL(17)000754_SOL002_SOL003_Remove_redundant_description_of_vnfConfigurabl - NFVSOL(17)000757_SOL003_Add_clarification_of_ExManagedVirtualLink_and_ExtVirt - NFVSOL(17)000768_SOL003_Annex_A_2_Operation_Name_and_Resource_URI_Alignment - NFVSOL(17)000775_SOL003_VnfPkgm_undoing_the_timestamp_change_from_CR_666r1 - NFVSOL(17)000780r1_SOL003_Fixing_leftovers_of_onboardedVnfPkg_Info_Id - NFVSOL(17)000789r2_Additional_modifications_for_SOL002_and_SOL003 Some editorial fixes were applied.
February 2018	2.3.5	Incorporated change request: - NFVSOL(18)000022_SOL003_Temporarily_remove_IFA027_reference
February 2018	2.4.1	Publication

Date	Version	Information about changes
March 2018	2.4.2	Contributions incorporated that were approved at NFVSOL#55 F2F: - NFVSOL(18)000037r1_SOL003_API_Authorization_clarification - NFVSOL(18)000058r2_SOL003ed251_Making_Authz_negotiation_more_flexible - NFVSOL(18)000060r2_SOL003ed251_disambiguating_artifactPath - NFVSOL(18)000091r2_SOL003ed251_evolving_the_registry_annex_part_1 - NFVSOL(18)000081r1_SOL003ed251_empty_collections_clarification_adressing_Plugte
March 2018	2.4.3	Contributions incorporated that were approved at NFVSOL#57 call: - NFVSOL(18)000095r1_SOL003ed251_evolving_the_registry_annex_part_2
May 2018	2.4.4	Contributions incorporated that were approved from NFVSOL#58 call to NFVSOL#64 F2F: - NFVSOL(18)000127r1_SOL003ed251_fixing_tracker_issues_0007747_and_000774 78 - NFVSOL(18)000059r1_SOL003_Updating_JSON_RFC_reference - NFVSOL(18)000131r2_SOL003ed251__Fix_cardinality_of_the_operationParams_att ribut - NFVSOL(18)000153r6_SOL003_Version_Management (Rapporteur has slightly changed the statement "and API version retrieval" added to 5.2 etc. to read "and API version information retrieval", as we are not retrieving API versions, but API version information.) - NFVSOL(18)000181r2_SOL003ed251__Change_the_cardinality_of_the_subscriptio nId_at - NFVSOL(18)000210r1_SOL003_Attribute_selectors - NFVSOL(18)000214r2_SOL003_Adding_status_codes - NFVSOL(18)000215_SOL003_Aligning_operation_type_enums_with_IFA007 - NFVSOL(18)000216r1_SOL003_MAC_address_optional_in_IpOverEthernetAddress Info - NFVSOL(18)000217r1_SOL003_Small_fixes - NFVSOL(18)000218_SOL003_SOL005_VnfPkgm_small_fix - NFVSOL(18)000219_SOL003_fix_for_the_enhanced_patch_rules - NFVSOL(18)000220r1_SOL003_enhanced_patch_rules_-_deletion_of_array_entries - NFVSOL(18)000224r2_SOL003_Fixing_the_sequence_of_400_response_code_defi nitions - NFVSOL(18)000226r1_SOL003_different_names_for_virtual_link_descriptor_ids - NFVSOL(18)000227_SOL003_Granting_policies_from_IFA007 - NFVSOL(18)000234_SOL003_Fixing_cardinality_of_ConstraintResourceRef Editorials - TRUE -> true consistently - Removed smart quotes
June 2018	2.4.5	Contributions incorporated that were approved at NFVSOL#66 F2F: - NFVSOL(18)000053r2_SOL003ed251_Bring_back_IFA027_reference - NFVSOL(18)000209r2_SOL003_Attribute_filters - NFVSOL(18)000212r1_SOL003_Normative_attribute_filters_support (in addition to implementing the tracked changes in the CR, the rapporteur also applied one missed change in 6.4.2.3.2 that is part of the pattern: adding "by the VNFM in the filter expression" to the last sentence of the description of the "filter" parameter). - NFVSOL(18)000213r2_SOL003_Support_for_links_in_notifications - NFVSOL(18)000221_SOL003_metadata_for_CP_IEs - NFVSOL(18)000250_SOL003_small_fix_replace_queried_by_read - NFVSOL(18)000257_SOL003ed251__Remove_the_current_values_of_the_MonitoringPara
July 2018	2.4.6	Contributions incorporated that were approved in EA following NFVSOL#66 F2F: - NFVSOL(18)000309_SOL003_Define_Number_and_String_data_types - NFVSOL(18)000241r2_SOL003_Changes_from_IFA_CRs_412r2_and_411r1 Editorials: - Replaced "present specification" by "present document" - Fixed missing comma in table 4.3.2.2-1
