ETSI GR NFV-REL 012 V1.1.1 (2021-11)

Network Functions Virtualisation (NFV); Reliability; Report on availability and reliability under failure and overload conditions in NFV-MANO

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
DGR/NFV-REL012

Keywords
availability, MANO, NFV, robustness

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

This study assumes a fault management model as defined by 3GPP TS 32.111-1 [i.3], which in turn is based on Recommendation ITU-T X.733 [i.4].

This is done in consistency with ETSI GS NFV-IFA 031 [i.6].

Modal verbs terminology

In the present document "**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 reports on impacts of NFV-MANO failures and overload conditions, including human errors, on the availability and reliability of NFV-MANO. A set of use cases will be described and analysed which include interactions between NFV-MANO functional entities under such conditions and other functional blocks (VNF, EM, OSS, ...). Also situations are analysed, where availability is achieved by a system of collaborating NFV-MANO functional entities possibly provided by different vendors. As a result, recommendations for the requirements of an available and reliable NFV-MANO system will be derived.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

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 GR NFV 003: "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

[i.2] ETSI GS NFV-REL 001: "Network Functions Virtualisation (NFV); Resiliency Requirements".

[i.3] 3GPP TS 32.111-1 (V16.0.0): "Telecommunication management; Fault Management; Part 1: 3G fault management requirements".

[i.4] Recommendation ITU-T X.733: "Systems Management: Alarm reporting function".

[i.5] ETSI GR NFV-REL 011: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Report on NFV-MANO Software Modification".

[i.6] ETSI GS NFV-IFA 031 (V3.4.1): "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Requirements and interfaces specification for management of NFV-MANO".

[i.7] ETSI GS NFV-IFA 008 (V3.4.1): "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Ve-Vnfm reference point - Interface and Information Model Specification".

[i.8] IETF RFC 7230: "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

alarm: information about a specific condition requiring attention

NOTE: An alarm does or does not represent an error.

alarm notification: message used to report an alarm

error: discrepancy between a computed, observed, or measured value or condition and a true, specified, or theoretically correct value or condition

NOTE 1: Error is a consequence of a fault.

NOTE 2: See ETSI GS NFV-REL 001 [i.2].

failure: deviation of the service from fulfilling its functionality

NOTE: See ETSI GS NFV-REL 001 [i.2].

fault: adjudged or hypothesized cause of an error

NOTE: See ETSI GS NFV-REL 001 [i.2].

3.2 Symbols

Void.

3.3 Abbreviations

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

DDoS	Distributed Denial of Service
FE	Functional Entity
RU	Resource Unit
RUI	Resource Unit Instance
SU	Service User

4 Architectural overview

4.1 NFV-MANO architectural considerations

The internal architecture of an NFV-MANO functional entity is not visible to the external world and it can follow different architectural paradigms.

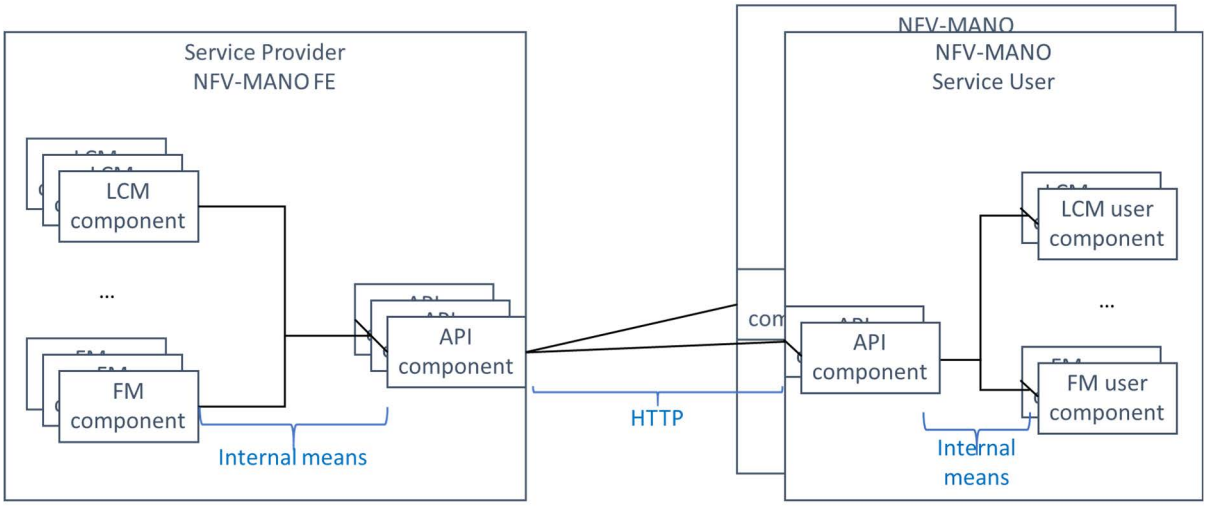

Figure 4.1-1: Example of the internal architecture of an NFV-MANO functional entity and the users of its services

One of the popular paradigms is the microservice based architecture according to which an NFV-MANO functional entity could be built as a set of different microservices. For example, an NFV-MANO functional entity can include a microservice implementing the Life Cycle Management (LCM) operations, another for Fault Management (FM) and yet another handling the HTTP API communication needs of the microservices as shown in figure 4.1-1. These different microservices are supported by different sets of components of the NFV-MANO functional entity to provide the NFV-MANO services in accordance with the ETSI GS NFV-IFA 031 [i.6]. An example of the Service Provider (SP) NFV-MANO functional entity could be a VNFM.

The same architectural considerations apply to the users of the NFV-MANO services provided by the NFV-MANO functional entity. Examples of the NFV-MANO Service User (SU) could be a VNF or the NFVO.

NOTE: The SP NFV-MANO functional entity is not aware of the internal structure of the NFV-MANO SU and vice versa. These details are shown and discussed for the purpose of the use case analysis.

When it comes to the reliability of the communication between these two categories of entities, i.e. the SP NFV-MANO functional entity and the NFV-MANO SU, two kinds of communication segments need to be considered. On the one hand, between the entities on the external portion of the communication path, HTTP is used as the communication protocol as defined in the ETSI NFV-SOL specifications. On the other hand, the means of internal communication - among the components of each of the entities - is left to the implementer (e.g. vendor) of each of these entities.

In addition, in the ETSI NFV specifications, two communication patterns are considered. The two-way communication pattern is implemented through the exchange of a request followed by a response. While in the one-way communication pattern (also referred as fire-and-forget), a message is sent without the need for follow-up.

This means that, in case of two-way communication, for example, when an LCM user component of the SU sends a request to an LCM component of the SP, the request passes through the SU-internal, the external and again on the SP-internal portions of the communication path between these components. The same applies in reverse order to the response. If the communication fails on any portion of this communication path, it can be detected by the SU LCM user component as it would not receive the response sent by the SP; and therefore, it can take actions as needed or sees appropriate.

In case of one-way communication, the sender, for example, an FM component of the SP, does not expect any response to the notification it sends. Nevertheless, the delivery of this notification to all intended receivers, i.e. FM user components of the SUs, is important to ensure that they can take any necessary actions. However, the sender of such communication - the FM component of the SP - has no way to detect if the notification was not delivered. Therefore, it is typically expected that the underlying communication mechanism guarantees the delivery to the receiving end(s) - FM user components of the SUs.

To this end, the HTTP protocol mandated for the external portion of the communication path does not cover the internal portions of the path, hence it cannot detect any loss occurring on the internal portions of the communication path.

With respect to the HTTP portion of the communication path itself, according to IETF RFC 7230 [i.8]:

"HTTP does not define specific error handling mechanisms except when they have a direct impact on security, since different applications of the protocol require different error handling strategies."

Also, HTTP allows for the chaining of connections through intermediaries, in which case the end-to-end delivery through this chain cannot always be guaranteed without appropriate error handling mechanism.

In clauses 5.2.3 and 5.2.4, different message loss scenarios and their mitigation are investigated through different use cases.

4.2 NFV-MANO functional entity redundancy

The internal architecture of an NFV-MANO functional entity is exposed only to the extent of enabling a network operator to manage the redundancy of the deployment of the NFV-MANO functional entity. For this purpose, the ETSI GR NFV-REL 011 [i.5] report has proposed a refinement to the concepts defined in the ETSI GS NFV-IFA 031 [i.6] specification. Accordingly, an NFV-MANO functional entity consists of one or more NFV-MANO functional entity Redundancy Unit(s)(RU). Each RU can be deployed redundantly according to a redundancy model. This redundancy model is one of the vendor defined redundancy models. An NFV-MANO functional entity redundancy unit might be further decomposed into NFV-MANO functional entity components. However, these NFV-MANO functional entity components are generally hidden from the network operator. If desired and available, the network operator can choose a redundancy model that deploys multiple RU instances.

To clarify these concepts that are essential for the understanding of use cases described in clause 5.2.2, figure 4.2-1 provides an example of the internal architecture of a deployed NFV-MANO functional entity.

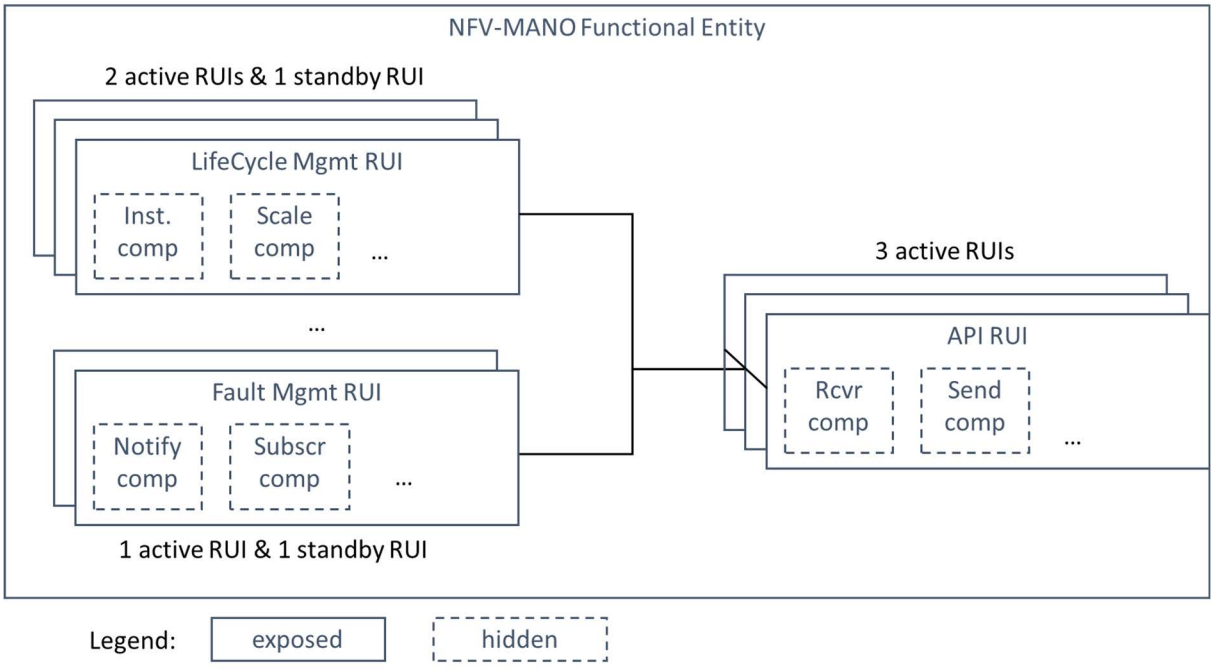

Figure 4.2-1: Example of the internal architecture of an NFV-MANO functional entity

The NFV-MANO functional entity in this example has three NFV-MANO functional entity redundancy units:

- The LifeCycle Mgmt redundancy unit (LifeCycle Mgmt RU) is deployed with two active and a standby Redundancy Unit Instances (RUIs). This is a selectable redundancy model for this redundancy unit specified by the vendor. The redundancy unit is further decomposed into components for handling the instantiation (Inst. comp), the scaling (Scale comp) and other services the LifeCycle Mgmt redundancy unit provides, which are not exposed. The components of the active and standby instances of the redundancy unit collaborate with each other to provide these services seamlessly in case of failures of components or an entire redundancy unit instance.

- The Fault Mgmt redundancy unit (Fault Mgmt RU) is deployed with one active and one standby instances. Again, this is one of the selectable redundancy models for this redundancy unit and accordingly the active and standby instances collaborate with each other. In the example, this redundancy unit consists of at least two components not visible for the network operator: one for handling the subscriptions (Subscr comp) and another for generating the notifications (Notify comp).

- The API Redundancy Unit (API RU) is deployed with three active instances. The redundancy unit is composed of the sender (Send comp) and the receiver (Rcvr comp) components. These RU instances do not collaborate with each other, meaning that if one of them fails the other RU instances will not have any information about the messages sent and received by the failed RU instance. But they will handle any new incoming and outgoing messages. The service thus remains available, but its continuity might not be guaranteed.

An NFV-MANO functional entity might include an internal availability management, which is capable of deploying the appropriate number of redundancy unit instances according to the redundancy model selected for instantiation. It would also monitor these instances and perform healing actions as they might become necessary. Note, however, that the internal availability management cannot detect and heal failures impacting the entire NFV-MANO functional entity. This requires an external manager.

It is also possible that there is no internal availability management, but, due to their need for collaboration, for example, the redundancy units or their components can detect that their instances have been deployed redundantly by an external manager. In this case, the redundancy units are able to report if there is any problem with their redundant peer. But since the life cycle of the redundancy unit instances is managed externally, the task of healing a failed redundancy unit instance also remains with this external manager. In addition, the external manager would need to monitor the health of the NFV-MANO functional entity redundancy unit instances if they cannot report each other's failure - for example because they are deployed all active without any need for collaboration other than sharing the load. Such external monitoring and management are also necessary to detect and heal a failure impacting the entire NFV-MANO functional entity.

Finally, it is possible that an entire NFV-MANO functional entity is deployed redundantly. In this case, the NFV-MANO functional entity instances are not aware of each other by default and the external manager should not only manage the life cycle of the NFV-MANO functional entity instances, but also facilitate their collaboration. This collaboration might be very limited and typically would be implemented by external means, e.g. via external database/file.

To achieve higher reliability and availability, the options above can be combined. That is, the internal and external availability/life cycle managers can be used in combination with each other, each responsible for a particular scope of management. For example, the internal availability manager would handle the internal failures of components and redundancy units of the NFV-MANO functional entity. While the external life cycle manager monitors the NFV-MANO functional entity as a whole and performs healing actions at the NFV-MANO functional entity level.

For certain NFV-MANO functional entities, geo-redundant deployment might be necessary. This could be achieved either by redundant deployment of the entire NFV-MANO functional entity or its redundancy units. The main difference is that when the NFV-MANO functional entity redundancy unit(s) is/are deployed redundantly, they still act together as a single NFV-MANO functional entity instance, as they all represent the same identity. When the entire NFV-MANO functional entity is deployed redundantly, each instance will have its own identity and the collaboration of these different instances does not go beyond any applicable interface specifications (e.g. Or-Or).

5 Use cases

5.1 Introduction

Clause 5 describes use cases for NFV-MANO failure and overload conditions. Two functions are introduced for the purpose to describe the use cases, the Alarm-Aggregator and the MANO-Monitor functions. The task of the Alarm-Aggregator function is to maintain an aggregated list of alarm conditions that exist in the NFV system, while the MANO-Monitor function is responsible to take actions towards resolving the root cause of an alarm.

The use cases do not make any assumption what entity or entities can play such roles. The roles could be fulfilled by an administrator or OSS, or they could be new functionalities offered by NFV-MANO.

5.2 NFV-MANO failures

5.2.1 NFV-MANO failure detection and reporting

5.2.1.1 Handling of an alarm reported by an NFV-MANO functional entity

5.2.1.1.1 Introduction and goal

An NFV-MANO functional entity may detect an internal error that prevents it from providing a service as specified. If this error cannot be recovered internally, it is a failure. This failure situation should be reported to other interested parties by sending an alarm notification. The NFV-MANO functional entity will track the state of this alarm by adding it to its own active alarm list.

When receiving an alarm notification, the Alarm-Aggregator will inform the registered entities to enable them to take precautions to mitigate the impact of the failure of the faulty NFV-MANO functional entity.

The MANO-Monitor will acknowledge the alarm notification and take over the responsibility to resolve the root cause of the alarm. The MANO-Monitor will maintain a list of active alarms that it has acknowledged. After resolving the root cause, the normal operation resumes.

