**Agenda**

- **ETSI NFV Concepts and IFA Specifications**
  - NFV architectural framework
  - Main Management and Orchestration concepts

- **VNF lifecycle management: drilling down into IFA007, IFA008, and IFA011**
  - What is a VNF?
  - Packaging a VNF
  - Managing the VNF lifecycle

- **Conclusion**
PART 1
ETSI NFV CONCEPTS AND IFA SPECIFICATIONS
Network Functions Virtualisation: Management of NFV Components

Application & Functional Management

Functional Management of Network Service

Functional Management of (V)NFs

Functional Management of SW Instances

Virtualised Resources Management

Network Service (NS) Management

VNF Management
ETSI NFV Architecture, and NFV-MANO

(Specified in ETSI GS NFV-MAN 001)

**NFV Management & Orchestration**

- **Network Service Management**
  - Manage combinations of connected VNFs

- **VNF Management**
  - Manage individual VNFs

- **Virtual Resource Management**
  - Manage the use of NFVI resources

---

**NFV Orchestrator (NFVO)**

**VNF Manager (VNFM)**

**Virtualised Infrastructure Manager (VIM)**

**NFV Orchestration & Management (NFV-MANO)**

---

**OSS/BSS**

**EM**

**VNF**

**NFVI**

---

© ETSI 2017. All rights reserved
ETSI NFV IFA WG Group Specifications related to NFV-MANO interfaces and IM

http://docbox.etsi.org/ISG/NFV/Open/Drafts/
http://www.etsi.org/nfv
PART 2
VNF LIFECYCLE MANAGEMENT
VNF lifecycle management requires the VNF Descriptor (IFA011) and a number of interfaces defined in IFA007 and IFA008.

(*) not all operations
What is a VNF?
A look inside

VNF instance uses virtualized resources (Compute, Networking, Storage)

- VNF Components (VNFCs)
  - Internal components of a VNF, each providing a part of the functionality of a VNF (i.e. a part of the VNF’s application software).
  - Each VNFC instance maps 1:1 to a “Virtualization Container (VC)” (typically a Virtual Machine).
  - Resources are: Virtualised Compute plus optional Virtualised Storage.
- Internal virtual links (VLs, virtual networking), interconnecting the VNFCs via internal Connection Points (CPs).

A VNF provides external connection points

- external CPs allow to connect the VNF via external VLs to other virtual or physical network functions.
- external VLs are part of the Network Service, not of the VNF.
Packaging a VNF:

**VNF Package**

- The **VNF Package** contains:
  - the **VNF descriptor (VNFD)** that defines metadata for package onboarding and VNF management,
  - the **software images** needed to run the VNF, and
  - (optional) **additional files** to manage the VNF (e.g. scripts, vendor-specific files, etc.).

- The VNF Package is **digitally signed** and delivered by the VNF provider as a whole.
  - The VNF Package is immutable (protected from modification).

- The VNF Package is **stored in a repository** by the NFVO.

- The VNF Package **can be accessed by VNFM**.

Reference:
- ETSI GS NFV-IFA 011
- ETSI GS NFV-SOL 004
Packaging a VNF: VNF Package identification and versioning

The VNFD in the VNF package contains a set of information elements which allow unique identification of a VNF package (as created by the VNF provider), and keeping track of VNF package versions.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vnfdId</td>
<td>Identifier of the VNFD and the associated VNF Package. This attribute shall be globally unique. It is also used in interfaces.</td>
</tr>
<tr>
<td>vnfProvider</td>
<td>Provider of the VNF and of the VNFD.</td>
</tr>
<tr>
<td>vnfProductName</td>
<td>Name to identify the VNF Product. Invariant for the VNF Product lifetime.</td>
</tr>
<tr>
<td>vnfSoftwareVersion</td>
<td>Software version of the VNF. This is changed when there is any change to the software that is included in the VNF Package.</td>
</tr>
<tr>
<td>vnfVersion</td>
<td>Identifies the version of the VNFD.</td>
</tr>
<tr>
<td>vnfProductInfoName</td>
<td>Human readable name for the VNF Product. Can change during the VNF Product lifetime.</td>
</tr>
<tr>
<td>vnfProductInfoDescription</td>
<td>Human readable description of the VNF Product. Can change during the VNF Product lifetime.</td>
</tr>
</tbody>
</table>

Reference: ETSI GS NFV-IFA 011
Packaging a VNF: VNF Descriptor (VNFD)

The VNFD defines VNF properties, such as:

- resources needed (amount and type of Virtual Compute, Storage, Networking),
- software metadata,
- connectivity:
  - External Connection Points (described via CP Descriptors, CPD).
  - Internal Virtual Links (described via VL Descriptors, VLD)
  - Internal Connection Points (described via CP Descriptors, CPD)
- lifecycle management behavior (e.g. scaling, instantiation),
- supported lifecycle management operations, and their configuration,
- supported VNF specific parameters, and
- affinity / anti-affinity rules.