July 2018	2.4.7	Contributions incorporated: - NFVSOL(18)000316r2_SOL003_Define_Minor_version_number - NFVSOL(18)000317_SOL003_Retry_as_reaction_to_error_responses_during_notific at - NFVSOL(18)000337_SOL003_attribute_selector_attribute_filter_small_fixes - NFVSOL(18)000339_SOL003_mirror_of_332_-_Add_annex_with_a_reference_to_OpenAPI - NFVSOL(18)000345_SOL003_registry_link_update - NFVSOL(18)000434r1_SOL003_remaining_ENs_resolution

Date	Version	Information about changes
August 2018	2.4.8	Draft as input for approval process towards publication as 2.5.1. Contributions incorporated: - NFVSOL(18)000418r2_SOL003_closing_pagination_gap - NFVSOL(18)000441_SOL003_Define_Patch_version_number - NFVSOL(18)000437r2_SOL003_VIM_registration - NFVSOL(18)000237r5_SOL003_-_API_Version_Identification o Rapporteur has applied editorial changes on top of this CR, including deletion of the trailing phrase "of the previous version (see clause 4.6.2.2)" in the second paragraph of clause 4.6.2.1 which did not fit to the sentence and is considered an editing leftover. - NFVSOL(18)000343r4_SOL003_version_signalling - NFVSOL(18)000471r1_SOL003ed251__Add_note_to_MAJOR_version_field Editorials: - Use "annex" (lowercase) consistently.
September 2018	2.5.1	Publication by ETSI
September 2018	2.5.2	Contributions incorporated: - NFVSOL(18)000550r3_SOL003ed261_Definition_of_the_Grant_data_type - NFVSOL(18)000551_SOL003ed261_fixing_of_reference_error - NFVSOL(18)000552r5_SOL003ed261_Handling_of_inputs_for_bootdata_in_the_API Editorials: - Added frontmatter for drafts
November 2018	2.5.3	Contributions incorporated: - NFVSOL(18)000584r2_SOL003ed261_Referring_to_SOL013 - NFVSOL(18)000581r2_SOL003ed261_Metadata_Extension_ConfigurableProps_clarificati
December 2018	2.5.4	Contributions incorporated: - NFVSOL(18)000692r4_SOL003ed261_Add_best_effort_in_PlacementContraint - NFVSOL(18)000726r1_SOL003ed261_Garbage_collection_of_lcmOpOcc_resources - NFVSOL(18)000728r1_SOL003ed261_Small_technical_fixes - NFVSOL(18)000729r1_SOL003ed261_Aligning_conditions_for_vduId_and_resource Templa - NFVSOL(18)000754r1_SOL003ed261_Remove_shalls_in_selected_SOL013_references - NFVSOL(18)000743r2_SOL003ed261_declaration_of_metadata_and_extensions - NFVSOL(18)000689_SOL003ed261_Normative_statements_for_TST
February 2019	2.5.5	Contributions incorporated: - NFVSOL(19)000015r1_SOL003ed261_Normative_changes_for_TST_part_2 - NFVSOL(19)000016_SOL003ed261_moving_note_to_correct_place - NFVSOL(19)000022r3_SOL003ed261_VnfPkgm_bugfixes - NFVSOL(19)000025_SOL003ed261_alternative_access_to_VNF_package_resources_base - NFVSOL(19)000040_SOL003ed261_artifactPath_clarification - NFVSOL(19)000055_SOL003ed261_remove_editor_s_notes - NFVSOL(19)000057_SOL003ed261_vnfdId_replacing_vnfPkgId_in_LCM_interface_-_NBW - NFVSOL(19)000103_SOL003ed261_Version_fields_update_for_publication Editorials: - Fixed colored text in 5.5.2.2 - Fixed various small typos, including vPkgId -> vnfPkgId
April 2019	2.6.1	Publication by ETSI
July 2019	2.6.2	Contributions incorporated: - NFVSOL(19)000331_SOL003ed271_Aligning_with_SOL015 - NFVSOL(19)000211_SOL003ed271_bugfix_of_placement_constraint_example

Date	Version	Information about changes