NOTE: It cannot always be assumed, that an NFV-MANO functional entity is able to detect, that it cannot provide service as specified and report this. The NFV-MANO functional entity may not even be operational anymore. The use case of the detection ofa potential failure by an external entity is described in clause 5.2.1.2.

5.2.1.1.2 Actors and roles

Table 5.2.1.1.2-1 describes the use case actors and roles.

Table 5.2.1.1.2-1: Handling of an alarm reported by an NFV-MANO functional entity actors and roles

#	Role	Description
1	Faulty NFV-MANO functional entity	The entity that detects a failure on itself.
2	Registered entity	Entity that has registered with the Alarm-Aggregator to be informed in case of an alarm.
3	MANO-Monitor	Entity responsible to resolve the root cause of the alarm.
4	Alarm-Aggregator	Entity responsible for maintaining an aggregated list of alarm conditions in the NFV system. For this purpose, it has registered with the NFV-MANO functional entities to receive alarm notifications. The Alarm-Aggregator will forward notifications to registered entities.

5.2.1.1.3 Pre-conditions

Table 5.2.1.1.3-1 describes the use case pre-conditions.

Table 5.2.1.1.3-1: Handling of an alarm reported by an NFV-MANO functional entity pre-conditions

#	Pre-condition	Additional description
1	All NFV-MANO functional entities, the Alarm-Aggregator and the MANO-Monitor are running correctly	This includes the NFV-MANO functional entity that will become faulty.
2	The Alarm-Aggregator has registered with all the NFV-MANO functional entities to receive alarm notifications	
3	The MANO-Monitor has registered with the Alarm-Aggregator to receive alarm notifications	
4	All NFV-MANO functional entities have registered with the Alarm-Aggregator to be informed about alarms they are interested in	

5.2.1.1.4 Post-conditions

Table 5.2.1.1.4-1 describes the use case post-conditions.

Table 5.2.1.1.4-1: Handling of an alarm reported by an NFV-MANO functional entity post-conditions

#	Post-condition	Additional description
1	All NFV-MANO functional entities, the Alarm-Aggregator and the MANO-Monitor are running correctly	This includes the NFV-MANO functional entity that was faulty.

5.2.1.1.5 Flow description

Table 5.2.1.1.5-1 describes the use case flow.

Table 5.2.1.1.5-1: Handling of an alarm reported by an NFV-MANO functional entity flow description

#	Actor/Role	Action/Description
Begins when	Faulty NFV-MANO functional entity	The faulty NFV-MANO functional entity detects an internal error. This error prevents it from providing a service as specified, thus it is a failure. It cannot recover from this failure on its own. It creates an entry in its active alarm list.
Step 1	Faulty NFV-MANO functional entity -> Alarm-Aggregator	The faulty NFV-MANO functional entity sends an alarm notification to the Alarm-Aggregator.
Step 2	Alarm-Aggregator -> Registered entities	The Alarm-Aggregator creates an entry in its global list of active alarms and sends the alarm notification to the registered entities. This includes the MANO-Monitor.
Step 3	MANO-Monitor-> Faulty NFV-MANO functional entity	The MANO-Monitor acknowledges to the faulty NFV-MANO functional entity that it has received the alarm notification and the responsibility to recover from the failure is taken over. The faulty NFV-MANO functional entity can stop trying to recover from the failure locally and it does not need to send subsequent alarm notifications for the same failure if the state of the alarm is the same.
Step 4	Faulty NFV-MANO functional entity -> Alarm-Aggregator	The faulty NFV-MANO functional entity sends an updated alarm notification with ackState set to true to the Alarm-Aggregator.
Step 5	Alarm-Aggregator -> Registered entities	The Alarm-Aggregator updates its global list of active alarms and forwards the alarm notification to the registered entities. This includes the MANO-Monitor.
Step 6	MANO-Monitor	The MANO-Monitor takes the necessary actions to recover from the failure. This may include the involvement of other NFV-MANO functional entities, non NFV-MANO functional entities, or an administrator.
Step 7	MANO-Monitor-> Faulty NFV-MANO functional entity	The MANO-Monitor detects/is informed that the root cause of the failure of the faulty NFV-MANO functional entity was probably removed. It informs the faulty NFV-MANO functional entity about the potential removal of the root cause (see note 1).
Step 8	Faulty NFV-MANO functional entity -> MANO-Monitor	The faulty NFV-MANO functional entity confirms the message about the potential removal of the root cause.
Step 9	Faulty NFV-MANO functional entity -> Alarm-Aggregator	The faulty NFV-MANO functional entity checks that the root cause of the failure was removed. It sends an alarm cleared notification to the Alarm-Aggregator and marks the corresponding entry in its active alarm list accordingly (see note 2).
Ends when	Alarm-Aggregator -> Registered entities	The Alarm-Aggregator sends the alarm cleared notification to the registered entities. This includes the MANO-Monitor. It sets the state of corresponding entry in its global list of active alarms to cleared.
NOTE 1: The information may be transmitted by proposing a change of the perceived severity to cleared, similar to the EscalatePerSevRequest operation available in ETSI GS NFV-IFA 008 [i.7]. NOTE 2: If there is still a failure condition, an updated alarm notification is sent to the Alarm-Aggregator and the flow continues in Step 2.		

5.2.1.2 Detection of a failure of another NFV-MANO functional entity

5.2.1.2.1 Introduction and goal

An NFV-MANO functional entity can detect that another NFV-MANO functional entity might be in a failure situation if, for example, it receives from the other NFV-MANO functional entity an unexpected message.

5.2.1.2.2 Actors and roles

Table 5.2.1.2.2-1 describes the use case actors and roles.

Table 5.2.1.2.2-1: Detection of a failure of another NFV-MANO functional entity actors and roles

#	Role	Description
1	Faulty NFV-MANO functional entity	Entity not providing a service as specified.
2	Failure detecting NFV-MANO functional entity	Entity that detects that another NFV-MANO functional entity does not provide a service as expected.
3	Registered entity	Entity that has registered with the Alarm-Aggregator to be informed in case of an alarm.
4	MANO-Monitor	Entity responsible to resolve the root cause of alarms.
5	Alarm-Aggregator	Entity responsible for maintaining an aggregated list of alarm conditions in the NFV system. For this purpose, it has registered with the NFV-MANO functional entities to receive alarm notifications. The Alarm-Aggregator will forward notifications to registered entities.

5.2.1.2.3 Pre-conditions

Table 5.2.1.2.3-1 describes the use case pre-conditions.

Table 5.2.1.2.3-1: Detection of a failure of another NFV-MANO functional entity pre-conditions

#	Pre-condition	Additional description
1	All NFV-MANO functional entities, the Alarm-Aggregator and the MANO-Monitor are running correctly	This includes the NFV-MANO functional entity that will detect the failure.
2	The Alarm-Aggregator has registered with all the NFV-MANO functional entities to receive alarm notifications	
3	The MANO-Monitor has registered with the Alarm-Aggregator to receive alarm notifications	
4	All NFV-MANO functional entities have registered with the Alarm-Aggregator to be informed about alarms they are interested in	

5.2.1.2.4 Post-conditions

Table 5.2.1.2.4-1 describes the use case post-conditions.

Table 5.2.1.2.4-1: Detection of a failure of another NFV-MANO functional entity post-conditions

#	Post-condition	Additional description
1	All NFV-MANO functional entities, the Alarm-Aggregator and the MANO-Monitor are running correctly	This includes the NFV-MANO functional entity that was faulty.

5.2.1.2.5 Flow description

Table 5.2.1.2.5-1 describes the use case flow.

Table 5.2.1.2.5-1: Detection of a failure of another NFV-MANO functional entity flow description

#	Actor/Role	Action/Description
Begins when	Faulty NFV-MANO functional entity -> Failure detecting NFV-MANO functional entity	The failure detecting NFV-MANO functional entity receives a message from an NFV-MANO functional entity which it was not expecting.
Step 1	Failure detecting NFV-MANO functional entity -> Alarm-Aggregator	By receiving an unexpected message, the failure detecting NFV-MANO functional entity assumes that the sender NFV-MANO functional entity is faulty. Therefore, it raises an alarm and creates an entry in its active alarm list and sends an alarm notification to the Alarm-Aggregator.
Step 2	Alarm-Aggregator -> Registered entities	The Alarm-Aggregator creates an entry in its global list of active alarms and forwards the alarm notification to the registered entities. This includes the MANO-Monitor. It can also include the faulty NFV-MANO functional entity.
Step 3	MANO-Monitor-> Failure detecting NFV-MANO functional entity	The MANO-Monitor acknowledges that it has received the alarm notification from the failure detecting NFV-MANO functional entity about the faulty NFV-MANO functional entity. The failure detecting NFV-MANO functional entity does not need to send subsequent alarm notifications for the same failure other than updates, including clearing it when the failure is not present anymore.
Step 4	Registered entities	The registered entities take notice of the alarm notification. If possible and beneficial, the registered entities take precautions to mitigate the impact of the failure of the faulty NFV-MANO functional entity.
Step 5	Failure detecting NFV-MANO functional entity -> Alarm-Aggregator	The failure detecting NFV-MANO functional entity sends an updated alarm notification with ackState set to true to the Alarm-Aggregator.
Step 6	Alarm-Aggregator -> Registered entities	The Alarm-Aggregator updates the entry in the global list of active alarms and forwards the alarm notification to all registered entities. This includes the MANO-Monitor.
Step 7	MANO-Monitor	The MANO-Monitor takes the necessary actions to recover from the failure. This can include the involvement of other NFV-MANO functional entities, including the faulty NFV-MANO functional entity and/or the failure detecting NFV-MANO functional entity, non NFV-MANO functional entities or an administrator.
Step 8	MANO-Monitor -> Failure detecting NFV-MANO functional entity	The MANO-Monitor detects/is informed that the root cause of the failure of the faulty NFV-MANO functional entity was successfully removed. Accordingly, it proposes to the failure detecting NFV-MANO functional entity to clear the alarm.
Step 9	Failure detecting NFV-MANO functional entity -> MANO-Monitor	The failure detecting NFV-MANO functional entity confirms the reception of the clearing proposal.
Step 10	Failure detecting NFV-MANO functional entity	The failure detecting NFV-MANO functional detects that the problem is solved (see note).
Step 11	Failure detecting NFV-MANO functional entity -> Alarm-Aggregator	The failure detecting NFV-MANO functional removes the alarm from its active alarm list and sends the alarm clearing notification to the Alarm-Aggregator.
Ends when	Alarm-Aggregator -> Registered entities	The Alarm-Aggregator sends the alarm clearing notification to all registered entities. This includes the MANO-Monitor and could include the faulty NFV-MANO functional entity. The Alarm-Aggregator removes the entry from the global list of active alarms.
NOTE: If the failure condition persists, the flow continues at Step 1 with sending an updated alarm notification to the Alarm-Aggregator.		

5.2.1.3 Alarm escalation

5.2.1.3.1 Introduction and goal

The severity of an alarm can change (see clause 5.2.3.3.6 as an example). In this case, it should be assured that all entities that have registered for a notification about this alarm are aware of the severity change. This is especially important for the MANO-Monitor if the responsibility of resolving the alarm has been taken over by means of acknowledging the alarm.

In this use case, a reliable notification delivery is assumed. The case of a notification loss is discussed in the use case in clause 5.2.3.3.

5.2.1.3.2 Actors and roles

Table 5.2.1.3.2-1 describes the use case actors and roles.

Table 5.2.1.3.2-1: Alarm escalation actors and roles

#	Role	Description
1	Faulty NFV-MANO functional entity	
2	MANO-Monitor	Entity responsible to resolve the root cause of the alarm.
3	Alarm-Aggregator	Entity responsible for maintaining an aggregated list of alarm conditions in the NFV system. For this purpose, it has registered with the NFV-MANO functional entities to receive alarm notifications. The Alarm-Aggregator will forward notifications to registered entities.

5.2.1.3.3 Pre-conditions

Table 5.2.1.3.3-1 describes the use case pre-conditions.

Table 5.2.1.3.3-1: Alarm escalation pre-conditions

#	Pre-condition	Additional description
1	All NFV-MANO functional entities have registered with the Alarm-Aggregator to be informed about alarms they are interested in	
2	The MANO-Monitor has registered with the Alarm-Aggregator to receive alarm notifications	
3	The Alarm-Aggregator has registered with all the NFV-MANO functional entities to receive alarm notifications	
4	A faulty NFV-MANO functional entity has an alarm with a given severity in its active alarm list	Because of an alarm acknowledgement from the MANO-Monitor the NFV-MANO functional entity is not responsible for the resolution of the root cause of the alarm.
5	The Alarm-Aggregator has distributed the alarm information to all registered entities	This includes the MANO-Monitor.
6	The MANO-Monitor is aware of the alarm raised by the NFV-MANO functional entity	The MANO–Monitor has acknowledged the alarm. The required steps to resolve this alarm have been initiated.

5.2.1.3.4 Post-conditions

Table 5.2.1.3.4-1 describes the use case post-conditions.

Table 5.2.1.3.4-1: Alarm escalation post-conditions

#	Post-condition	Additional description
1	The faulty NFV-MANO functional entity has updated the severity of the alarm	The alarm that was present in its active alarm list at the beginning of the use case has an updated severity.
2	The Alarm-Aggregator has distributed the alarm information with the updated severity to all registered entities	This includes the MANO-Monitor.

5.2.1.3.5 Flow description

Table 5.2.1.3.5-1 describes the flow of the use case.

NOTE: It is assumed that the notification delivery is reliable.

Table 5.2.1.3.5-1: Alarm escalation flow description

#	Actor/Role	Action/Description
Begins when	Faulty NFV-MANO functional entity	The faulty NFV-MANO functional entity detects that the severity of an alarm differs from the one currently stated in the alarm attributes (see note 1).
Step 1	Faulty NFV-MANO functional entity -> Alarm-Aggregator	The faulty NFV-MANO functional entity updates the severity of the alarm and sends an alarm notification to the Alarm-Aggregator (see note 2). This alarm notification contains the alarm information element, including the changed perceivedSeverity attribute.
Step 2	Alarm-Aggregator -> Registered entities	The Alarm-Aggregator updates the entry in its global list of active alarms and forwards the alarm notification to the registered entities (see note 2). This includes the MANO-Monitor.
Ends when	MANO-Monitor	The MANO-Monitor processes the alarm notification. It is an implementation decision, whether recovery actions, started for the original alarm, can be continued or should be modified.
NOTE 1: An example is the repeated timeout when sending a notification as described in clause 5.2.3.3.6. NOTE 2: Without reliable notification delivery, the solution proposed by this flow is insufficient.		

5.2.2 NFV-MANO failure recovery

5.2.2.1 NFV-MANO functional entity internal failover

5.2.2.1.1 Introduction and goal

To provide service availability and continuity, an NFV-MANO functional entity can be deployed redundantly using two or more instances of its different NFV-MANO functional entity redundancy units as introduced in clause 4.2.

Whenever a failure happens to a redundant instance of an NFV-MANO functional entity redundancy unit, the NFV-MANO service instance(s) provided by the failed NFV-MANO functional entity redundancy unit instance is(are) failed over to the remaining healthy instance(s) of the NFV-MANO functional entity redundancy unit automatically by an appropriate availability management mechanism within the NFV-MANO functional entity. It is expected that this failover is performed transparently, that is, the consumers of the NFV-MANO service interface(s) - aka NFV-MANO service users - do not detect any change in their interaction with the NFV-MANO functional entity.

Depending on the cause of the failure, the NFV-MANO functional entity might or might not be able to repair the failed NFV-MANO functional entity redundancy unit instance on its own. Therefore, an external manager entity, a MANO-Monitor is considered to whom the NFV-MANO functional entity can report the need for external assistance. For example, in case of the failure of some resources, the NFV-MANO functional entity might not be able to repair the failed NFV-MANO functional entity redundancy unit instance(s) until it can obtain the appropriate resources from such an external manager to replace the failed ones.

As described in clause 4.2, the NFV-MANO functional entity may have different NFV-MANO functional entity redundancy units for its different NFV-MANO functional entity components, which may use different redundancy schemes appropriate for the functionality of the internal NFV-MANO functional entity component(s). It is assumed that these different NFV-MANO functional entity redundancy units are visible within the NFV-MANO functional entity, while the NFV-MANO functional entity components and their functionality within an NFV-MANO functional entity redundancy unit are typically not exposed.