The VNFD defines deployment flavours (size-bounded deployment configurations, e.g. related to capacity).

Reference:
- ETSI GS NFV-IFA 011
- ETSI GS NFV-SOL 001
Managing the VNF lifecycle: VNF runtime information

Based on the definitions in the VNFD, **VNF instances** can be created in the NFVI (aka cloud).

The runtime information of each VNF instance, **VnflInfo**, is managed by the VNFM.

The VnflInfo element includes information such as:

- VNF instance identifier, VNF instance state,
- scale status (current “size” of VNF),
- metadata (version info, pointer to VNFD and VNF package, vendor-specific metadata),
- virtualised resources used (Virtualised Compute, Storage, Network),
- list of VNFCs,
- configurable parameters,
- external connectivity (external VLs, external CPs), and
- connectivity to VIM(s) used to manage the resources of the VNF.
VNF lifecycle management operations can influence the allocation of virtualized resources to a VNF instance, and/or modify the state of the VNF instance.

The following VNF LCM operations are defined by ETSI GS NFV-IFA 007 and IFA 008. Support of certain operations by a concrete VNF may depend on the capabilities of the VNF itself (e.g., whether a VNF is “ scalable ”).

<table>
<thead>
<tr>
<th>Operation</th>
<th>Support by VNF</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instantiate VNF</td>
<td>Mandatory</td>
<td>Allocate virtualised resources, configure them, start the application, trigger configuration of the application.</td>
</tr>
<tr>
<td>Scale VNF</td>
<td>Optional</td>
<td>Change the amount of virtualised resources allocated to a VNF.</td>
</tr>
<tr>
<td>Query VNF</td>
<td>Mandatory</td>
<td>Obtain runtime information about the VNF instance (VnfInfo).</td>
</tr>
<tr>
<td>Terminate VNF</td>
<td>Mandatory</td>
<td>Terminate the VNF, and release the virtualised resources.</td>
</tr>
<tr>
<td>Change VNF flavour</td>
<td>Optional</td>
<td>Change the deployment flavor of the VNF, which typically includes changing the amount of virtualised resources, and the topology.</td>
</tr>
<tr>
<td>Heal VNF</td>
<td>Optional</td>
<td>Virtualisation-related corrective actions to repair a faulty VNF, and/or its VNFC instances and internal VNF Virtual Link(s).</td>
</tr>
<tr>
<td>Operate VNF</td>
<td>Optional</td>
<td>Start or stop the VNF software.</td>
</tr>
<tr>
<td>Modify VNF Info</td>
<td>Mandatory</td>
<td>Change certain items of the VNF runtime information (VnfInfo).</td>
</tr>
<tr>
<td>Auto-Scale and Auto-Heal</td>
<td>Optional</td>
<td>Variants of Scale VNF and Heal VNF, triggered automatically in the VNFM, by monitoring the VNF</td>
</tr>
</tbody>
</table>
Managing the VNF lifecycle: How VNF LCM operations work

- Typically, **LCM operations are long-running operations** (minutes, hours) → tracking is essential.

- Therefore, each individual VNF LCM operation occurrence
  - can be identified (for correlation),
  - has a status (e.g. ongoing, error, success) that can be queried.

- VNFM will **notify start and completion** of each operation
  - notification sent to subscribed functional blocks (e.g. NFVO, EM),
  - each notification identifies the affected VNF and applied operation occurrence,
  - “completion” notification contains information about the changes to the VNF’s consumption of virtualised resources (success case), and
  - “completion” notification communicates error information (error case).
Managing the VNF lifecycle: Lifecycle Operation Granting: Ask the Orchestrator!

NFVO is responsible for Network Service Orchestration and Resource Orchestration:
- VNFM has therefore need to obtain permission for LCM operations, and
- NFVO needs to tell the VNFM in which part of the NFVI (data center, zone) the resources can be allocated.

Solution: granting exchange.
Managing the VNF lifecycle: Putting it together (an example)

VNF LCM operation message sequence

1. InstantiateVnf
2. NOTIFY(start, InstantiateVnf)
3. GrantRequest(InstantiateVnf, resourcePlan, constraints)
4. GrantResponse(OK, configParams)
5. AllocateVirtNetworks
6. AllocateVirtCompute
7. AllocateVirtStorage
8. NOTIFY(result, InstantiateVnf, allocatedResources)
Managing the VNF lifecycle: Scaling a VNF

- **Basic idea:** Elasticity
  A VNF’s resource consumption (e.g. number of VNFCs) changes with load.
- **VNF scaling shall be non-service disruptive.**
- **Modes:**
  - Horizontal scaling (scale in/out) → Add/remove virtualised resources (e.g. VNFCs)
  - Vertical scaling (scale up/down) → Reconfigure the capacity / size of existing virtualised resources (e.g., VM flavor, storage size)
  - In the ETSI NFV current release only horizontal scaling of the VNFs is supported

**Scaling triggers**
- on demand (Scale VNF LCM operations), and
- automatically by the VNFM when certain performance figures cross a threshold.
Managing the VNF lifecycle: Scaling a VNF, the model

- **Requirement from VNF designs:**
  A VNF may be scaled in multiple independent aspects (e.g. scale database storage capacity independent from call processing capacity).