October 2019	2.6.3	Contributions incorporated: - NFVSOL(19)000454_SOL003ed271_fixes_to_align_with_SOL002 - NFVSOL(19)000467_SOL003_fixes_related_to_IFA027 - NFVSOL(19)000477r2_SOL003ed271_Bulk_fetch_of_package_artifacts.docx - NFVSOL(19)000482r1_SOL003ed271_Fetching_the_manifest - NFVSOL(19)000483_SOL003ed271_Exposing_MaxScaleLevel - NFVSOL(19)000514_SOL003ed271_moving_datatypes - NFVSOL(19)000523r1_SOL003ed271_example_of_artifactPath_in_GET - NFVSOL(19)000525_SOL003ed271_Marking_non-MANO_artifacts_in_VNF_Package_Info - NFVSOL(19)000541_SOL003ed271_bugfix_ModifyVnfInfo_condition - NFVSOL(19)000564r2_SOL003ed271_How_to_transmit_accessInfo_JSON_data_structure - NFVSOL(19)000576_SOL003Ed271_-_Initial_configurable_properties_values - NFVSOL(19)000581r1_SOL003ed271_fixing_the_PM_interface_wrt_subscriptions - NFVSOL(19)000584_SOL003ed271_fixes_to_FM_interface - NFVSOL(19)000588r2_SOL003ed271_Moving_pre_and_post-conditions_into_normative_cl - NFVSOL(19)000602_SOL003ed271_PATCH_alarm_acknowledge_status Editorials: - Fixed small typos and comma issues - Fixed wrong cross-references in 5.4.1.2 (correct: 5.4.13) and 5.4.18.3.1 (correct: 5.4.19) - Fixed the line styles of task resources in the resource trees - Replaced "Those" in the recurring note 'Those attributes are marked as "required" in the VNFD.' by "Required" to align with SOL002 - Fixed reference number to SOL015 - Fixed styles in annexes - Fixing the introduction clause of Annex B.3 (B.3.1) (Title and wrong references) - Table 5.5.3.8-1: swap cardinality and type in last row
October 2019	2.6.4	Contributions incorporated: - NFVSOL(17)000392r1_SOL003_SOL002_missing_notification_triggers: This contribution was missed earlier, but most of the proposed changes were executed by later CRs. There are two bits of text remaining that were still missing from SOL003 and that have been added, related to the changes in 5.5.2.16 and to the AlarmListRebuiltNotification trigger. None of these is technically critical; they are clarifications. - NFVSOL(19)000328r3_SOL003ed271_Clarify_passing_of_external_connectivity_informa - NFVSOL(19)000563r2_SOL003ed271_Update_attribute_description_and_state_diagram - NFVSOL(19)000569r3_SOL003ed271_Support_rollback_for_failing_VNF_instantiation - NFVSOL(19)000579r3_SOL003ed271_Enable_mapping_of_VR_and_zone - NFVSOL(19)000580r2_SOL003ed271_Mapping_of_granted_and_allocated_resources - NFVSOL(19)000653r2_SOL003ed271_Representing_the_artifact_path_of_external_artif - NFVSOL(19)000659_SOL003ed271_Fixing_non-supprt_of_range_requests - NFVSOL(19)000661_SOL003ed271_replacing_client_by_API_producer - NFVSOL(19)000667_SOL003ed271_fixes_to_IFA_mapping_annex_related_to_dropping_s - NFVSOL(19)000675r3_SOL003ed271_Obtaining_artifact_security_info_via_the_API - NFVSOL(19)000679_SOL003ed271_adding_error_response_for_failed_notification_en - NFVSOL(19)000675r3_SOL003ed271_Obtaining_artifact_security_info_via_the_API - NFVSOL(19)000677_SOL003_fixing_VnfPackageChangeNotification_condition - NFVSOL(19)000712r1_SOL003ed271_mark_for_testing_and_license_artifacts - NFVSOL(19)000722r1_SOL003ed271_VNF_package_metadata_in_VnfPkgInfo Editorials: - Voided "Author and contributors" annex - Typos corrected - Fixed: NFVSOL(17)000218r2 was implemented but NFVSOL(17)000218r1 was listed in the history box

Date	Version	Information about changes
		- Fixed: NFVSOL(17)000404 was implemented in V 0.10.0 but not listed in the history box - Removed remaining rapporteur's notes
November 2019	2.6.5	Contributions incorporated: - NFVSOL(19)000658r1_SOL003ed271_Mirror_of_649_Missing_error_state_in_VNF_Package - NFVSOL(19)000708r2_SOL003ed271_Further_clarify_the_zip_file_structure_returned (editorial update: merged the three occurrences of the Note into one) - NFVSOL(19)000749_SOL003ed271_mirror_of_474r1_version - NFVSOL(19)000752r2_SOL003ed271_rapporteur_s_cleanup - NFVSOL(19)000753_SOL003ed271_API_versions_for_V2_7_1 - NFVSOL(19)000755_SOL003ed271_Fix_for_the_disabled_VNF_packages Editorials: - Fixed history box as it did not mention 712r1 even though that CR was implemented in 2.6.4.