NOTE: An alarm notification sent by the NFV-MANO functional entity is delivered to the MANO-Monitor via the Alarm-Aggregator. This routing is elaborated in the use case of clause 5.2.1.1. Therefore, to improve readability the detailed steps are omitted from the flows described in this clause 5.2.2.

5.2.2.1.2 Actors and roles

Table 5.2.2.1.2-1 describes the actors and roles of the use case.

Table 5.2.2.1.2-1: NFV-MANO functional entity internal failover actors and roles

#	Role	Description
1	NFV-MANO functional entity	NFV-MANO functional entity providing NFV-MANO services.
2	Service User	User of an NFV-MANO service interacting with the NFV-MANO functional entity. There could be more than one such Service Users.
3	MANO-Monitor	External manager entity which is capable of assisting the NFV-MANO functional entity in its internal recovery.
4	Alarm-Aggregator	Entity responsible for maintaining an aggregated list of alarm conditions in the NFV system. For this purpose, it has registered with the NFV-MANO functional entities to receive alarm notifications. The Alarm-Aggregator will forward notifications to registered entities.

5.2.2.1.3 Pre-conditions

Table 5.2.2.1.3-1 describes the pre-conditions of the use case.

Table 5.2.2.1.3-1: NFV-MANO functional entity internal failover pre-conditions

#	Pre-condition	Additional description
1	NFV-MANO functional entity running normally and providing NFV-MANO services	The NFV-MANO functional entity is deployed with redundant NFV-MANO functional entity redundancy units.
2	Service User operating normally	The service user interacts normally with the NFV-MANO functional entity using the NFV-MANO service interface.
3	The MANO-Monitor is registered with the Alarm-Aggregator and the Alarm-Aggregator is registered with NFV-MANO functional entities	Alarm notifications are routed through the Alarm-Aggregator, but the Alarm-Aggregator itself and, therefore, this routing of alarm notifications are not shown in the flow descriptions.
4	MANO-Monitor operating normally	

5.2.2.1.4 Post-conditions

Table 5.2.2.1.4-1 describes the post-conditions of the use case.

Table 5.2.2.1.4-1: NFV-MANO functional entity internal failover post-conditions

#	Post-condition	Additional description
1	NFV-MANO functional entity running normally and providing NFV-MANO services	The NFV-MANO functional entity has restored its internal redundancy after the failure of an NFV-MANO functional entity redundancy unit instance.
2	Service User operating normally	The service user interacts normally with the NFV-MANO functional entity using the NFV-MANO service interface without being aware of the internal failure.
3	MANO-Monitor operating normally	All outstanding alarms related to the internal failure of the NFV-MANO functional entity have been cleared.

5.2.2.1.5 Flow description

The flow for the use case is described in table 5.2.2.1.5-1.

Table 5.2.2.1.5-1: NFV-MANO functional entity internal failover flow description

#	Actor/Role	Action/Description
Begins when	NFV-MANO functional entity	The internal availability management mechanism of the NFV-MANO functional entity detects that an instance of its NFV-MANO functional entity redundancy units failed.
Step 1	NFV-MANO functional entity -> MANO-Monitor	The NFV-MANO functional entity sends a notification to the MANO-Monitor that an error (i.e. internal failure) has occurred and recovery is in progress.
Step 2	NFV-MANO functional entity	The internal availability management mechanism of the NFV-MANO functional entity identifies the remaining healthy instance(s) of the NFV-MANO functional entity redundancy unit and activates it/them indicating as appropriate the NFV-MANO service instance(s) the failed instance was serving.
Step 3	Service User -> NFV-MANO functional entity	Any new interactions from the Service Users that were served by the failed NFV-MANO functional entity redundancy unit instance are routed to the activated instance(s) of the NFV-MANO functional entity redundancy unit.
Step 4	NFV-MANO functional entity	The internal availability management mechanism of the NFV-MANO functional entity attempts to repair the failed NFV-MANO functional entity redundancy unit instance.
Step 5	NFV-MANO functional entity -> MANO-Monitor	If the repair attempt in Step 4 was successful, go to Step 7. Otherwise, the NFV-MANO functional entity sends an alarm notification to the MANO-Monitor requesting assistance, e.g. requesting resources which can be used to restore the redundancy. For details on the alarm notification see clause 5.2.1.
Step 6	MANO-Monitor -> NFV-MANO functional entity	The MANO-Monitor resolves the alarm, e.g. by providing the needed resources to the NFV-MANO functional entity, which can return to Step 4.
Step 7	NFV-MANO functional entity -> MANO-Monitor	If the repair in Step 4 was successful, the NFV-MANO functional entity clears any related alarm notification that was sent to the MANO-Monitor and also sends a notification that the recovery from the error (i.e. internal failure) was completed.
Step 8	NFV-MANO functional entity	If necessary, the internal availability management mechanism of the NFV-MANO functional entity redistributes the roles and the load of the NFV-MANO service instance(s) among the instances of the NFV-MANO functional entity redundancy unit.
Ends when	NFV-MANO functional entity	The NFV-MANO service instances are provided, and the redundancy of the NFV-MANO functional entity has been restored to its initial level.

5.2.2.2 Externally managed failover of NFV-MANO functional entity redundancy units

5.2.2.2.1 Introduction and goal

In this clause, the use case of external handling of the failure of a redundant NFV-MANO functional entity redundancy unit is discussed. In this case, the NFV-MANO functional entity offering redundancy units could imply both: that different instances of the same NFV-MANO functional entity redundancy unit do not need to collaborate to preserve the service state (i.e. the NFV-MANO service is stateless), or that they do collaborate to protect the state of the services they are providing. In the latter case, the redundancy units of the NFV-MANO functional entity can detect themselves when they are deployed redundantly.

The MANO-Monitor instantiates the NFV-MANO functional entity redundancy units and monitors them subsequently to detect different kinds of failures. It also subscribes with the NFV-MANO functional entity for failure notifications. It is the external manager handling the failure of a redundant NFV-MANO functional entity redundancy unit.

5.2.2.2.2 Actors and roles

Table 5.2.2.2.2-1 describes the actors and roles of the use case.

Table 5.2.2.2.2-1: Externally managed failover of NFV-MANO functional entity redundancy units actors and roles

#	Role	Description
1	SP FE Redundancy Unit1	A Redundancy Unit (RU) of the Service Provider (SP) NFV-MANO Functional Entity (FE), which provides and protects the NFV-MANO services together with SP FE Redundancy Unit2.
2	SP FE Redundancy Unit2	A redundancy unit of the Service Provider (SP) NFV-MANO Functional Entity (FE), which provides and protects the NFV-MANO services together with SP FE Redundancy Unit1.
3	Service User	NFV-MANO service user using the NFV-MANO services provided by the Service Provider FE NFV-MANO functional entity as a whole. There could be more than one such Service Users.
4	MANO-Monitor	External manager entity, which is managing the life cycle of the redundancy units (SP FE Redundancy Unit1 and SP FE Redundancy Unit2) of the Service Provider NFV-MANO functional entity, monitors their health and is registered to receive relevant notifications for the Service Provider NFV-MANO functional entity.
5	Alarm-Aggregator	Entity responsible for maintaining an aggregated list of alarm conditions in the NFV system. For this purpose, it has registered with the NFV-MANO functional entities to receive alarm notifications. The Alarm-Aggregator will forward notifications to registered entities.

5.2.2.2.3 Pre-conditions

Table 5.2.2.2.3-1 describes the pre-conditions of the use case.

Table 5.2.2.2.3-1: Externally managed failover of NFV-MANO functional entity redundancy units pre-conditions

#	Pre-condition	Additional description
1	SP FE Redundancy Unit1 and SP FE Redundancy Unit2 running normally and providing NFV-MANO services	The NFV-MANO functional entity is deployed with redundant instances of the NFV-MANO functional entity redundancy unit (SP FE Redundancy Unit1 and SP FE Redundancy Unit2), which provide and protect the NFV-MANO services.
2	Service User operating normally	The Service User interacts with the NFV-MANO functional entity as a whole to use the NFV-MANO services.
3	MANO-Monitor registered with the Alarm-Aggregator and Alarm-Aggregator registered with NFV-MANO functional entities	Alarm notifications are routed through the Alarm-Aggregator, but the Alarm-Aggregator itself and, therefore, this routing of alarm notifications are not shown in the flow descriptions.
4	MANO-Monitor operating normally	The MANO-Monitor has instantiated the NFV-MANO functional entity redundancy unit instances (SP FE Redundancy Unit1 and SP FE Redundancy Unit2). It monitors them and is also registered with the NFV-MANO functional entity to receive notifications.

5.2.2.2.4 Post-conditions

Table 5.2.2.2.4-1 describes the post-conditions of the use case.

Table 5.2.2.2.4-1: Externally managed failover of NFV-MANO functional entity redundancy units post-conditions

#	Post-condition	Additional description
1	Redundancy units of the NFV-MANO functional entity running normally and providing NFV-MANO services	The redundancy of the NFV-MANO functional entity has been restored after the failure of a redundancy unit instance. The restored redundancy unit (SP FE Redundancy Unit1) and the remaining one (SP FE Redundancy Unit2) together are providing and protecting the NFV-MANO services.
2	NFV-MANO service user operating normally	The Service User continues to interact with the NFV-MANO functional entity as a whole to use the NFV-MANO services. It was not, or was only minimally, impacted by the failure of one of the redundancy units.
3	MANO-Monitor operating normally	The MANO-Monitor has re-instantiated the failed NFV-MANO functional entity redundancy unit.

5.2.2.2.5 Flow description for collaborating NFV-MANO functional entity redundancy units

The first flow for the use case described in table 5.2.2.2.5-1 represents the case when the NFV-MANO functional entity redundancy units collaborate with each other to provide and protect the NFV-MANO service. Therefore, they can detect each other's failure and report this to the MANO-Monitor.

Table 5.2.2.2.5-1: Collaborating NFV-MANO functional entity redundancy units flow description

#	Actor/Role	Action/Description
Begins when	Service User -> SP FE Redundancy Unit1	One of the NFV-MANO functional entity redundancy units failed while it was serving a request of the Service User. The failed NFV-MANO functional entity redundancy unit was in collaboration with the other NFV-MANO functional entity redundancy unit, SP FE Redundancy Unit2.
Step 1	SP FE Redundancy Unit2	The other NFV-MANO functional entity redundancy unit detects that the first redundancy unit has failed and assumes the latest state it knows for the failed redundancy unit, which includes the request of the Service User.
Step 2	SP FE Redundancy Unit2 -> MANO-Monitor	The remaining redundancy unit, on behalf of the NFV-MANO functional entity, sends an alarm notification to the MANO-Monitor about the failure of the first redundancy unit (SP FE Redundancy Unit1) and then takes over its services including ongoing requests.
Step 3	MANO-Monitor -> SP FE Redundancy Unit1	The MANO-Monitor cleans up the failed SP FE Redundancy Unit1 then re-instantiates it.
Step 4	SP FE Redundancy Unit2 -> SP FE Redundancy Unit1	The second redundancy unit SP FE Redundancy Unit2 detects that the failed (first) redundancy unit SP FE Redundancy Unit1 has been repaired. The two redundancy units pair with each other and synchronize the state of NFV-MANO services they provide. SP FE Redundancy Unit1 receives back the assignment to serve the request of the Service User (see note).
Step 5	SP FE Redundancy Unit2 -> MANO-Monitor	On behalf of the NFV-MANO functional entity, SP FE Redundancy Unit2 clears the alarm it reported earlier to the MANO-Monitor in Step 2.
Ends when	SP FE Redundancy Unit1 -> Service User	The Service User receives the result of the request it has sent before the failure occurred (see note).
NOTE: If the assignment to serve the request of the Service User is not handled back in Step 4 to SP FE Redundancy Unit1, the flow ends when the result is provided back to the Service User by SP FE Redundancy Unit2.		

5.2.2.2.6 Flow description for externally monitored NFV-MANO functional entity redundancy units

The redundant NFV-MANO functional entity redundancy units are not always aware of each other, in which case they cannot detect each other's failure and need to be monitored externally by the MANO-Monitor. The flow described in table 5.2.2.2.6-1 presents such a case. In this case, the NFV-MANO functional entity redundancy units are also stateless otherwise they would be aware of each other through the state synchronization.

Table 5.2.2.2.6-1: Externally monitored NFV-MANO functional entity redundancy units flow description

#	Actor/Role	Action/Description
Begins when	Service User -> SP FE Redundancy Unit1	One of the NFV-MANO functional entity redundancy units failed while it was serving a request of the Service User.
Step 1	MANO-Monitor	The MANO-Monitor detects that the redundancy unit SP FE Redundancy Unit1 has failed.
Step 2	MANO-Monitor	If necessary, the MANO-Monitor directs all traffic to the remaining SP FE Redundancy Unit2.
Step 3	MANO-Monitor -> SP FE Redundancy Unit1	The MANO-Monitor cleans up the failed SP FE Redundancy Unit1 then re-instantiates it.
Step 4	MANO-Monitor	After successful re-instantiation of SP FE Redundancy Unit1, the MANO-Monitor re-directs back to SP FE Redundancy Unit1 any traffic redirected in Step 2.
Step 5	Service User -> SP FE Redundancy Unit1	The Service User resends its request since it has not received any response yet since the request was lost when SP FE Redundancy Unit1 has failed (see note).
Ends when	SP FE Redundancy Unit1 -> Service User	The Service User receives the result of the request it has sent before the failure occurred (see note).
NOTE: The Service User might resend its request while SP FE Redundancy Unit2 handles all traffic (i.e. before Step 4), in which case SP FE Redundancy Unit2 fulfils its request.		

5.2.2.3 Failover of NFV-MANO functional entities

5.2.2.3.1 Introduction

In this clause, the use case of handling the failure of complete NFV-MANO functional entities is discussed. An NFV-MANO functional entity can be deployed as a standalone instance or redundantly.

In the standalone case, if the NFV-MANO functional entity instance fails as a whole, a new instance of the NFV-MANO functional entity needs to be instantiated to deliver the NFV-MANO services the failed instance was providing. The new instance of the NFV-MANO functional entity is only able to continue the NFV-MANO services, such as ongoing operations that were requested, if the state for these NFV-MANO services (e.g. the state of the ongoing operations) has been stored externally before the old instance failed. This storage is external to the NFV-MANO functional entity instance, so that it is not impacted by the failure of the NFV-MANO functional entity instance. In this case, the new instance can read this state – including the state of the started operation - and, after verification, continue from it. The verification is needed to synchronize the stored state with any changes that occurred in the system while the NFV-MANO services were not available (e.g. if the service user requesting the operation is still present). Also, an external entity such as the MANO-Monitor is needed to detect the failure and manage the life cycle of the NFV-MANO functional entity instances.

If redundant NFV-MANO functional entity instances are deployed, they can share the load of providing an NFV-MANO service. In this case, each of them will provide a subset of the NFV-MANO service, for example, by executing a different set of operations from beginning to completion. The set of operations each NFV-MANO functional entity instance handles can be referred to as an NFV-MANO service instance. However, it is not expected that the different NFV-MANO functional entity instances are aware of: each other, or the NFV-MANO service instances provided by each other (e.g. the set of ongoing operations handled by the other(s)). Accordingly, if one of the NFV-MANO functional entity instances fails, the other instance(s) cannot detect the failure and cannot take over the NFV-MANO service instances (e.g. the set of ongoing operations) it provided. Hence, similarly to the previous case, the MANO-Monitor should detect the failure and recover the failed NFV-MANO functional entity instance, as well as, if desired, to orchestrate the failover of the NFV-MANO service instances to the remaining instance(s). Any service state (e.g. the state of the ongoing operations) needs to be saved on an external storage accessible to the remaining NFV-MANO functional entity instance(s) to be able to transfer this state as part of the failover.

NOTE: If the instances of an NFV-MANO functional entity are aware of each other and possibly collaborating with each other to protect the NFV-MANO service instances they are providing, then these instances are considered to be the instances of the redundancy unit of the NFV-MANO functional entity. This case is described in the clause 5.2.2.1.

In the flows of this use case, the MANO-Monitor instantiates the NFV-MANO functional entity instances and monitors them subsequently to detect different kinds of failures. An instance of the NFV-MANO functional entity stores the state associated with the NFV-MANO service instances externally (e.g. the state of the ongoing operations).

5.2.2.3.2 Actors and roles

Table 5.2.2.3.2-1 describes the actors and roles of the use case.