- **Scaling aspect**
  - Also known as “scaling dimension”.
  - Describes in an abstracted manner what “property” of the VNF to scale.
  - Each scale level of a scaling aspect defines a valid size of the VNF w.r.t that aspect.
  - Scaling takes place in discrete steps, changing the size from one level to another one.
  - Operating experience: E.g. slider model.
Managing the VNF lifecycle: Illustration of VNF instantiation and scaling

Scaling aspects abstract the internal composition of the VNF, allowing unified fine-granular control and a good operating experience.

Example VNF internal view: Groups of VNFCs

**External view: Scaling aspects**

- **DB**
  - 0,0 Initial increment (Instantiate)
  - DB=1
  - DB=2

- **Proc**
  - Proc=1
  - Proc=2
  - Proc=3
  - Proc=4

**Example**

- DB VNFC
- OAM VNFC
- Processing VNFC
- Processing Helper VNFC

© ETSI 2017. All rights reserved
Managing the VNF lifecycle:
Generic VNFM

- One VNFM for all VNFs (from multiple vendors).
- VNF-related Multivendor integration interface: Ve-Vnfm (IFA008).
- Provides standard basic management capabilities to all VNFs.
- Additionally, if required, VNF-specific management functionality is covered „lifecycle management scripts“ that are defined by the VNF vendor and included with the VNF package
  - Standardization of a universal scripting language for such scripts is future work.
  - Near-term Generic VNFM implementations may support a few existing scripting languages selected by the VNFM vendor, and VNF vendors have to adapt their VNFs to use one of the available languages.

Source: ETSI GS NFV-IFA 009
Managing the VNF lifecycle: VNFM specific to a set of VNFs

- A VNFM that can manage a particular set of VNFs.
- Typically, delivered by the VNF vendor together with the VNF.
- VNF-related multivendor integration interfaces: Vi-Vnfm (IFA006) towards VIM, Or-Vnfm (IFA007) towards NFVO. Interface towards VNF and EM may be proprietary or based on IFA008.
- Allows the VNF vendor to encapsulate in a VNFM particular VNF-specific, complex or advanced lifecycle management procedures.

NOTE: IFA-specified interfaces and operations do not differentiate whether the VNFM is generic or specific, and thus applicable to all types of VNFM.
CONCLUSION
Conclusion

This tutorial has ...

... provided an overview of the main NFV concepts, MANO architecture, interfaces and functional blocks, and the work items of the ETSI NFV IFA working group,

... outlined what a VNF is, how a VNF is structured and packaged, what the role and composition of the VNF descriptor is and how the descriptions in it relate to the management of the lifecycle of a VNF,

... introduced how the lifecycle of a VNF is managed by its VNF Manager, which are the main lifecycle management operations, and what the typical call flow of a lifecycle management operation looks like, and

... briefly touched the different deployment options of the VNF Manager.
More information:

NFV Technology Page (information)
http://www.etsi.org/nfv

NFV Portal (working area)
http://portal.etsi.org/nfv

NFV Proofs of Concept (information)
http://www.etsi.org/nfv-poc

NFV Plugtest (information & registration)
http://www.etsi.org/nfvplugtest

Open Area:

Drafts http://docbox.etsi.org/ISG/NFV/Open/Drafts/

BACKUP
IFA WG Scope

- Specification of functional requirements and Interfaces
- Specification of NFV Information Modeling
- Specification of NFV Descriptors
- Reporting and specification of Acceleration Use cases
- Reporting and specification of new features evaluating interface and architecture enhancements

Dec '13
- ETSI NFV Architectural Framework v1.1.1 published

Nov '14
- ETSI NFV MANO WG Closed
- ETSI NFV IFA WG created
- NFV Release 2 work starts

Jan '15
- ETSI NFV Architectural Framework v1.2.1 published
- MAN001 - ETSI NFV Management and Orchestration Report published

Apr '16
- First IFA WG Release 2 specifications published
- NFV Release 3 work starts

Sep '16
- Completion of Release 2 work on requirements, interfaces and information model