December 2019	2.7.1	Contributions incorporated: - Last minute CR in NFV(19)000291 incorporated to SOL003_Final_Draft_GS_SOL003_V_2_6_5_ - CR implemented in Table 10.5.3.3-1
November 2019	3.0.1	Contributions incorporated: - BWC: NFVSOL(18)000638r1_SOL003ed311_FEAT15_Clause_5_1__5_2_and_5_3_Adding_snapshot_c - BWC: NFVSOL(18)000639r1_SOL003ed311_FEAT15_Clause_5_4_Adding_snapshot_resources - BWC: NFVSOL(18)000640r1_SOL003ed311_FEAT15_Clause_5_5_Adding_snapshot_data_types - NBWCP: NFVSOL(19)000223_SOL003ed311_FEAT04_Clause_9_5_introduce_host_reservation_in - BWC: NFVSOL(19)000249_SOL003ed311_FEAT15_Clause_5_1__5_2_and_5_3_Revert_to_snapsho - BWC: NFVSOL(19)000288r1_SOL003ed311_FEAT15_New_clause_VNF_Snapshot_Pkg_API - BWC: NFVSOL(19)000289r1_SOL003ed311_FEAT15_VNF_snapshot_pkg_mgmt__API_resources - BWC: NFVSOL(19)000290_SOL003ed311_FEAT15_VNF_snapshot_pkg_mgmt_creation - BWC: NFVSOL(19)000291r1_SOL003ed311_FEAT15_VNF_snapshot_pkg_mgmt_query__update_and_d - BWC: NFVSOL(19)000292r1_SOL003ed311_FEAT15_VNF_snapshot_pkg_mgmt_upload_and_fetch - BWC: NFVSOL(19)000293_SOL003ed311_FEAT15_VNF_snapshot_pkg_mgmt_fetch_artifact - BWC: NFVSOL(19)000294r1_SOL003ed311_FEAT15_VNF_snapshot_pkg_mgmt_build - BWC: NFVSOL(19)000298r1_SOL003ed311_FEAT15_Fixes_to_create_VNF_snapshot_task - BWC: NFVSOL(19)000299_SOL003ed311_FEAT15_Fixes_to_VNF_snapshot_data_types - BWC: NFVSOL(19)000353r2_SOL003ed311_FEAT15_Improving_snapshot_creation_process - BWC: NFVSOL(19)000354r1_SOL003ed311_FEAT15_VNF_snapshot_pkg_mgmt_extract_option_A - ED: NFVSOL(19)000826r1_SOL003ed331_mirror_of_825_fixing_leftovers_of_server

Date	Version	Information about changes
		Editorials: - Changed Release number to 3 - Fixed grammar issues with comma before "and" - Fixed use of lowercase and uppercase in added CRs to be in line with the rest of the GS - Applied the convention for {apiMajorVersion} to those resources that were newly added on top of content from V 2.6.5.