Table 5.2.2.3.2-1: Failover of NFV-MANO functional entities actors and roles

#	Role	Description
1	SP FE Instance1	An instance of the Service Provider (SP) NFV-MANO Functional Entity (FE) which provides an NFV-MANO service by executing a set of operation requests.
2	SP FE Instance2	Another instance of the Service Provider (SP) NFV-MANO Functional Entity (FE) which provides the same NFV-MANO service by executing another set of operation requests.
3	Service User	An NFV-MANO service user using the NFV-MANO service provided by SP FE Instance1 and SP FE Instance2.
4	MANO-Monitor	External manager entity which manages the life cycle of the NFV-MANO functional entity instances, monitors their health and is capable of interacting with them.
5	Alarm-Aggregator	Entity responsible for maintaining an aggregated list of alarm conditions in the NFV system. For this purpose, it has registered with the NFV-MANO functional entities to receive alarm notifications. The Alarm-Aggregator will forward notifications to registered entities.
6	External storage	Storage accessible to both SP FE Instance1 and SP FE Instance2, where any state information can be stored for the operations they are executing. External means that the life cycle of this storage is independent of the life cycle of both SP FE Instance1 and SP FE Instance2.

5.2.2.3.3 Pre-conditions

Table 5.2.2.3.3-1 describes the pre-conditions of the use case.

Table 5.2.2.3.3-1: Failover of NFV-MANO functional entities pre-conditions

#	Pre-condition	Additional description
1	SP FE Instance1 and SP FE Instance2 running normally and providing NFV-MANO service	The instances of the service provider NFV-MANO functional entity are not aware of each other.
2	Service User operating normally	At any time, each operation request of a service user is served by only one of the service provider NFV-MANO functional entity instances.
3	MANO-Monitor registered with the Alarm-Aggregator and Alarm-Aggregator registered with NFV-MANO functional entities	Alarm notifications are routed through the Alarm-Aggregator, but the Alarm-Aggregator itself and, therefore, this routing of alarm notifications are not shown in the flow descriptions.
4	MANO-Monitor operating normally	The MANO-Monitor has instantiated the instances of the NFV-MANO functional entity (SP FE Instance1 and SP FE Instance2). It monitors them, and is also capable of interacting with them.
5	External storage operating normally	The external storage is accessible to both SP FE Instance1 and SP FE Instance2.

5.2.2.3.4 Post-conditions

Table 5.2.2.3.4-1 describes the post-conditions of the use case.

Table 5.2.2.3.4-1: Failover of NFV-MANO functional entities post-conditions

#	Post-condition	Additional description
1	SP FE Instance1 and SP FE Instance2 running normally and providing NFV-MANO service	The failed SP FE Instance1 of the service provider NFV-MANO functional entity has been restarted.
2	Service User operating normally	At any time, each operation request of a service user is served by only one of the service provider NFV-MANO functional entity instances.
3	MANO-Monitor operating normally	The MANO-Monitor has re-instantiated the failed SP FE Instance1, it monitors SP FE Instance1 and SP FE Instance2 and it is also capable of interacting with them.
4	External storage operating normally	The external storage is accessible to both SP FE Instance1 and SP FE Instance2.

5.2.2.3.5 Flow description for recovering the service of a failed NFV-MANO functional entity

The first flow for the use case described in table 5.2.2.3.5-1 represents the case when a single instance of the NFV-MANO functional entity is deployed to provide the NFV-MANO service. This NFV-MANO service is represented in the flow by an operation request of a service user, the execution of which is interrupted by the failure of the NFV-MANO functional entity instance.

When the NFV-MANO functional entity instance fails, the MANO-Monitor detects the failure and restarts it. After restart, the new NFV-MANO functional entity instance continues the execution of the requested operation from the state the failed NFV-MANO functional entity instance stored on an external storage.

The NFV-MANO functional entity instance might be serving simultaneously multiple different operation requests from the same or different NFV-MANO service users.

Table 5.2.2.3.5-1: Recovering the service of a failed NFV-MANO functional entity flow description

#	Actor/Role	Action/Description
Begins when	Service User -> SP FE Instance1	The instance of the NFV-MANO functional entity SP FE Instance1 receives an operation request from Service User.
Step 1	SP FE Instance1 -> External storage	SP FE Instance1 creates an entry for the operation request of Service User and starts its execution while updating this entry as necessary during execution.
Step 2	SP FE Instance1	SP FE Instance1 fails while processing the operation request.
Step 3	MANO-Monitor	The MANO-Monitor detects that SP FE Instance1 has failed (see note).
Step 4	MANO-Monitor -> SP FE Instance1	The MANO-Monitor cleans up the failed SP FE Instance1 then re-instantiates it. If necessary, the MANO-Monitor passes to the new SP FE Instance1 the information necessary to recover the state of the operation request ongoing before the failure.
Step 5	SP FE Instance1 -> External storage	The new SP FE Instance1 reads the External storage to recover the state of the ongoing operation request of Service User.
Step 6	SP FE Instance1	The new SP FE Instance1 performs any necessary check to verify the state of the ongoing operation request, then continues with its execution.
Ends when	SP FE Instance1 -> Service User	The SP FE Instance1 sends the result of the execution to Service User. The result can indicate successful completion or failure to fulfil the request.
NOTE: The MANO-Monitor can use any means to monitor the health of the service provider NFV-MANO functional entity instance, e.g. health check.		

5.2.2.3.6 Flow description for recovering the service of a failed instance of the NFV-MANO functional entity among many

The flow described in table 5.2.2.3.6-1 presents the case when two instances of the NFV-MANO functional entity are deployed. These instances are redundant in terms of sharing the load of the NFV-MANO services they provide, but they are not aware of each other. The NFV-MANO service is represented in the flow by an operation request of a service user, the execution of which is interrupted by the failure of the serving NFV-MANO functional entity instance. Different operation requests are distributed between the redundant NFV-MANO functional entity instances forming different subsets referred to as different NFV-MANO service instances.

The MANO-Monitor detects the failure of such an NFV-MANO functional entity instance and assists in the recovery of the operation requests it was serving (i.e. the NFV-MANO service instance).

Table 5.2.2.3.6-1: Recovering the service of a failed instance of the NFV-MANO functional entity among many flow description

#	Actor/Role	Action/Description
Begins when	Service User -> SP FE Instance1	The instance of the NFV-MANO functional entity SP FE Instance1 receives an operation request from Service User.
Step 1	SP FE Instance1 -> External storage	SP FE Instance1 creates an entry for the operation request of Service User and starts its execution while updating this entry as necessary during execution.
Step 2	SP FE Instance1	SP FE Instance1 fails while processing the operation request.
Step 3	MANO-Monitor	The MANO-Monitor detects that SP FE Instance1 has failed (see note).
Step 4	MANO-Monitor -> SP FE Instance2	If necessary, the MANO-Monitor directs all traffic to the remaining SP FE Instance2. If SP FE Instance2 is capable of taking over ongoing operations, the MANO-Monitor provides it with the information necessary to recover the state of the operation request the SP FE Instance1 was serving before the failure.
Step 5	SP FE Instance2 -> External storage	If SP FE Instance2 is capable, it fetches from the External storage the state of the ongoing operation request served by the failed SP FE Instance1. SP FE Instance2 verifies the state and continues the execution of the operation request of Service User.
Step 6	MANO-Monitor -> SP FE Instance1	The MANO-Monitor cleans up the failed SP FE Instance1, then re-instantiates it. If necessary, the MANO-Monitor passes to the new SP FE Instance1 the information necessary to recover the state of the operation request ongoing before the failure if SP FE Instance2 was not capable of taking it over. In this case, SP FE Instance1 performs any necessary check to verify and update the state and continues the execution of the request of Service User.
Step 7	MANO-Monitor	After successful re-instantiation of SP FE Instance1, the MANO-Monitor restores the load sharing schema between SP FE Instance1 and SP FE Instance2.
Ends when	SP FE Instance1 or SP FE Instance2 -> Service User	The SP FE Instance1 or SP FE Instance2 sends the result of the execution to Service User. The result can indicate successful completion or failure to fulfil the request.
NOTE: The MANO-Monitor can use any means to monitor the health of the service provider NFV-MANO functional entity instance, e.g. health check.		

5.2.3 Failures in the interworking of NFV-MANO functional entities

5.2.3.1 Correlation of failures of NFV-MANO functional entities

5.2.3.1.1 Introduction and goal

Two different NFV-MANO functional entities might detect failures, which are related to each other. Both will generate an alarm notification according to the use case "Handling of an alarm reported by an NFV-MANO functional entity" (see clause 5.2.2.1).

The MANO-Monitor will take over the responsibility to resolve the root cause of the alarms. In this use case, the alarm notification of the first NFV-MANO functional entity reports the root cause of the alarm notification of the second NFV-MANO functional entity. After the analysis of the reported alarms, the MANO-Monitor establishes the correlation between the alarms and ensures that the first alarm is cleared before the second alarm.

As an example, the following scenario is considered: the NFVO issues a query image operation to the VIM. While processing this request, the VIM detects that the connection to the file storage is lost. This generates an alarm notification, which is sent to the MANO-monitor and which is processed according to the use case of clause 5.2.1. The VIM returns a failed operation result to the NFVO. As a result, the NFVO also sends an alarm notification to MANO-Monitor informing it about the inability to query information about the image. As long as the first alarm is not cleared, a query image operation will not succeed. Therefore, it is essential that a correlation can be, and is, established between the alarms and, accordingly, the alarm sent by the VIM is resolved first.

5.2.3.1.2 Actors and roles

Table 5.2.3.1.2-1 describes the use case actors and roles.

Table 5.2.3.1.2-1: Correlation of failures of NFV-MANO functional entities actors and roles

#	Role	Description
1	NFV-MANO functional entity 1	Entity that detects a failure.
2	NFV-MANO functional entity 2	Another entity that detects a failure.
3	MANO-Monitor	Entity responsible for handling of the alarms. To be able to do so, it has registered with the Alarm-Aggregator.
4	Alarm-Aggregator	Entity responsible for maintaining an aggregated list of alarm conditions in the NFV system. For this purpose, it has registered with the NFV-MANO functional entities to receive alarm notifications. The Alarm-Aggregator will forward notifications to registered entities.

5.2.3.1.3 Pre-conditions

Table 5.2.3.1.3-1 describes the use case pre-conditions.

Table 5.2.3.1.3-1: Correlation of failures of NFV-MANO functional entities pre-conditions

#	Pre-condition	Additional description
1	All NFV-MANO functional entities, the MANO-Monitor and the Alarm-Aggregator are running correctly	
2	The Alarm-Aggregator has registered with the NFV-MANO functional entities to receive alarm notifications	
3	The MANO-Monitor has registered with the Alarm-Aggregator to receive alarm notifications	

5.2.3.1.4 Post-conditions

Table 5.2.3.1.4-1 describes the use case post-conditions.

Table 5.2.3.1.4-1: Correlation of failures of NFV-MANO functional entities post-conditions

#	Post-condition	Additional description
1	All NFV-MANO functional entities, the MANO-Monitor and the Alarm-Aggregator are running correctly	

5.2.3.1.5 Flow description

Table 5.2.3.1.5-1 describes the use case flow.

Table 5.2.3.1.5-1: Correlation of failures of NFV-MANO functional entities flow description

#	Actor/Role	Action/Description
Begins when	NFV-MANO functional entity 2 -> NFV-MANO functional entity 1	NFV-MANO functional entity 1 receives an operation request from NFV-MANO functional entity 2 that requires accessing the connected storage.

#	Actor/Role	Action/Description
Step 1	NFV-MANO functional entity 1	NFV-MANO functional entity 1 detects that it cannot process the received operation because of the inability to access the connected storage.
Step 2	NFV-MANO functional entity 1 -> NFV-MANO functional entity 2	NFV-MANO functional entity 1 sends an 'operation failed' result for the requested operation, which raises an alarm in NFV-MANO functional entity 2.
Step 3a	NFV-MANO functional entity 1 -> Alarm-Aggregator	NFV-MANO functional entity 1 creates an alarm 1 in its active alarm list and sends an alarm notification to the Alarm-Aggregator (see note).
Step 3b	NFV-MANO functional entity 2 -> Alarm-Aggregator	NFV-MANO functional entity 2 creates an alarm 2 in its active alarm list and sends an alarm notification to the Alarm-Aggregator (see note).
Step 4a	Alarm-Aggregator -> MANO-Monitor	The Alarm-Aggregator creates an entry in the global list of active alarms for alarm 1 and sends the alarm notification to the MANO-Monitor and any other registered entity (see note).
Step 4b	Alarm-Aggregator -> MANO-Monitor	The Alarm-Aggregator creates an entry in the global list of active alarms for alarm 2 and sends the alarm notification to the MANO-Monitor and any other registered entity (see note).
Step 5a	MANO-Monitor -> NFV-MANO functional entity 1	The MANO-Monitor acknowledges alarm 1 of NFV-MANO functional entity 1. With this, it will take over the responsibility of resolving the issue (see note).
Step 5b	MANO-Monitor -> NFV-MANO functional entity 2	The MANO-Monitor acknowledges alarm 2 of the NFV-MANO functional entity 2. With this, it will take over the responsibility of resolving the issue (see note).
Step 6a	NFV-MANO functional entity 1 -> Alarm-Aggregator	NFV-MANO functional entity 1 sends an updated alarm notification for alarm 1 with ackState set to true to the Alarm-Aggregator (see note).
Step 6b	NFV-MANO functional entity 1 -> Alarm-Aggregator	NFV-MANO functional entity 2 sends an updated alarm notification for alarm 2 with ackState set to true to the Alarm-Aggregator (see note).
Step 7a	Alarm-Aggregator -> MANO-Monitor	The Alarm-Aggregator updates the entry in the global list of active alarms for alarm 1 and sends the alarm notification to the MANO-Monitor and any other registered entity (see note).
Step 7b	Alarm-Aggregator -> MANO-Monitor	The Alarm-Aggregator updates the entry in the global list of active alarms for alarm 2 and sends the alarm notification to the MANO-Monitor and any other registered entity (see note).
Step 8	MANO-Monitor	The MANO-Monitor correlates the received alarms and decides to handle first alarm 1.
Step 9	MANO-Monitor	The MANO-Monitor takes necessary actions to remove the root cause for alarm 1. This can include the involvement of other NFV-MANO functional entities, non-NFV-MANO functional entities or an administrator.
Step 10	MANO-Monitor -> NFV-MANO functional entity 1	The MANO-Monitor detects/is informed that the root cause of alarm 1 of NFV-MANO functional entity 1 was successfully removed. Accordingly, it proposes to NFV-MANO functional entity 1 to clear alarm 1.
Step 11	NFV-MANO functional entity 1 -> MANO-Monitor	NFV-MANO functional entity 1 confirms the reception of the clearing proposal and determines that the problem is solved and it is able to access the storage.
Step 12	NFV-MANO functional entity 1 -> Alarm-Aggregator	NFV-MANO functional entity 1 removes the alarm 1 from its active alarm list and sends the alarm clearing notification to the Alarm-Aggregator.
Step 13	Alarm-Aggregator -> MANO-Monitor	The Alarm-Aggregator removes the entry from the global list of active alarms for alarm 1 and sends the alarm clearing notification to the MANO-Monitor and any other registered entity.
Step 14	MANO-Monitor -> NFV-MANO functional entity 2	Knowing that NFV-MANO functional entity 1 has cleared its alarm that was the root cause of the alarm of NFV-MANO functional entity 2, the MANO-Monitor proposes to NFV-MANO functional entity 2 to clear alarm 2.
Step 15	NFV-MANO functional entity 2 -> MANO-Monitor	NFV-MANO functional entity 2 confirms the reception of the clearing proposal and proceeds with the operation request to the NFV-MANO functional entity 1.
Step 16	NFV-MANO functional entity 2 -> Alarm-Aggregator	When the operation succeeds after removing alarm 2 from its active alarm list, NFV-MANO functional entity 2 sends an alarm clearing notification for alarm 2.
Ends when	Alarm-Aggregator -> MANO-Monitor	The Alarm-Aggregator removes the entry from the global list of active alarms for alarm 2 and sends the alarm clearing notification to the MANO-Monitor and any other registered entity.
NOTE: Steps Xa and Xb occur in parallel.		