Nov '16
- Publication of IFA WG Release 2 specifications

Now
- Maintenance of IFA Release 2 specifications
- Completing reports about Release 3 features
Network Functions Virtualization: VNFs, NS and E2E Network Service

The NFV Idea:
Decoupling network functions functionality from infrastructure and relocating the network functions from dedicated appliances to pools of resources leveraging commodity-of-the-shelf (COTS) hardware.
Softwarization of the network enabling automation of deployment and operations.
NFVO – NFV Orchestrator

- Manages the lifecycle of NS
- Exposes NS lifecycle management interfaces to the OSS/BSS
- Sends NS lifecycle management notifications to the OSS/BSS
- Exposes virtualized resource management interfaces to the VNFM
- Sends virtualized resource management notifications to the VNFM
- Manages the VNF lifecycle via the interfaces exposed by the VNFM
- Manages virtualized resources via the interfaces exposed by the VIM.
VNFM – VNF Manager

- **VNF Manager (VNFM)** manages the lifecycle of VNFs.
- **VNF Manager (VNFM)** manages virtualized resources associated to the VNF it manages via the interfaces exposed by the VIM or NFVO.
- **VNF Manager (VNFM)** exposes VNF lifecycle management interfaces/APIs to the VNF, EM and NFVO.
- **VNF Manager (VNFM)** sends VNF lifecycle management notifications to the VNF, EM and NFVO.
- **VNF Manager (VNFM)** manages VNF initial configuration via the interfaces exposed by the VNF.
VIM – Virtualised Infrastructure Manager

- **OSS/BSS**
- **EM**
- **VNF**
- **NFVI**

**NFV Orchestrator (NFVO)**
- NFV Service Catalog
- VNF Catalog
- NFV Instances
- NFVI Resources

**VNF Manager (VNFM)**
- VNF
- NFV
- VNF Manager
- VNFM

**Virtualised Infrastructure Manager (VIM)**
- Vn-Nf
- Os-Ma-nfvo
- Ve-Vnfm-em
- Ve-Vnfm-vnf
- Vi-Vnfm
- Or-Vnfm
- Or-Vi
- Ni-Vi

- **Manages the NFV infrastructure resources** (compute, network and storage) in one or more NFVI-PoPs.
- **Exposes virtualized resource management interfaces/APIs to the VNFM and NFVO**
- **Sends virtualized resource management notifications to the VNFM and the NFVO**
IFA WG deliverables:
Acceleration Reports and Specifications

- GR IFA001 – Acceleration Overview & use Cases
- GS IFA002 – VNF acceleration interface specifications
- GS IFA003 – vSwitch Requirements
- GS IFA004 – Acceleration Management Aspects
- (In progress) GS IFA018 – Resource Management Acceleration
- (In progress) GS IFA019 – Acceleration Interface

GS: Group Specification (normative)
GR: Group Report (informative)
IFA WG deliverables:
Other Reports and Specifications

- GR IFA009 – Architectural Options
- GR IFA015 – NFV Information Model
- GR IFA016 – Papyrus Guidelines
- GR IFA017 – UML Modeling Guidelines
- (In progress) GR IFA020 – NFVO Decomposition Options (Release 3)
- (In progress) GR IFA021 – MANO and Automated Deployment (Release 3)
- (In progress) GR IFA022 – Multi Site Services (Release 3)
- (In progress) GR IFA023 – Policy Management in MANO (Release 3)
- (In progress) GR IFA024 – External Touchpoints related to NFV Information Model
- (In progress) GR IFA025 – Real-time/ultra-low latency aspects in NFV related to service and network handling
- (In progress) GS IFA026 – Architecture enhancements for security management
- (In progress) GS IFA027 – Performance measurements
- (In progress) GR IFA028 – Architecture options to support multiple administrative domains (Release 3)
- (In progress) GR IFA029 – Enhancements of the NFV architecture towards cloud native and PaaS (Release 3)
How is a VNF managed?
Two operations to scale a VNF

ScaleVnf: Incremental scaling
- Changes **one aspect** only at a time
- Semantics: apply a delta (how many steps) based on current level of aspect to scale
- Two options: Scale from the current position by one scaling step (1) or by multiple scaling steps (2)

ScaleVnfToLevel: Go to target
- Typically changes **multiple aspects** at once
- Semantics: Specify a new target in scaling space (where do you want to go)
- The target may be a pre-defined instantiation level (3) or any arbitrary target in scaling space (4)

Support for the different scaling modes can be defined by the VNF provider in the VNFD.

Most VNFs only support a subset of these.
How is a VNF managed? Different VNF instantiation sizes

Use case: Operator does not want to always instantiate the VNF at minimum size, but rather, e.g.
- At minimum size
- At some intermediate size(s)
- At maximum size

Solution: Instantiation level is a tool that allows the VNFM to instantiate different sizes, using the defined scaling space.