January 2020	3.0.2	Contributions incorporated: - NBWCP: NFVSOL(19)000508r4_SOL003ed331_FEAT02_Add_changeCurrentVnfPackage_LCM - NBWCP: NFVSOL(19)000509r1_SOL003ed331_FEAT02_Add_changeCurrentVnfPackage_Granting - BWC: NFVSOL(19)000618r2_SOL003ed331_FEAT02_Add_vnfdId_to_resources_to_model_partial - BWC: NFVSOL(19)000657r2_SOL003ed331_Feature_annex - BWC: NFVSOL(19)000732_SOL003ed331_FEAT10_Add_specification_for_Multi-Site_Connecti Editorials: - year changed to 2020 - aligned with changes done by editHelp when publishing 2.7.1
January 2020	3.0.3	Contributions incorporated: - BWC: NFVSOL(19)000619r4_SOL003ed331_FEAT02_extensions__confprops_and_metadata_during Editorials: - Fixed occurrences of "callbackURI" and "callbackUri" in flow text which need to be "callback URI". - Fixed font issues in 10.4.4.3.2 - Replaced leftovers of "consumer" by "API consumer" and "producer" by "API producer" where applicable - Fixed commas
March 2020	3.0.4	Contributions incorporated: - NBWCP: NFVSOL(19)000368r7_SOL003ed331_Support_of_Trunking o Classification changed from ERROR to NBWCP by NFVSOL(20)000401r1 - NBWCP: NFVSOL(19)000845r1_SOL003ed331_Patch_semantics_of_passing_metadata__extensions - BWC: NFVSOL(19)000860r2_SOL003ed331_Notification_callback_URI_testing_in_VNFM - BWC: NFVSOL(20)000012r2_SOL003ed331_adding_missing_extensions_and_vnfConfigrableProp - BWC: NFVSOL(20)000069_SOL003ed331_mirror_of_NFVSOL_20_000034_fixing_enumeration_of (Editorial: fixed awkward "(s)" plural) - BWC: NFVSOL(20)000071r2_SOL003ed331_add_missing_support_statements - BWC: NFVSOL(20)000083_SOL003ed331_FEAT15_EN_resolution_VnfSnapshotRes - BWC: NFVSOL(20)000084r1_SOL003ed331_FEAT15_EN_resolution_identifiers_in_reversion - BWC: NFVSOL(20)000085r2_SOL003ed331_FEAT15_EN_resolution_VNFD_in_VNF_snapshot - BWC: NFVSOL(20)000086_SOL003ed331_FEAT15_RN_resolution_types_operations_in_VNF_sna - BWC: NFVSOL(20)000087r1_SOL003ed331_FEAT15_EN_VNF_snapshot_package_state_model

Date	Version	Information about changes
		- BWC: NFVSOL(20)000088r1_SOL003ed331_FEAT15_EN_resolution_VNFD_in_snapshot_package - BWC: NFVSOL(20)000089r1_SOL003ed331_FEAT15_EN_resolution_VNF_snapshot_info_in_packag - BWC: NFVSOL(20)000090r2_SOL003ed331_FEAT15_RN_alignments_to_VNF_packages_API - BWC: NFVSOL(20)000100r2_SOL003ed331_Correction_of_small_bugs - BWC: NFVSOL(20)000110r1_SOL003ed331_Notification_callback_URI_testing_in_VNFM__fixes - BWC: NFVSOL(20)000123_SOL003ed331_Feature_Annex_F_2_1_mirror_of_097 - BWC: NFVSOL(20)000156r1_Revision_of_NFVSOL_19_000851_-_SOL003ed331_addressing_rappor Editorials: - Fixed wrong clause and reference numbering - Aligned spelling of "Individual …" resource
April 2020	3.0.5	Contributions implemented - BWC: NFVSOL(20)000116r2_SOL003ed331_FEAT15_Enhancing_granting_related_to_VNF_snapsho - BWC: NFVSOL(20)000153_SOL003ed331_mirror_of_152r1_Normative_statement_to_reject_Cr - BWC: NFVSOL(20)000159r1_SOL003ed331_Short_LcmOpOccNotifications - BWC: NFVSOL(20)000176_SOL003ed331_mirror_of_173r1_Bulk_fetch_of_external_artifacts - BWC: NFVSOL(20)000178r1_SOL003ed331_Failing_Instantiate_and_ChangeFlavour_in_case_of - BWC: NFVSOL(20)000184_SOL003ed331_FEAT15_Normative_language_for_snapshot_operation - BWC: NFVSOL(20)000185_SOL003ed331_FEAT15_EN_resolution_by_deleting_them - BWC: NFVSOL(20)000186r1_SOL003ed331_FEAT15_EN_resolution_about_checksum - BWC: NFVSOL(20)000199r1_SOL003ed331_FEAT15_vnfSnapshotInfo_bugfix_on_top_of_083 - BWC: NFVSOL(20)000273r1_SOL003ed331_FEAT15_Moving_VNF_snapshot_package_API Editorials: - Added CR BWC/NBWC classification
April 2020	3.0.6	Contributions implemented - BWC: NFVSOL(20)000233r2_SOL003ed331_FEAT15_VNF_snapshot_API_final_fixes - BWC: NFVSOL(20)000260r1_SOL003ed331_Indicator_changes_triggered_by_changeCurrentVnfP - BWC: NFVSOL(20)000286_SOL003ed331_FEAT15_Updates_to_VNF_snapshot_due_to_moving_Vnf - BWC: NFVSOL(20)000288_SOL003ed331_Forward_mirror_of_230r2_SOL016_review_alignments - BWC: NFVSOL(20)000302_SOL003ed331_Forward_mirror_of_095r1_Fixing_notifying_informa - BWC: NFVSOL(20)000337_SOL003ed331_FEAT15_Correction_on_top_of_286.docx - BWC: NFVSOL(20)000345_SOL003ed331_Addressing_ENs_and_RNs Editorials

Date	Version	Information about changes
		- aligning the description of "extensions" and "vnfConfigurableProperties" in tables 5.5.2.4-1, 5.5.2.6-1, 5.5.2.7-1, 5.5.2.11a-1.