5.2.3.2 Communication failure between NFV-MANO functional entities

5.2.3.2.1 Introduction and goal

This use case describes the situation when a failure is detected in the communication between two NFV-MANO functional entities by receiving a timeout. In this case, one of the NFV-MANO functional entities requests an operation from the other unsuccessfully. This requestor NFV-MANO functional entity reports the error as an alarm via the Alarm-Aggregator, which in turn informs other registered entities about the alarm condition as well as the MANO-Monitor. The MANO-Monitor acknowledges the alarm and by that it takes the responsibility to resolve the alarm condition. The requestor NFV-MANO functional entity retries the operation only after it is informed about the removal of the fault causing the error in the communication.

5.2.3.2.2 Actors and roles

Table 5.2.3.2.2-1 describes the use case actors and roles.

Table 5.2.3.2.2-1: Communication failure between NFV-MANO functional entities actors and roles

#	Role	Description
1	NFV-MANO functional entity requesting an operation	The NFV-MANO functional entity that sends an operation request to another NFV-MANO functional entity.
2	Unreachable NFV-MANO functional entity	The NFV-MANO functional entity that cannot be reached due to a failure in the communication with another NFV-MANO functional entity.
3	MANO-Monitor	The entity responsible to resolve the root cause of alarms. To receive alarms, the MANO-Monitor registers with the Alarm-Aggregator.
4	Alarm-Aggregator	The entity responsible for maintaining an aggregated list of alarm conditions in the NFV system. For this purpose, it has registered with the NFV-MANO functional entities to receive alarm notifications. The Alarm-Aggregator also forwards the alarm notifications to registered entities according to their subscription.
5	Registered entities	Entities registered with the Alarm-Aggregator to receive alarm notifications of their interest from other NFV-MANO functional entities.

5.2.3.2.3 Pre-conditions

Table 5.2.3.2.3-1 describes the use case pre-conditions.

Table 5.2.3.2.3-1: Communication failure between NFV-MANO functional entities pre-conditions

#	Pre-condition	Additional description
1	All NFV-MANO functional entities, the Alarm-Aggregator and the MANO-Monitor are running correctly	This includes the NFV-MANO entity requesting an operation that will experience communication problems with another unreachable NFV-MANO functional entity.
2	The NFV-MANO functional entity requesting an operation has sent a request and started a timer	This NFV-MANO functional entity requesting an operation will encounter a problem with the requested operation indicated by the expiration of the timer.
3	All NFV-MANO functional entities have registered with the Alarm-Aggregator to be informed about alarms they are interested in	
4	The MANO-Monitor has registered with the Alarm-Aggregator to receive alarm notifications	
5	The Alarm-Aggregator has registered with all the NFV-MANO functional entities to receive alarm notifications	

5.2.3.2.4 Post-conditions

Table 5.2.3.2.4-1 describes the use case post-conditions.

Table 5.2.3.2.4-1: Communication failure between NFV-MANO functional entities post-conditions

#	Post-condition	Additional description
1	All NFV-MANO functional entities, the Alarm-Aggregator and the MANO-Monitor are running correctly	This includes the NFV-MANO entity requesting an operation that had experienced communication problems with another unreachable NFV-MANO functional entity.

5.2.3.2.5 Flow description

Table 5.2.3.2.5-1 describes the use case flow.

Table 5.2.3.2.5-1: Communication failure between NFV-MANO functional entities flow description

#	Actor/Role	Action/Description
Begins when	NFV-MANO functional entity requesting an operation	The NFV-MANO functional entity requesting an operation detects that the timer it started to safeguard its request has expired without receiving any answer to its request. This raises an alarm situation. Therefore, it creates an entry in its active alarm list.
Step 1	NFV-MANO functional entity requesting an operation -> Alarm-Aggregator	The NFV-MANO functional entity requesting an operation sends an alarm notification to the Alarm-Aggregator.
Step 2	Alarm-Aggregator -> Registered entities	The Alarm-Aggregator creates an entry in the global list of active alarms and forwards the alarm notification to all registered entities. This includes the MANO-Monitor and it could include the unreachable NFV-MANO functional entity giving it the possibility to remove any potential problem.
Step 3	MANO-Monitor-> NFV-MANO functional entity requesting an operation	The MANO-Monitor acknowledges the alarm and by that it takes the responsibility of resolving the alarm condition. The NFV-MANO functional entity requesting an operation does not need to send subsequent alarm notifications for the same failure other than changes and clearing it when appropriate.
Step 4	NFV-MANO functional entity requesting an operation -> Alarm-Aggregator	The NFV-MANO functional entity requesting an operation sends an updated alarm notification with ackState set to true to the Alarm-Aggregator.
Step 5	Alarm-Aggregator-> Registered entities	The Alarm-Aggregator updates the entry in the global list of active alarms and forwards the alarm notification to all registered entities. This includes the MANO-Monitor.
Step 6	MANO-Monitor	The MANO-Monitor takes the necessary actions to recover from the failure. The possible root cause can be in the communication system or in the unreachable NFV-MANO functional entity.
Step 7	MANO-Monitor-> NFV-MANO functional entity requesting an operation	The MANO-Monitor detects/is informed that the root cause of the communication problem was successfully removed. Accordingly, it proposes to the NFV-MANO functional entity requesting an operation to clear the alarm.
Step 8	NFV-MANO functional entity requesting an operation -> MANO-Monitor	The NFV-MANO functional entity requesting an operation confirms the reception of the clearing proposal.
Step 9	NFV-MANO functional entity requesting an operation	The NFV-MANO functional entity requesting an operation resends its operation request (see note).
Step 10	NFV-MANO functional entity requesting an operation	The NFV-MANO functional entity requesting an operation receives the expected answer to its request.
Step 11	NFV-MANO functional entity requesting an operation -> Alarm-Aggregator	The NFV-MANO functional entity requesting an operation removes the alarm from its active alarm list and sends the alarm clearing notification to the Alarm-Aggregator.

#	Actor/Role	Action/Description
Ends when	Alarm-Aggregator -> Registered entities	The Alarm-Aggregator sends the alarm clearing notification to all registered entities. This includes the MANO-Monitor and could include the unreachable NFV-MANO functional entity. The Alarm-Aggregator removes the entry from the global list of active alarms.
NOTE: This includes the start of a new timer. If the failure condition persists (the timeout happens again), an updated alarm notification is sent to the Alarm-Aggregator and the flow continues with Step 1.		

5.2.3.3 Notifications delivery by an NFV-MANO functional entity

5.2.3.3.1 Introduction and goal

The ETSI NFV-IFA interface specifications describe the different types of notifications used in NFV systems. These are messages which are not expected to be followed up by any receipt of delivery. If such a notification is not delivered to an intended receiving entity, this might happen without receiving any information about the failure. Depending on the type of the notification, the importance of such a loss can vary. In case the notification is associated with an action which is triggered by its reception, the failure of the notification delivery needs to be detected so that appropriate actions can be executed, e.g. resending the message. An example for a notification whose loss needs to be detected is an alarm notification.

NOTE 1: Entities of non-NFV-MANO functional blocks can also send notifications to the NFV-MANO. These notifications can also be lost. However, entities of non-NFV-MANO functional blocks currently do not send alarm notifications to the NFV-MANO.

According to the internal architecture described in figure 4.1-1, the failure can happen at different places. The flows of the present use case address these different scenarios after presenting the successful delivery.

Any detected failure in the notification delivery creates an alarm. The details of alarm handling are discussed in the clause 5.2.1.1.

NOTE 2: Since alarms are themselves reported as notifications, the issue causing the failure of the notification delivery may also prevent sending the alarm notification. It is expected that the FM component sending the alarm notification will write a log entry to assure that the information about the failure is preserved.

NOTE 3: The service component of the SP NFV-MANO functional entity cannot distinguish a successful notification delivery from an unsuccessful delivery for which no failure is reported back. If it is important that the notification is delivered (e.g. alarm notifications that need to be handled to ensure correct NFV operation), the loss should be detected at a higher protocol layer (e.g. by supervising the receipt of a corresponding AcknowledgeAlarmsRequest from the NFV-MANO Service User).

5.2.3.3.2 Actors and roles

Table 5.2.3.3.2-1 describes the use case actors and roles.

Table 5.2.3.3.2-1: Notifications delivery by an NFV-MANO functional entity actors and roles

#	Role	Internal Actor	Description
1	SP NFV-MANO functional entity	Service component of the SP NFV-MANO functional entity	A component of an NFV-MANO functional entity that provides an NFV-MANO service. As part of this service, the component needs to send a notification.
2		API component of the SP NFV-MANO functional entity	The component within the service provider NFV-MANO functional entity responsible for transferring messages to the service users.
3	NFV-MANO SU	API component of the NFV-MANO SU	The component within the NFV-MANO service user receiving the messages from the API component of the service provider NFV-MANO functional entity.
4		Service component of the NFV-MANO SU	A service component of the NFV-MANO service user which has registered to receive notifications on behalf of the NFV-MANO service user.

5.2.3.3.3 Pre-conditions

Table 5.2.3.3.3-1 describes the use case pre-conditions.

Table 5.2.3.3.3-1 Notifications delivery by an NFV-MANO functional entity pre-conditions

#	Pre-condition	Additional description
1	SP NFV-MANO functional entity and NFV-MANO Service User(s) running and operating correctly	SP NFV-MANO functional entity whose service component will send a notification and NFV-MANO service user(s) whose service component(s) is/are the receiver(s) of this notification.
2	NFV-MANO Service User(s) registered to receive notifications	

5.2.3.3.4 Post-conditions

Table 5.2.3.3.4-1 describes the use case post-conditions.

Table 5.2.3.3.4-1: Notifications delivery by an NFV-MANO functional entity post-conditions

#	Post-condition	Additional description
1	SP NFV-MANO functional entity and NFV-MANO service user(s) are running	This includes the SP NFV-MANO functional entity whose service component has sent a notification and NFV-MANO service user(s) that were registered to receive notifications and have been addressed to receive this notification.
2	The notification has been delivered or failures have been reported	The result of the notification sending differs in the different flows. This may be a successful delivery or a failure.

5.2.3.3.5 Flow description of a successful notification delivery

Table 5.2.3.3.5-1 describes the flow of a notification being delivered successfully to the service component of the NFV-MANO SU.

Table 5.2.3.3.5-1: Successful notification delivery flow description

#	Actor/Role	Action/Description
Begins when	Service component of the SP NFV-MANO functional entity	A service component of the SP NFV-MANO functional entity detects that it should send a notification to the NFV-MANO SU registered to receive notifications (see note).
Step 1	Service component of the SP NFV-MANO functional entity -> API component of the SP NFV-MANO functional entity	The request to send the notification is transferred to the API component. If the transfer is not successful, the service component of the SP NFV-MANO functional entity detects this and performs vendor specific actions. This can include resending the notification or writing a log entry or both.
Step 2	API component of the SP NFV-MANO functional entity -> API component of the NFV-MANO SU	The API component of the SP NFV-MANO functional entity sends the notification to the API component of the NFV-MANO SU. The protocol to be used is specified by the ETSI NFV stage 3 specifications. The current specifications use a HTTP POST request to send notifications.
Step 3	API component of the SP NFV-MANO functional entity	The API component of the SP NFV-MANO functional entity starts a timer to supervise the delivery of the HTTP POST request.
Step 4	API component of the NFV-MANO SU -> API component of the SP NFV-MANO functional entity	The API component of the NFV-MANO SU sends a HTTP 204 to confirm that the notification has been successfully delivered to the NFV-MANO SU.
Step 5	API component of the NFV-MANO SU -> Service component of the NFV-MANO SU	The API component of the NFV-MANO SU sends the notification to the service component of the NFV-MANO SU. If the transfer is not successful, the API component of the NFV-MANO SU detects this and performs vendor specific actions. This can include resending the notification or writing a log entry or both.

#	Actor/Role	Action/Description
Ends when	API component of the SP NFV-MANO functional entity	The API component of the SP NFV-MANO functional entity cancels the timer that was started in Step 3 to supervise the successful delivery of the notification.
NOTE: There may be multiple entities registered to receive a notification. To each of them, the notification will be sent.		

5.2.3.3.6 Flow description of a timeout in delivering the notification to the NFV-MANO service user API component

Table 5.2.3.3.6-1 describes the flow where the timer supervising the notification sent to the API component of the NFV-MANO SU expires.

Table 5.2.3.3.6-1: Timeout in delivering the notification to the NFV-MANO service user API component flow description

#	Actor/Role	Action/Description
Begins when	Service component of the SP NFV-MANO functional entity	A service component of the SP NFV-MANO functional entity detects that it should send a notification to the NFV-MANO SU registered to receive notifications (see note 2).
Step 1	Service component of the SP NFV-MANO functional entity -> API component of the SP NFV-MANO functional entity	The request to send the notification is transferred to the API component. If the transfer is not successful, the service component of the SP NFV-MANO functional entity detects this and performs vendor specific actions. This can include resending the notification or writing a log entry or both.
Step 2	API component of the SP NFV-MANO functional entity -> API component of the NFV-MANO SU	The API component of the SP NFV-MANO functional entity sends the notification to the API component of the NFV-MANO SU. The protocol to be used is specified by the ETSI NFV stage 3 specifications. The current specifications use HTTP POST request to send notifications (see note 4).
Step 3	API component of the SP NFV-MANO functional entity	The API component of the SP NFV-MANO functional entity starts a timer to supervise the delivery of the HTTP POST request.
Step 4	API component of the SP NFV-MANO functional entity	After the expiration of the timer started in Step 3, the API component of the SP NFV-MANO functional entity still has not received any response from the NFV-MANO SU (see note 1). If this is the first try, an alarm with severity 'warning' is created, if this is the second try the severity of the alarm is escalated to 'minor' (see note 3).
Step 5	API component of the SP NFV-MANO functional entity	If there was a first timer expiration in Step 4, the flow continues immediately with the resending of the notification in Step 2. Otherwise, the resending is postponed until it is indicated that the alarm condition might have been resolved.
Ends when	API component of the SP NFV-MANO functional entity	The API component of the SP NFV-MANO functional entity receives an HTTP 204 that confirms that the notification was delivered. This may be because of a repair action that was triggered by the alarm sent in Step 4. With this, the alarm condition is cleared. The API component of the SP NFV-MANO functional entity cancels the timer that was started in Step 3 to supervise the successful delivery of the notification.
NOTE 1: The timer expiry does not imply that the notification has not been delivered. If it has been delivered, the receiver of the notification needs to be able to detect a duplicate message caused by resending of the notification. NOTE 2: There may be multiple entities registered to receive a notification. To each of them, the notification will be sent. NOTE 3: For details of the alarm handling procedure and the notifications involved, see clause 5.2.1. NOTE 4: ETSI NFV-SOL specifications of NFV release 4.		

5.2.3.3.7 Flow description where an error code is received by the SP API component indicating an unsuccessful delivery

Table 5.2.3.3.7-1 describes the flow where an error code is received by the SP API component indicating an unsuccessful delivery.

Table 5.2.3.3.7-1: Error code received by the SP API component indicating an unsuccessful delivery flow description

#	Actor/Role	Action/Description
Begins when	Service component of the SP NFV-MANO functional entity	A service component of the SP NFV-MANO functional entity detects that it should send a notification to the NFV-MANO SU registered to receive notifications (see note 1).
Step 1	Service component of the SP NFV-MANO functional entity -> API component of the SP NFV-MANO functional entity	The request to send the notification is transferred to the API component. If the transfer is not successful, the service component of the SP NFV-MANO functional entity detects this and performs vendor specific actions. This can include resending the notification or writing a log entry or both.
Step 2	API component of the SP NFV-MANO functional entity -> API component of the NFV-MANO SU	The API component of the SP NFV-MANO functional entity sends the notification to the API component of the NFV-MANO SU. The protocol to be used is specified by the ETSI NFV stage 3 specifications. The current specifications use a HTTP POST request to send the notification.
Step 3	API component of the SP NFV-MANO functional entity	The API component of the SP NFV-MANO functional entity starts a timer to supervise the delivery of the HTTP POST request.
Step 4	-> API component of the SP NFV-MANO functional entity	The API component of the SP NFV-MANO functional entity receives a HTTP status code different from 204. This indicates that there was a problem during the delivery of the notification.
Step 5	API component of the SP NFV-MANO functional entity	The API component of the SP NFV-MANO functional entity cancels the timer that was started in Step 3 to supervise the successful delivery of the notification.
Ends when	API component of the SP NFV-MANO functional entity	Depending on the HTTP status code received in Step 4, the API component of the SP NFV-MANO functional entity takes the action specified by the ETSI NFV stage 3 specifications. This may include the resending of the notification or creating an alarm or both (see notes 2, 3 and 4).
NOTE 1: There may be multiple entities registered to receive a notification. To each of them, the notification will be sent.		
NOTE 2: For details of the alarm handling procedure, see clause 5.2.1.		
NOTE 3: An alarm is potentially created and its parameters depend on the received status code.		
NOTE 4: The decision to resend depends on the HTTP return code. There may be additional actions necessary before resending the notification (e.g. acquire authorization in case of receiving a 401 return code).		

5.2.4 Failures in the interworking of NFV-MANO functional entities with non-MANO functional blocks

5.2.4.1 Communication with an entity of a non-NFV-MANO functional block

5.2.4.1.1 Introduction and goal

A failure can occur in the communication of an NFV-MANO functional entity with an entity of a non-NFV-MANO functional block, for instance, with an EM.

For example, the entity of a non-NFV-MANO functional block can request an operation from the NFV-MANO functional entity. When the NFV-MANO functional entity tries to return the result of the operation requested by the requestor entity, two cases can be considered. For short operations, the NFV-MANO functional entity returns the result in the response to the request: for example, for a create VNF identifier request, the identifier is returned. For long operations, such as the VNF instantiation, in the response, an operation occurrence id is returned first to the requestor, and the final result of the operation is returned later in a notification, e.g. LCM operation occurrence notification.

When considering these exchanges at the HTTP level, the request-response pair initiated by the requestor entity is mapped to an HTTP POST request-response pair. For short operations where the result is returned in the response for the successfully completed operation, a 200 OK or a 201 Created HTTP response can be returned by the NFV-MANO functional entity. This means that the NFV-MANO functional entity cannot detect if this response was not delivered to the operation requestor. But the operation requestor expecting a response can definitely detect when the response is missing. Since the operation requestor is an entity of a non-NFV-MANO functional block, its reaction to such fault falls beyond the scope of NFV.

For a long running operation, if the request is accepted, a 202 Accepted HTTP response is sent by the NFV-MANO functional entity, for which the same applies as above. In addition, the final result is returned in a notification, which maps to a HTTP POST request-response pair in the opposite direction – that is from the NFV-MANO functional entity to the operation requestor. This means that the delivery of the result follows the pattern discussed in clause 5.2.3.3 discussing the delivery of notifications.

NOTE: Situations similar to those described in this clause can occur between NFV-MANO functional entities as well, and they need to be resolved. The main difference is that while entities of the non-NFV-MANO functional blocks are not in scope for NFV, NFV-MANO functional entities are in scope.

5.2.4.1.2 Actors and roles

Table 5.2.4.1.2-1 describes the use case actors and roles.

Table 5.2.4.1.2-1: Communication with an entity of a non-NFV-MANO functional block actors and roles

#	Role	Description
1	NFV-MANO functional entity	NFV-MANO functional entity that receives an operation request from an entity of a non-NFV-MANO functional block.
2	Entity of a non-NFV-MANO functional block	Entity that does not belong to NFV-MANO but has a communication relationship with an NFV-MANO functional entity.

5.2.4.1.3 Pre-conditions

Table 5.2.4.1.3-1 describes the use case pre-conditions.

Table 5.2.4.1.3-1: Communication problem with an entity of a non-NFV-MANO functional block pre-conditions

#	Pre-condition	Additional description
1	The NFV-MANO functional entity and the Entity of a non-NFV-MANO functional block are running correctly	

5.2.4.1.4 Post-conditions

Table 5.2.4.1.4-1 describes the use case post-conditions.

Table 5.2.4.1.4-1: Communication problem with an entity of a non-NFV-MANO functional block post-conditions

#	Post-condition	Additional description
1	The NFV-MANO functional entity is running correctly	
2	The Entity of a non-NFV-MANO functional block is running, but the requested operation may or may not have been performed	The operation could have been executed twice (see flow in clause 5.2.4.1.6).

5.2.4.1.5 Flow description of the case when the requestor does not receive an expected response

This flow demonstrates, how an operation can be unintentionally executed twice in case of a lost response message. If the operation is a non-idempotent scale in operation, this can lead to a VNF with insufficient resources assigned. Table 5.2.4.1.5-1 describes the details of the flow.

Table 5.2.4.1.5-1: Use case flow description of the case when the requestor does not receive an expected response

#	Actor/Role	Action/Description
Begins when	Entity of a non-NFV-MANO functional block -> NFV-MANO functional entity	An Entity of a non-NFV-MANO functional block requests an operation from an NFV-MANO functional entity. This is a long lasting operation, whose progress will be reported by separate notifications. The Entity of a non-NFV-MANO functional block starts a local timer to supervise the receipt of a response (see note 1).
Step 1	NFV-MANO functional entity -> (Entity of a non-NFV-MANO functional block)	The NFV-MANO functional entity returns a response with a HTTP 202 (accepted) status code containing the operation occurrence id `occurrenceId1` for the operation initiated by the request. Because of a communication problem, the Entity of a non-NFV-MANO functional block does not receive the response.
Step 2	NFV-MANO functional entity -> Entity of a non-NFV-MANO functional block	The NFV-MANO functional entity starts the operation and sends a notification that the operation with the id `occurrenceId1` has been started. The Entity of a non-NFV-MANO functional block is not aware of `occurrenceId1` and ignores the notification (see note 2).
Step 3	Entity of a non-NFV-MANO functional block	The timer started by the Entity of a non-NFV-MANO functional block at sending the operation request in the 'Begins when' Step expires.
Step 4	Entity of a non-NFV-MANO functional block -> NFV-MANO functional entity	The Entity of a non-NFV-MANO functional block repeats the request of the operation from an NFV-MANO functional entity. The Entity of a non-NFV-MANO functional block starts a local timer to supervise the receipt of a response.
Step 5	NFV-MANO functional entity -> Entity of a non-NFV-MANO functional block	The NFV-MANO functional entity returns a response with a HTTP 202 (accepted) status code containing the operation occurrence id `occurrenceId2` for the operation initiated by the request.
Step 6	Entity of a non-NFV-MANO functional block	After receiving the response from the NFV-MANO functional block informing about the creation of an operation with `occurrenceId2`, the Entity of a non-NFV-MANO functional block cancels the timer started in Step 4. It can start a timer to supervise the receipt of a notification indicating the start of the operation.
Step 7	NFV-MANO functional entity -> Entity of a non-NFV-MANO functional block	The NFV-MANO functional entity starts the operation for the second time and sends a notification that the operation with the id `occurrenceId2` has been started. The Entity of a non-NFV-MANO functional block is now aware of an operation being executed identified by `occurrenceId2`. It cancels the timer started in Step 6 and it creates a timer to supervise the execution of the operation.
Step 8	NFV-MANO functional entity -> Entity of a non-NFV-MANO functional block	After completion of the first operation identified by `occurrenceId1`, the NFV-MANO functional sends a notification to the Entity of a non-NFV-MANO functional block to inform about this completion. The Entity of a non-NFV-MANO functional block is not aware of an operation identified by `occurrenceId1` and ignores the notification (see note 2).
Ends when	NFV-MANO functional entity -> Entity of a non-NFV-MANO functional block	After completion of the second operation identified by `occurrenceId2`, the NFV-MANO functional sends a notification to the Entity of a non-NFV-MANO functional block to inform about this completion. The Entity of a non-NFV-MANO functional block cancels the timer that was started in Step 7 to supervise the execution of the operation.
NOTE 1: An example of a long lasting operation is a scale-in LCM operation. NOTE 2: Depending on the implementation/entity, the reaction to the unexpected message may be different.		

5.2.4.1.6 Flow description of the case when the requestor receives a response late

This flow demonstrates how an operation can be unintentionally executed twice, if the timer supervising the operation expires before the operation response is received. If the operation is a non-idempotent scale in operation, this can lead to a VNF with insufficient resources assigned. Table 5.2.4.1.6-1 describes the details of the flow.

Table 5.2.4.1.6-1: Use case flow description of the case when the requestor receives a late response

#	Actor/Role	Action/Description
Begins when	Entity of a non-NFV-MANO functional block -> NFV-MANO functional entity	An Entity of a non-NFV-MANO functional block requests an operation from an NFV-MANO functional entity. This is a long lasting operation, whose progress will be reported by separate notifications. The Entity of a non-NFV-MANO functional block starts a local timer to supervise the receipt of a response (see note 1).
Step 1	NFV-MANO functional entity -> Entity of a non-NFV-MANO functional block)	The NFV-MANO functional entity returns a response with a HTTP 202 (accepted) status code containing the operation occurrence id `occurrenceId1` for the operation initiated by the request to the Entity of a non-NFV-MANO functional block. The Entity of a non-NFV-MANO functional block does not immediately receive the response. It is received only in Step 5.
Step 2	Entity of a non-NFV-MANO functional block	The timer started by the Entity of a non-NFV-MANO functional block at sending the operation request in the 'Begins when' Step expires.
Step 3	Entity of a non-NFV-MANO functional block -> NFV-MANO functional entity	The Entity of a non-NFV-MANO functional block repeats the request of the operation from an NFV-MANO functional entity. The Entity of a non-NFV-MANO functional block starts a local timer to supervise the receipt of a response.
Step 4	NFV-MANO functional entity -> Entity of a non-NFV-MANO functional block	The NFV-MANO functional entity returns a response with a HTTP 202 (accepted) status code containing the operation occurrence id `occurrenceId2` for the operation initiated by the request. The Entity of a non-NFV-MANO functional block does not immediately receive the response. It is received only in Step 6.
Step 5	NFV-MANO functional entity ->Entity of a non-NFV-MANO functional block	After receiving the response from the NFV-MANO functional block informing about the creation of an operation with `occurrenceId1`, the Entity of a non-NFV-MANO functional block deletes the timer started in Step 3. It can start a timer to supervise the receipt of a notification indicating the start of the operation. This is the response that was sent in Step 1.
Step 6	NFV-MANO functional entity -> Entity of a non-NFV-MANO functional block	After receiving the response from the NFV-MANO functional block informing about the creation of an operation with `occurrenceId2` , the Entity of a non-NFV-MANO functional block does not do anything since the receipt of the response is not expected (see note 2). This is the response that was sent in Step 4.
Step 7	NFV-MANO functional entity -> Entity of a non-NFV-MANO functional block	The NFV-MANO functional entity starts the operation for the first time and sends a notification that the operation with the id `occurrenceId1` has been started. The Entity of a non-NFV-MANO functional block is now aware of an operation is being executed identified by `occurrenceId1`. It cancels the timer started in Step 5 and it creates a timer to supervise the execution of the operation.
Step 8	NFV-MANO functional entity -> Entity of a non-NFV-MANO functional block	The NFV-MANO functional entity starts the operation for the second time and sends a notification that the operation with the id `occurrenceId2` has been started. The Entity of a non-NFV-MANO functional block is not aware of `occurrenceId2` and ignores the notification (see note 2).
Step 9	NFV-MANO functional entity -> Entity of a non-NFV-MANO functional block	After completion of the first operation identified by `occurrenceId1`, the NFV-MANO functional sends a notification to the Entity of a non-NFV-MANO functional block to inform about this completion. The Entity of a non-NFV-MANO functional block cancels the timer that was started in Step 7 to supervise the execution of the operation.
Ends when	NFV-MANO functional entity -> Entity of a non-NFV-MANO functional block	After completion of the second operation identified by `occurrenceId2`, the NFV-MANO functional entity sends a notification to the Entity of a non-NFV-MANO functional block to inform about this completion. The Entity of a non-NFV-MANO functional block is not aware of an operation identified by `occurrenceId1` and ignores the notification (see note 2).
NOTE 1: An example of a long lasting operation is a scale-in LCM operation. NOTE 2: Depending on the implementation/entity, the reaction to the unexpected message may be different.		

5.2.4.1.7 Flow description of the case when the request is lost

This flow demonstrates the case when the request of an operation is lost. Therefore, the operation is not initiated at all. The requestor cannot distinguish this case from the case when the response is lost/delayed and, therefore, resending the request can result in the execution of the operation twice as described in the flows 5.2.4.1.6 and 5.2.4.1.7.

Table 5.2.4.1.7-1 describes the details of the flow when the request is lost.

Table 5.2.4.1.7-1: Use case flow description of the case when the request is lost

#	Actor/Role	Action/Description
Begins when	Entity of a non-NFV-MANO functional block -> NFV-MANO functional entity	An Entity of a non-NFV-MANO functional block sends a request to an NFV-MANO functional entity to execute a long running operation, which is not delivered to the NFV-MANO functional entity. The entity of the non-NFV-MANO functional block starts a timer to supervise the receipt of the response (see note 1).
Step 1	Entity of a non-NFV-MANO functional block	The timer started by the entity of the non-NFV-MANO functional block at the 'Begins when' Step expires.
Step 2	Entity of a non-NFV-MANO functional block -> NFV-MANO functional entity	The entity of the non-NFV-MANO functional block resends the request to the NFV-MANO functional entity for the same operation. The entity of the non-NFV-MANO functional block starts a timer to supervise the receipt of the response.
Step 3	NFV-MANO functional entity -> Entity of a non-NFV-MANO functional block	The NFV-MANO functional entity returns a response with a HTTP 202 (accepted) status code containing the operation occurrence id `occurrenceId1` for the operation to be initiated by the request.
Step 4	Entity of a Non-NFV-MANO functional block	After receiving the response from the NFV-MANO functional entity with `occurrenceId1`, the entity of the non-NFV-MANO functional block cancels the timer started in Step 2. It can start a timer to supervise the receipt of a notification indicating the start of the operation (see note 2).
Step 5	NFV-MANO functional entity -> Entity of a Non-NFV-MANO functional block	The NFV-MANO functional entity starts the operation and sends a notification that the operation with the id `occurrenceId1` has been started. The entity of the non-NFV-MANO functional block receiving the notification cancels the timer started in Step 4. It starts a timer to supervise the execution of the operation (see note 2).
Ends when	NFV-MANO functional entity -> Entity of a Non-NFV-MANO functional block	After completion of the operation identified by `occurrenceId1`, the NFV-MANO functional entity sends a notification to the entity of the non-NFV-MANO functional block about the completion. The entity of the non-NFV-MANO functional block cancels the timer started in Step 5 to supervise the execution of the operation.
NOTE 1: An example of a long running operation is the scale out LCM operation. NOTE 2: If a started timer expires, the entity of the non-NFV-MANO functional block proceeds to enquire the operation status.		

5.2.5 Failures caused by human errors

Errors at the interfaces may be the result of human errors. Since it is unknown to the API producer, whether the API consumer is a human or not, the use cases do not make a difference whether the consumer is a human or not. Both cases are covered by the use cases.

For instance, it does not matter whether in the use case of the clause 5.2.3.2 the reason for the communication problem is the result of a human action or not.

One area of human interaction with NFV-MANO is life cycle management of NFV-MANO. For instance, the MANO-Monitor can be a human. Since NFV-MANO LCM is not addressed by the current set of NFV specifications, the use cases do not cover these scenarios.

All this means that the difference between errors made by humans or by automatic procedures is not discussed in the present document.

5.3 NFV-MANO overload

5.3.1 NFV-MANO load management overview

The workload NFV-MANO functional entities handle can vary over time. Increased load can happen due to legitimate (e.g. disaster) or illegitimate (e.g. DDoS attack) causes. In either case, it is essential that the NFV-MANO functional entities remain in control of the NFV system they are managing and are able to provide their services, i.e. interact with their peers and the entities they manage in a timely fashion.

To achieve this, the NFV-MANO functional entities need to be prepared to handle workload fluctuations and cope with their potentially adverse increase.

For VNFs and NSs, the primary method for handling workload fluctuations is scaling. That is, the workload is monitored, for example, using PM jobs and (auto-)scaling is triggered whenever the load crosses a given threshold. This approach can be applicable to NFV-MANO functional entities as well.