May 2020	3.0.7	Contributions implemented - NBWCP: NFVSOL(20)000208r1_SOL003ed331_introducing_maps - BWC: NFVSOL(20)000280_SOL003ed331_mirror_of_42r5_VnfcResourceInfo_bugfix - BWC: NFVSOL(20)000335r3_SOL003ed331_FEAT15_VNF_state_snapshot_data_for_packaging - BWC: NFVSOL(20)000362r1_SOL003ed331_Forward_mirror_of_294_Guidelines_link_ports_noti - BWC: NFVSOL(20)000368_SOL003ed331_mirror_of_349_adding_PM_job_id_to_notification - BWC: NFVSOL(20)000401r1_SOL003ed331_Addressing_RN_on_CR_classification_issues - BWC: NFVSOL(20)000416r1_SOL003ed331_VnfExtCpData_EN_resolution - BWC: NFVSOL(20)000431r1_SOL003ed331_fixing_more_ENs_and_RNs - BWC: NFVSOL(20)000436r1_SOL003ed331_Forward_mirror_of_434_Correction_external_images - BWC: NFVSOL(20)000444_SOL003ed331_fixing_EN_on_ModificationsTriggeredByVnfPkgChang Editorials - Fixed an issue with the implementation of NFVSOL(20)000337 in table 5.4.23.3.1-2 - Replaced "(online)" and "(inline)" with "(inlined)".
June 2020	3.0.8	Contributions implemented: - BWC: NFVSOL(20)000439r2_SOL003ed331_FEAT02_address_ENs_on_metadata_during_ChangeCurr - NBWCP: NFVSOL(20)000210r2 SOL003ed331 ChangeExtVnfConnectivity using patch semantics - BWC: NFVSOL(20)000351r2_SOL003ed331_API_versions - BWC: NFVSOL(20)000407r3_SOL003ed331_Nokia_review_comments - BWC: NFVSOL(20)000430r1 SOL003ed331 update features annex - BWC: NFVSOL(20)000506_SOL003ed331_mirror_of_505_Further_changes_from_SOL016 - BWC: NFVSOL(20)000521r1_SOL003ed331_FEAT15_Correction_wrong_implementation - BWC: NFVSOL(20)000551_SOL003ed331_Feature_annex_resolving_EN - BWC: NFVSOL(20)000533r3_SOL003ed331_clarification_for_VNFD_content - BWC: NFVSOL(20)000555r1_SOL003ed331_Aligning_SOL003_V030008r3_with_SOL002 - BWC: NFVSOL(20)000559r1_SOL003ed331_mirror_of_558_FEAT15_Solving_issue_SOL010_normat - BWC: NFVSOL(20)000567_SOL003ed331_mirror_of_566_FEAT15_Issue_alignment_VnfSnapshot Editorials: - Replaced "New type" by "New resource data type" in Annex F to align with the agreed set of classifications as per NFVSOL(20)000509r1 - Fixed various small typos - Consistent use of terms "virtualised", "artifact", "JSON Merge Patch" - Removed redundant "see note" related to notification endpoints in the resource tables
August 2020	3.3.1	Version update for publication
October 2020	3.3.2	Contributions implemented: - BWC: NFVSOL(20)000686r1_SOL003ed341_FEAT15_Clarifications_about_snapshot_images_and_

Date	Version	Information about changes
		- BWC: NFVSOL(20)000697r1_SOL003ed341_FEAT15_Improve_the_description_in_VNF_LCM_and_Ad - BWC: NFVSOL(20)000714r1_SOL003ed341_Fixing_of_the_flows_of_updating_the_callback_URI