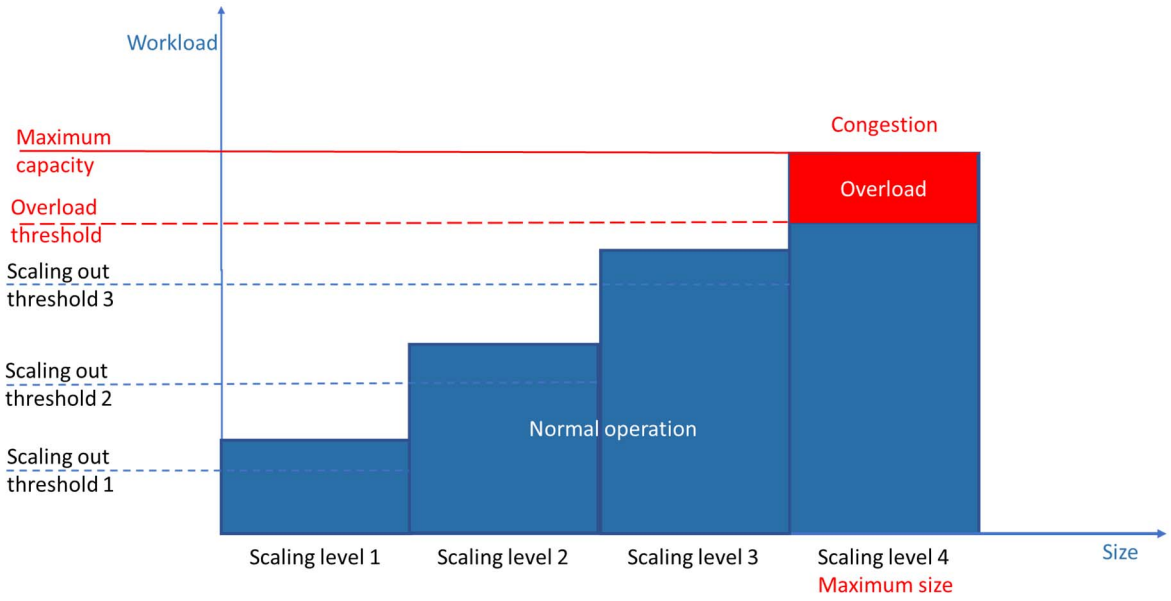

Figure 5.3.1-1: Relation between the different thresholds and the capacity of an NFV-MANO functional entity in relation to the workload and its resulting operational state

As shown in the example of figure 5.3.1-1, this means that in normal operation the NFV-MANO functional entity is scaled out if the workload increases beyond a threshold set according to the current size/capacity of the NFV-MANO functional entity, i.e. scaling out threshold. This can be detected, for instance, by using a PM job. When the workload decreases again, the NFV-MANO functional entity can be scaled in.

However, there might be limits to the extent an NFV-MANO functional entity can be scaled. Moreover, some NFV-MANO functional entity implementations might not be scalable at all. In both cases, there is a maximum workload that the NFV-MANO functional entity is able to handle, i.e. it has a maximum capacity. If this capacity is reached, the NFV-MANO functional entity will not be able to cope with the workload it is receiving and it will become congested, its input buffers and job queues will overflow, so by the time requests could be serve,d they have timed out. Eventually, the system might collapse all together. In such a situation, the main goal is to relieve the NFV-MANO functional entity from the congestion as soon as possible and return it to normal operation.

To avoid the negative effects of a congestion, proactive actions can be taken before the maximum capacity is reached. Therefore, a threshold below the maximum capacity can be defined, i.e. the overload threshold of figure 5.3.1-1. When reaching this overload threshold, the system is considered to be in the overload state.

At this time, i.e. in the overload state, the NFV-MANO functional entity is still capable of performing actions, provide feedback, prioritization, etc. in an attempt to reduce its workload in a (as much as possible) graceful manner. Different measures could be engaged simultaneously or in an escalation and, if successful, these measures might reduce the workload so that the NFV-MANO functional entity is able to cope with it and avoid congestion.

In the rest of this clause, the different scenarios related to overload and congestion control of NFV-MANO functional entities are investigated through different use cases.

5.3.2 Handling overload

5.3.2.1 Introduction and goal

During normal operation, the capacity of an NFV-MANO functional entity is adapted to the volume of incoming requests by scaling in or out, i.e. removing or adding components (see figure 5.3.1-1). But there is a limit for scaling. This could be because of resource limitations or because of implementation limitations (the implementation may not be scalable at all). Thus there is a capacity limit that cannot be extended by scaling.

To avoid unexpected behaviour when operating at the capacity limit, the load of an NFV-MANO functional entity needs to be monitored. This can be done by using a PM job. A threshold can be assigned to this PM job. One way of doing this is to define the number of maximum concurrent operations that an NFV-MANO can handle.

If the threshold is crossed, the NFV-MANO functional entity providing a service is in the overload state. The processing continues but any NFV-MANO service user requesting a service is instructed to delay future requests to allow for more processing time. It is expected that the NFV-MANO functional entity providing a service can use this time to process requests that it was not able to process before. With this, the rate of incoming requests might become lower than the rate of processed requests. As a consequence, the number of concurrent requests will fall below the maximum number, resulting in the NFV-MANO functional entity providing a service returning to the normal operation state.

5.3.2.2 Actors and roles

Table 5.3.2.2-1 describes the use case actors and roles.

Table 5.3.2.2-1: Handling overload actors and roles

#	Role	Description
1	NFV-MANO functional entity providing a service	NFV-MANO functional entity that provides a service.
2	NFV-MANO service user requesting a service	NFV-MANO service user requesting a service from the NFV-MANO functional entity providing a service. There may be different actors for this role.

5.3.2.3 Pre-conditions

Table 5.3.2.3-1 describes the use case pre-conditions.

Table 5.3.2.3-1: Handling overload pre-conditions

#	Pre-condition	Additional description
1	The NFV-MANO functional entity providing a service is running correctly	The NFV-MANO functional entity is at its maximum size. Enlarging by scaling is not possible (anymore).
2	A PM job has been created and a threshold to detect overload has been assigned to this PM job to monitor the load of the NFV-MANO functional entity providing a service	

5.3.2.4 Post-conditions

Table 5.3.2.4-1 describes the use case post-conditions.

Table 5.3.2.4-1: Handling overload post-conditions

#	Post-condition	Additional description
1	The NFV-MANO functional entity providing a service is running correctly	
2	The load has been reduced to a level that the threshold of the overload PM job is not crossed any longer. The NFV-MANO functional entity providing a service is no longer overloaded	

5.3.2.5 Flow description

Table 5.3.2.5-1 describes the flow of the use case.

Table 5.3.2.5-1: Handling an overload flow description

#	Actor/Role	Action/Description
Begins when	NFV-MANO service user requesting a service -> NFV-MANO functional entity providing a service	An NFV-MANO service user requests a service from an NFV-MANO functional entity providing a service. With this request, the threshold for detecting overload has been crossed. The NFV-MANO functional entity providing a service is now in the overload state. A notification of type ThresholdCrossedNotification with direction UP is sent to registered entities.
Step 1	NFV-MANO functional entity providing a service -> NFV-MANO service user requesting a service	The NFV-MANO functional entity providing a service sends a HTTP status code 2XX indicating that the request has been accepted and will be processed, together with information about how long the next request should be delayed.
Step 2	NFV-MANO functional entity providing a service -> NFV-MANO service user requesting a service	After processing the request, the answer is sent back to the NFV-MANO service user requesting a service.
Step 3	NFV-MANO service user requesting a service -> NFV-MANO functional entity providing a service	An NFV-MANO service user (potentially a different entity than the SU that requested the service in Step 1) requests a service from an NFV-MANO functional entity providing a service. If the overload threshold for the NFV-MANO functional entity providing a service is still crossed, the flow continues in Step 1 (see note).
Ends when	NFV-MANO functional entity providing a service -> NFV-MANO service user requesting a service	The NFV-MANO functional entity providing a service detects that the threshold for detecting overload is not crossed anymore. Registered entities are informed by sending a ThresholdCrossedNotification with direction DOWN. It sends a HTTP status code 202 Accepted to the NFV-MANO service user requesting a service indicating that the request has been accepted and will be processed.
NOTE: Service requests are handled as described in Step 1 until the threshold is not crossed anymore (described in 'Ends when'). There is no guarantee that this will always happen. Instead, it can happen that the NFV-MANO functional entity providing a service reaches its maximum capacity. In this case, use case of the clause 5.3.2 "Congestion control" is activated.		

5.3.3 Priority based request handling during overload

5.3.3.1 Introduction

When an NFV-MANO functional entity is in the overload state, it might not be able to handle all the incoming requests. However, it might still have some capacity to handle urgent or important requests, which means the NFV-MANO functional entity needs to decide which requests it is going to execute and which ones it is going to reject with an appropriate return code (e.g. HTTP 429 Too Many Requests). Accordingly, the overload threshold could be associated with a policy to be used to determine the priority of the different requests possible based on the priority of the requesting entity and/or the operation being requested. Based on the determined priority and its actual workload, the NFV-MANO functional entity can determine the applicable reaction (i.e. accept or reject).

5.3.3.2 Actors and roles

Table 5.3.3.2-1 describes the use case actors and roles.

Table 5.3.3.2-1: Overload handling actors and roles

#	Role	Description
1	Service Provider FE	The service provider NFV-MANO Functional Entity (FE) providing the NFV-MANO Service.
2	High Priority SU	An NFV-MANO Service User (SU) requesting the NFV-MANO Service with higher priority (see note 1).
3	Low Priority SU	An NFV-MANO service user requesting the NFV-MANO Service with lower priority (see note 1).
4	Existing SU	An NFV-MANO service user which has an ongoing request with the Service Provider FE for the NFV-MANO Service.
5	MANO-Monitor	A management entity that has started a PM job with a threshold on the NFV-MANO Service provided by the Service Provider FE and that has registered to receive notifications about related threshold crossings (see note 2).
6	NFV-MANO Service	The service offered by the Service Provider FE, for which the MANO-Monitor has set up a PM job with an overload threshold and which becomes overloaded.

NOTE 1: The priority of a request could depend on, among others, the priority of the service user, the type of the requested operation, or on their combination.

NOTE 2: To set an appropriate threshold, the MANO-Monitor can consider information such as:

- the implementation limitations provided by the vendor on the capacity of the Service Provider FE for the NFV-MANO Service;
- the applicable administrative limitations of the NFV-MANO Service capacity, for example, based on licencing;
- the applicable traffic patterns observed by the operator; as well as
- the policy to be used with the threshold and its expected effect on the workload.

5.3.3.3 Pre-conditions

Table 5.3.3.3-1 describes the use case pre-conditions.

Table 5.3.3.3-1: Overload handling pre-conditions

#	Pre-condition	Additional description
1	The Service Provider FE is operating correctly and has a PM job started by the MANO-Monitor with a threshold for overload defined and a policy associated	The workload associated with the NFV-MANO Service offered by the Service Provider NFV-MANO functional entity is being monitored by a PM job for which the overload threshold has been set up including an associated policy.
2	The policy determining the prioritization of valid requests under overload condition is known and has been associated with the PM job threshold	The Service Provider NFV-MANO functional entity has the information about the policy applicable at the crossing of the overload threshold, i.e. when it transitions to the overload state. Using this policy, the Service Provider NFV-MANO functional entity is capable of determining the priority of new incoming requests and the applicable reaction. The priority could depend on, among others, the priority of the service user, the type of the requested operation, or their combination. The applicable reaction can depend on the current workload associated with NFV-MANO Service.
3	The MANO-Monitor has subscribed to receive threshold crossing notifications	The MANO-Monitor has started a PM job on the NFV-MANO Service offered by the Service Provider NFV-MANO functional entity, and it has set up an overload threshold with an associated policy which determines the priority of incoming requests and their handling (see note).
NOTE: To set an appropriate threshold, the MANO-Monitor can consider information such as: – the implementation limitations provided by the vendor on the capacity of the Service Provider FE for the NFV-MANO Service; – the applicable administrative limitations of the NFV-MANO Service capacity, for example, based on licencing; – the applicable traffic patterns observed by the operator; as well as – the policy to be used with the threshold and its expected effect on the workload.		

5.3.3.4 Post-conditions

Table 5.3.3.4-1 describes the use case post-conditions.

Table 5.3.3.4-1: Overload handling post-conditions

#	Post-condition	Additional description
1	The workload of the Service Provider FE is below the overload threshold	
2	The Service Provider FE accepts valid requests for the NFV-MANO Service from any service user	

5.3.3.5 Flow description

Table 5.3.3.5-1 describes the flow of the use case in which the service provider NFV-MANO functional entity (Service Provider FE) has reached the overload state for the NFV-MANO Service by executing requests from different service users, including the Existing SU. As a result, subsequent requests are evaluated first according to the policy associated with the overload threshold. Requests evaluated as higher priority (coming from the High Priority SU) are accepted for execution, while requests evaluated as lower priority (coming from the Low Priority SU) are rejected with an appropriate return code. Once the execution of some of the ongoing requests completes, the workload of the Service Provider FE for the NFV-MANO Service drops below the overload threshold.

Table 5.3.3.5-1: Overload handling flow description

#	Actor/Role	Action/Description
Begins when	Existing SU -> Service Provider FE	The Service Provider FE has received a request from the Existing SU for the NFV-MANO Service, which was accepted, and the Service Provider FE has started to execute the requested operation. With the acceptance of the request, the Service Provider FE determines that it has crossed the overload threshold for the PM job and now it is in the overload state.
Step 1	Service Provider FE -> MANO-Monitor	The Service Provider FE sends a threshold crossed notification with direction UP to the MANO-Monitor and any other entity subscribing for such notifications (see note 1).
Step 2	MANO-Monitor -> Service Provider FE	The MANO-Monitor activates the policy associated with the threshold crossing (see note 2).
Step 3	High Priority SU -> Service Provider FE	The Service Provider FE receives a request from the High Priority SU for the NFV-MANO Service. The Service Provider FE evaluates the priority of the request using the policy associated with the overload threshold and determines that it has high priority. Therefore, it needs to be accepted and executed.
Step 4	Service Provider FE -> High Priority SU	In its response, the Service Provider FE indicates that the request of the High Priority SU has been accepted and is being executed.
Step 5	Low Priority SU -> Service Provider FE	The Service Provider FE receives a request from the Low Priority SU for the NFV-MANO Service. The Service Provider FE evaluates the priority of the request using the policy associated with the overload threshold and determines that it has lower priority. Therefore, it needs to be rejected.
Step 6	Service Provider FE -> Low Priority SU	In its response to the Low Priority SU, the Service Provider FE indicates that the request has been rejected due to overload and indicates that the request can be repeated at a later time.
Step 7	Low Priority SU	The Low Priority SU starts a waiting timer as indicated in the received rejection response.
Step 8	Service Provider FE -> Existing SU	The Service Provider FE has completed the execution of the operation associated with the request of the Existing SU. The Service Provider FE sends the results to the Existing SU. The workload of the Service Provider FE remains above the overload threshold.
Step 9	Service Provider FE -> High Priority SU	The Service Provider FE has completed the execution of the operation associated with the request of the High Priority SU. The Service Provider FE sends the results to the High Priority SU. With this, the workload of the Service Provider FE drops below the overload threshold.
Step 10	Service Provider FE -> MANO-Monitor	The Service Provider FE sends a threshold crossed notification with direction DOWN to the MANO-Monitor and any other entity subscribing for such notifications (see note 1).
Ends when	MANO-Monitor -> Service Provider FE	The MANO-Monitor deactivates the policy associated with the threshold crossing. Therefore, Service Provider FE can accept any valid request including the one the Low Priority SU will resend when its waiting timer started in Step 7 expires.
NOTE 1: The Existing SU, High Priority SU and Low Priority SU might be among the subscribers for the threshold crossing notifications. For example, the NFVO would want to subscribe to such notifications to know which VNFM or VIM is overloaded.		
NOTE 2: The policy associated with the threshold crossing could be installed (or transferred) at the time of the instantiation of the Service Provider FE, at the time the MANO-Monitor sets the threshold, or at the latest, as a result of the threshold crossing notification and before its activation.		

5.3.4 Congestion control

5.3.4.1 Introduction and goal

Crossing the overload threshold, an NFV-MANO functional entity will implement measures to reduce the load as described in the use cases of the clause 5.3.1. If these measures are insufficient and the load still increases, the maximum capacity will be reached (see figure 5.3.1-1) and the NFV-MANO functional entity will become congested. The maximum capacity can be determined based on different KPIs, for example, as the maximum number of concurrent operations as defined in ETSI GS NFV-IFA 031 [i.6]. In the congestion state, the functional entity will either reject incoming operations if possible or drop them silently. In the latter case, the requestor will not be aware of the problem, hence the rejection of incoming requests is the preferred option. This is the case for the HTTP response code 429 'Too Many Requests' referenced in ETSI NFV-SOL specifications. In addition, an alarm notification is sent to registered entities indicating that the NFV-MANO functional entity is in a congestion state.