January 2021	3.3.3	Contributions implemented: - BWC: NFVSOL(20)000696_SOL003ed341_stage_3_mirror_of_NFVIFA_20_000626 - BWC: NFVSOL(20)000709r2_SOL003ed351_VipCp_related_changes_from_IFA_CRs_600__664_and_ - BWC: NFVSOL(20)000789_SOL003ed351_fix_identifier_datatypes_in_VnfExtCpInfo - BWC: NFVSOL(20)000800_SOL003ed351_mirror_of_748r6_Adding_Trunk_Logical_Topology_be Editorials: - Year changed to 2021
March 2021	3.3.4	Contributions implemented: - BWC: NFVSOL(20)000778r2_SOL003ed351_extManagedVirtualLinkInfo_clarification - BWC: NFVSOL(21)000053r2_SOL003ed351_mirror_of_0052r1_lcmcoord_API_LcmOpOcc_additions - BWC: NFVSOL(21)000056_SOL003ed351_packageSecurityOption_cardinality_fix_ - BWC: NFVSOL(21)000066r1_SOL003ed351_mirror_of_51_lcmcoord_API_hooks_with_other_GS_pa - BWC: NFVSOL(21)000071r1_SOL003ed351_update_FEAT_Annex_list_regading_ExtCp_changes - NBWC: NFVSOL(21)000074_SOL003ed351_add_ModificationsTriggeredByVnfPkgChange_to_excl - BWC: NFVSOL(21)000090r1_SOL003ed351_warnings_in_LcmOpOcc
April 2021	3.3.5	Contributions implemented: - BWC: NFVSOL(21)000207r1_SOL003ed351_Handling_of_security_sensitive_properties - BWC: NFVSOL(21)000236_SOL003ed351_granting_clarifications
April 2021	3.3.6	Contributions implemented: - BWC: NFVSOL(21)000183r2_SOL003ed351_notification_delivery_clarification
May 2021	3.3.7	Contributions implemented: - BWC:NFVSOL(21)000282r2_SOL003ed351_FEAT02_Mirror_of_182_Add_enumeration_values_of_L - BWC:NFVSOL(21)000297_SOL003ed351_Updating_API_versions_and_Feature_Annex - BWC:NFVSOL(21)000306r2_SOL003ed351_Nokia_review_comment
May 2021	3.3.8	Contributions implemented: - BWC: NFVSOL(21)000331_SOL003ed351_ChangeCurrentVnfPkg_related_terminology_fix - BWC: NFVSOL(21)000334r1_SOL003ed351_vnfdId_related_fixes_in_Granting_of_ChangeCurren
July 2021	3.5.1	Version update for publication

Date	Version	Information about changes
August 2021	4.0.1	Contributions implemented: - BWC: NFVSOL(21)000365_SOL003ed421_FEAT10_Mirror_of_363_Correction_about_multi-VIM_ - BWC: NFVSOL(21)000366_SOL003ed421_ENH02_05_Add_targetScaleLevelInfo_to_complement_ - BWC: NFVSOL(21)000367r2_SOL003ed421_FEAT17_Cloud-native_VNFs - BWC: NFVSOL(21)000401_SOL003ed421_mirror_of_400_Fix_editorial_mistakes_aligning_wi - BWC: NFVSOL(21)000442r1_SOL003ed421_Mirror_of_441_Editorial_fix_on_the_API_version_o
October 2021	4.0.2	Contributions implemented: - BWC: NFVSOL(21)000413_SOL003ed411_Rel4_mirror_of_412_Clarification_of_VIM_assets - BWC: NFVSOL(21)000415_SOL003ed411_Rel4_mirror_of_414_Use_of_old_assets_after_ChgCu - BWC: NFVSOL(21)000419r1_SOL003ed411_Rel4_mirror_of_418_and_476 - BWC: NFVSOL(21)000424_SOL003Ed421_FEAT017_Link_Ports_in_ExtVirtualLinkData - BWC: NFVSOL(21)000478_SOL003ed431_Rel_4_mirror_of_477r2_Granting_issue - BWC: NFVSOL(21)000492_SOL003ed431_FEAT17_improve_the_description_of_vimConnectionI
December 2021	4.0.3	Contributions implemented: - BWC: NFVSOL(21)000518r1_SOL003ed431_Rel_4_mirror_of_517r1_Conflicting_affinity_antia - BWC: NFVSOL(21)000558_SOL003ed431_Rel_4_mirror_of_557r2_vnfdId_in_resource_info_el - BWC: NFVSOL(21)000579r1_SOL003ed431_FEAT03_mirror_of_578r_Add_values_to_Alarm_for_s - BWC: NFVSOL(21)000597_SOL003ed431_Rel4_Mirror_of_596_changeCurrentVnfPkg_bugfix - BWC: NFVSOL(21)000612_SOL003ed431_Rel_4_mirror_of_585_cpConfigId_clarifications - BWC: NFVSOL(21)000624_SOL003ed431_Rel_4_mirror_of_586r2_fixing_ambiguous_note_in_G - BWC: NFVSOL(21)000632_SOL003ed431_Rel_4_mirror_of_631r1_CP_configuration_informati - BWC: NFVSOL(21)000661_SOL003ed431_Rel_4_mirror_of_660_AffectedExtLinkPort_bugfix - BWC: NFVSOL(21)000680_SOL003ed431_mirror_of_676_update_annex_aligning_with_ed361

Date	Version	Information about changes
February 2022	4.0.4	Contributions implemented: - BWC: NFVSOL(21)000684r1_SOL003ed431_FEAT17_Add_data_type_McioInfo_ - BWC: NFVSOL(21)000699r2_ENH02_04_SOL003ed431_Add_vnfdExtInvariantId_in_VnfPkgInfo - BWC: NFVSOL(22)000005_SOL003ed431_Mirror_of_675r1_Description_update_for_additiona - BWC: NFVSOL(22)000007_SOL003ed431_Mirror_of_587r1_merging_information_passed_in_th - BWC: NFVSOL(22)000033_SOL003ed431_Clarification_of_API_consumer_in_VnfExtCpData - BWC: NFVSOL(22)000040_SOL003ed431_FEAT17_Runtime_modelling_of_VirtualCp Editorials: - Adjusted the format of the note in clause 5.5.3.24. - Fixed the wrong letter capitalization in clause 5.5.4.9.
March 2022	4.0.5	Contributions implemented: - BWC: NFVSOL(22)000053r4_FEAT17_SOL003ed431_ResourceHandle - BWC: NFVSOL(22)000092_SOL003ed431_FEAT17_Add_data_type_VirtualCpInfo - BWC: NFVSOL(22)000118_SOL003ed431_FEAT17_Conveying_StorageClassName Editorials: - Replaced "Id" by "identifier" in clause 4.4.1.7. - Replaced "url" by "URL" in clause 5.5.3.26. - Adjusted the format of the "Permitted values" in clause 5.5.3.27.
April 2022	4.0.6	Contributions implemented: - BWC: NFVSOL(21)000482r4_SOL003ed431_NAD_id - BWC: NFVSOL(22)000139r2_FEAT17_SOL003ed431_CpProtocolData - BWC: NFVSOL(22)000163_FEAT17_SOL003ed431_SwImageIdentifier Editorials: - Replaced "extCP" by "external CP" in clauses 4.4.1.10b, 5.5.3.9b and 5.5.3.17. - Replaced "extCp" by "external CP" in clauses 4.4.1.10b and 5.5.3.9b.
June 2022	4.0.7	Contributions implemented: - BWC: NFVSOL(22)000199r1_SOL003ed431_Update_Feature_Annex_and_resolve_editor_s_notes - BWC: NFVSOL(22)000249_SOL003ed431_ENH02_05_Update_the_description_of_instantiation - BWC: NFVSOL(22)000265r1_SOL003ed431_Mirror_of_66r5_HTTP_header_fields_update_in_GET- - BWC: NFVSOL(22)000280_SOL003ed431_Update_API_versions Editorials: - Replaced "extCP" by "external CP" in clauses 4.4.1.11 and 5.4.11.1,

History

Document history		
V4.3.1	July 2022	Publication