5.3.4.2 Actors and roles

Table 5.3.4.2-1 describes the use case actors and roles.

Table 5.3.4.2-1: Congestion actors and roles

#	Role	Description
1	NFV-MANO functional entity providing a service	NFV-MANO functional entity that provides a service.
2	NFV-MANO service user requesting a service	NFV-MANO service user requesting the service of the NFV-MANO functional entity providing a service.
3	Alarm-Aggregator	Entity responsible for maintaining an aggregated list of alarm conditions in the NFV system. For this purpose, it has registered with the NFV-MANO functional entities to receive alarm notifications. The Alarm-Aggregator will forward notifications to registered entities.

5.3.4.3 Pre-conditions

Table 5.3.4.3-1 describes the use case pre-conditions.

Table 5.3.4.3-1: Congestion pre-conditions

#	Pre-condition	Additional description
1	The NFV-MANO functional entity providing a service, the NFV-MANO service user requesting a service, and the Alarm-Aggregator are running correctly.	
2	The NFV-MANO functional entity providing a service is in the overload state	The load of the NFV-MANO functional entity providing a service has crossed the overload threshold, and is reaching its maximum capacity.
3	The Alarm-Aggregator has registered with the NFV-MANO functional entities to receive alarm notifications	

5.3.4.4 Post-conditions

Table 5.3.4.4-1 describes the use case post-conditions.

Table 5.3.4.4-1: Congestion post-conditions

#	Post-condition	Additional description
1	The NFV-MANO functional entity providing a service, the NFV-MANO service user requesting a service, and the Alarm-Aggregator are running correctly	
2	The NFV-MANO functional entity providing a service is in the overload state	The load of the NFV-MANO functional entity that experienced congestion is below its maximum capacity.

5.3.4.5 Flow description

Table 5.3.4.5-1 describes the flow of the use case in case of rejection of the request.

Table 5.3.4.5-1: Flow description in case of rejection of the request

#	Actor/Role	Action/Description
Begins when	NFV-MANO functional entity providing a service -> Alarm-Aggregator	The load level is so high that the maximum capacity is reached. With that, the NFV-MANO functional entity providing a service enters the state of congestion. The NFV-MANO functional entity creates an alarm in its active alarms list and sends an alarm notification indicating its state of congestion. The Alarm-Aggregator adds the alarm to the global list of active and sends the alarm notification to registered entities. This could include NFV-MANO service users.

#	Actor/Role	Action/Description
Step 1	NFV-MANO service user requesting a service -> NFV-MANO functional entity providing a service	The NFV-MANO service user requests the execution of an operation and starts a timer to supervise its request.
Step 2	NFV-MANO functional entity providing a service -> NFV-MANO service user requesting a service	Executing the operation is not possible as the maximum capacity has been reached. The NFV-MANO functional entity rejects the operation by sending a HTTP 429 'Too Many Requests' response code with a Retry-After header indicating a period for which no requests should be sent to the NFV-MANO service user requesting a service.
Step 3	NFV-MANO service user requesting a service	When receiving the reject, the NFV-MANO service user cancels the timer supervising its request and starts a new timer according to the response received from the NFV-MANO functional entity providing a service.
Step 4	NFV-MANO functional entity providing a service	The load level drops below the maximum capacity and the NFV-MANO functional entity providing a service comes out of the congestion, but remains overloaded.
Step 5	NFV-MANO functional entity providing a service-> Alarm-Aggregator	The NFV-MANO functional entity removes the alarm from its active alarms list and sends an alarm clearing notification. The Alarm-Aggregator updates the global list of active alarms and sends the alarm clearing notification to registered entities. This could include NFV-MANO service users.
Ends when	NFV-MANO service user requesting a service	When the timer started in Step 3 expires, the NFV-MANO service user requesting a service can resend its request.

Table 5.3.4.5-2 describes the flow of the use case in case of dropping the request.

Table 5.3.4.5-2: Flow description in case of dropping the request

#	Actor/Role	Action/Description
Begins when	NFV-MANO functional entity providing a service -> Alarm-Aggregator	The load level is so high that the maximum capacity is reached. With that, the NFV-MANO functional entity providing a service enters the state of congestion. The NFV-MANO functional entity creates an alarm in its active alarms list and sends an alarm notification indicating its state of congestion. The Alarm-Aggregator creates an entry in the global list of active alarms for the alarm and sends the alarm notification to registered entities. This could include NFV-MANO service users.
Step 1	NFV-MANO service user requesting a service -> NFV-MANO functional entity providing a service	The NFV-MANO service user requests the execution of an operation and starts a timer to supervise its request.
Step 2	NFV-MANO functional entity providing a service	Executing the operation is not possible as the maximum capacity has been reached. The NFV-MANO functional entity does not answer the request, i.e. the request is dropped.
Step 3	NFV-MANO functional entity providing a service	The load level drops below the maximum capacity and the NFV-MANO functional entity providing a service comes out of the congestion, but remains overloaded.
Step 4	NFV-MANO functional entity providing a service-> Alarm-Aggregator	The NFV-MANO functional entity removes the alarm from its active alarms list and sends an alarm clearing notification. The Alarm-Aggregator updates the global list of active alarms with the clearing and sends the alarm clearing notification to registered entities. This could include NFV-MANO service users.
Ends when	NFV-MANO service user requesting a service	The timeout of the timer started in Step 1 indicates that the service request might not have been processed (see notes 1 and 2).
NOTE 1: After receiving the timeout, the NFV-MANO service user requesting a service can decide whether to resend the request. The timeout value is either defined by the interface specification or is an implementation decision and could depend on whether the NFV-MANO service user have received the alarm notification about the congestion. NOTE 2: If the NFV-MANO service user requesting a service has not received the alarm notification about the congestion, then it is not able to distinguish a message loss (request or response) from the congested NFV-MANO functional entity dropping the request. Thus, resending a non-idempotent operation request can have side effects.		

6 Recommendations

6.1 Introduction

This clause provides recommendations for NFV-MANO which have been derived from the use cases discussed in clause 5. The recommendations are made from a reliability point of view.

The following terminology is used:

- "It is recommended that a requirement be specified" means that the recommendation should be addressed in subsequent specifications by creating requirements using the auxiliary "shall".
- "It is recommended that" means that the recommendation should be addressed in subsequent specifications by creating recommendations using the auxiliary "should".

6.2 General recomendations

Table 6.2-1 provides general recommendations related to the NFV-MANO functional entities.

Table 6.2-1: General recommendations related to the NFV-MANO functional entities

Identifier	Recommendation description	Use case reference
Gen.001	It is recommended that a requirement be specified to introduce load levels for NFV-MANO functional entities. At least two levels, overload and maximum load, need to be supported.	Clause 5.3.1
Gen.002	It is recommended that a requirement be specified that an NFV-MANO functional entity can compute its load level.	Clause 5.3.1
Gen.003	It is recommended that a requirement be specified that an NFV-MANO functional entity can be instructed to adapt its request handling to the load.	Clauses 5.3.3, 5.3.4
Gen.004	It is recommended that a requirement be specified that an NFV-MANO functional entity can be deployed redundantly.	Clauses 5.2.2.1, 5.2.2.2
Gen.005	It is recommended that a requirement be specified that an NFV-MANO functional entity supports reporting its status changes due to internal events or due to failures.	Clauses 5.2.2.1, 5.2.2.2, 5.2.2.3
Gen.006	It is recommended that a requirement be specified that an NFV-MANO functional entity supports capabilities of being monitored (e.g. healthcheck) by an external entity to detect failures.	Clauses 5.2.2.2, 5.2.2.3
Gen.007	It is recommended that a requirement be specified that an NFV-MANO functional entity supports capabilities of being repaired (e.g restart script) by an external entity.	Clauses 5.2.2.2, 5.2.2.3
Gen.008	It is recommended that a requirement be specified that an NFV-MANO functional entity supports capabilities of storing the state of its NFV-MANO services on a storage external to the NFV-MANO functional entity.	Clause 5.2.2.3
Gen.009	It is recommended that a requirement be specified that an NFV-MANO functional entity supports capabilities of restoring the state of its NFV-MANO services from a storage external to the NFV-MANO functional entity.	Clause 5.2.2.3
Gen.010	It is recommended that a requirement be specified to provide reliable notification delivery between NFV-MANO entities (see note).	Clause 5.2.1.3
Gen.011	It is recommended that a requirement be specified to provide reliable notification delivery to non-NFV-MANO functional blocks (see note).	Clauses 5.2.3.3, 5.2.4.1
Gen.012	It is recommended that a requirement be specified to maintain a global list of active alarms.	Clauses 5.2.1.1, 5.2.1.2, 5.2.1.3, 5.2.3.1, 5.2.3.2, 5.3.4
Gen.013	It is recommended that a requirement be specified that an NFV-MANO functional entity supports overload protection and congestion control mechanisms.	Clauses 5.3.2, 5.3.3, 5.3.4
NOTE: No assumption is made about the way how to provide this reliable delivery. This can be achieved, among others, by using a lower layer mechanism that sufficiently guarantees the notification delivery, or by implementing the notification delivery as a two way communication mechanism.		

6.3 Recommendations of functional requirements for NFV-MANO functional entities

Table 6.3-1 provides recommendations related to the functional requirements for NFV-MANO functional entities.

Table 6.3-1: Recommendations related to the functional requirements for NFV-MANO functional entities

Identifier	Recommendation description	Use case reference
Func.001	It is recommended that a requirement be specified that an NFV-MANO functional entity supports the option of being deployed as a set of redundancy units to protect against failures.	Clauses 5.2.2.1, 5.2.2.2
Func.002	It is recommended that a requirement be specified that an NFV-MANO functional entity deployed as a set of redundancy units uses an internal mechanism to detect failures of its redundancy unit instances.	Clauses 5.2.2.1, 5.2.2.2
Func.003	It is recommended that a requirement be specified that an NFV-MANO functional entity deployed as a set of redundancy units uses internal support to perform transparent failover between its redundancy unit instances (see note).	Clause 5.2.2.1
Func.004	It is recommended that a requirement be specified that an NFV-MANO functional entity deployed as a set of redundancy units provides a mechanism to repair failed redundancy unit instances in order to restore their redundancy.	Clause 5.2.2.1
Func.005	It is recommended that a requirement be specified that an NFV-MANO functional entity supports capabilities of being monitored by an external entity to detect failures on the NFV-MANO functional entity as whole, on NFV-MANO services it provides, and on its individual redundancy unit instances.	Clauses 5.2.2.2, 5.2.2.3
Func.006	It is recommended that a requirement be specified that an NFV-MANO functional entity supports capabilities of being repaired by an external entity as necessary at the NFV-MANO functional entity instances level and at the NFV-MANO functional entity redundancy unit instances level.	Clauses 5.2.2.2, 5.2.2.3
Func.007	It is recommended that a requirement be specified that an NFV-MANO functional entity supports capabilities of checking and updating the state of its NFV-MANO services with changes that occurred while the services were not available.	Clause 5.2.2.3
Func.008	It is recommended that a requirement be specified that an NFV-MANO functional entity supports capabilities of checking and verifying the state of ongoing operations initiated before a failure and resume their execution as necessary.	Clause 5.2.2.3
Fun.009	It is recommended that a requirement be specified that an NFV-MANO functional entity maintains a list of active alarms.	Clauses 5.2.1.1, 5.2.1.2, 5.2.1.3, 5.2.3.1, 5.2.3.2, 5.3.4
Fun.010	It is recommended that a requirement be specified that an NFV-MANO functional entity creates an entry in its active alarm list for every error or failure encountered with an appropriate severity. This includes internal errors as well as failures detected in other NFV-MANO functional entities.	Clauses 5.2.1.1, 5.2.1.2, 5.2.3.1, 5.2.3.2, 5.3.4
Fun.011	It is recommended that a requirement be specified that an NFV-MANO functional entity reports detected changes of the severity of any alarm on its active alarm list.	Clause 5.2.1.3
Fun.012	It is recommended that a requirement be specified that an NFV-MANO functional entity, while it is in the congestion state, is allowed to stop accepting new requests.	Clause 5.3.4
Fun.013	It is recommended that an NFV-MANO functional entity, while it is in the congestion state, continues processing already accepted requests.	Clause 5.3.4
Fun.014	It is recommended that a requirement be specified that enables an NFV-MANO functional entity to detect if one of its messages was not delivered. This includes notifications used for one way communication.	Clause 5.2.3.2
Fun.015	It is recommended that a requirement be specified that an NFV-MANO functional entity is allowed to handle (accept or reject) incoming requests according to the current load condition.	Clauses 5.3.3, 5.3.4
Fun.016	It is recommended to allow an NFV-MANO functional entity to assign priorities to service requests.	Clause 5.3.3
NOTE: A failover is transparent if the users of an NFV-MANO service detect no impact of the failure and its handling in their interaction with the NFV-MANO functional entity.		

6.4 Recommendations for interfaces of NFV-MANO functional entities

Table 6.4-1 provides recommendations related to the interfaces of NFV-MANO functional entities.

Table 6.4-1: Recommendations related to the interfaces of NFV-MANO functional entities

Identifier	Recommendation description	Use case reference
If.001	It is recommended that a requirement be specified to enable an NFV-MANO functional entity to reflect its load state in the HTTP status code that is sent in response to a request.	Clause 5.3.2
If.002	It is recommended that a requirement be specified that an NFV-MANO functional entity supports indicating its need for external assistance to recover from a failure.	Clause 5.2.2.1
If.003	It is recommended that a requirement be specified that an NFV-MANO functional entity supports being triggered by an external entity to resume recovery from a failure.	Clause 5.2.2.1
If.004	It is recommended that a requirement be specified that an NFV-MANO functional entity supports the capability of detecting a retransmission of a request and returning an appropriate response (see note 1).	Clauses 5.2.3.3, 5.3.4
If.005	It is recommended that a requirement be specified that an NFV-MANO functional entity sends an alarm notification when it reaches the congestion state.	Clause 5.3.4
If.006	It is recommended that a requirement be specified that an NFV-MANO functional entity sends an alarm clearing notification when it comes out of the congestion state.	Clause 5.3.4
If.007	It is recommended that an NFV-MANO functional entity, while it is in the congestion state, sends a "too many requests" rejection response to a new operation request it receives.	Clause 5.3.4
If.008	It is recommended that a requirement be specified that an alarm notification contains at least the information about the NFV-MANO functional entity which raised the alarm, the entity on which the alarm has been raised, and the NFV-MANO functional entity sending the alarm notification if different from the one which raised the alarm.	Clause 5.2.3.1
If.009	It is recommended that a requirement be specified that enables an NFV-MANO functional entity to expose to service users suggested timeout values to safeguard their requests.	Clause 5.2.4.1
If.010	It is recommended that a requirement be specified that enables an NFV-MANO service user to find out a list of operation requests satisfying certain criteria that was received by a service provider NFV-MANO functional entity (see note 2).	Clause 5.2.4.1
If.011	It is recommended that a requirement be specified that enables an NFV-MANO functional entity to indicate in a response to a service user the expected behaviour with respect to its future requests.	Clause 5.3.3
NOTE 1: An appropriate response to a retransmitted request means that the response will indicate the operation occurrence id generated in response to the original operation request and the status appropriate for the execution status. NOTE 2: It is the responsibility of the service provider NFV-MANO functional entity to limit the size of the history of received operations to a suitable value.		

6.5 Recommendations for the Alarm-Aggregator

The term "Alarm-Aggregator" is used for the purpose to describe the functionality and to address the entity that is providing that functionality. No assumption is made about what entity can play such a role.

Table 6.5-1 provides recommendations related to the Alarm-Aggregator.

Table 6.5-1: Recommendations related to the Alarm-Aggregator

Identifier	Recommendation description	Use case reference
Aag.001	It is recommended that a requirement be specified to provide an Alarm-Aggregator function that maintains a global list of active alarms. The global list of active alarms is the union of the active alarm lists of all NFV-MANO functional entities.	Clauses 5.2.1, 5.2.3, 5.3.4
Aag.002	It is recommended that a requirement be specified that the Alarm-Aggregator registers with all NFV-MANO functional entities to receive an alarm information for every alarm that is created by these NFV-MANO functional entities.	Clauses 5.2.1, 5.2.3, 5.3.4
Aag.003	It is recommended that a requirement be specified to provide a single point of enquiry of all active alarms in NFV-MANO, i.e. the Alarm-Aggregator function.	Clauses 5.2.1, 5.2.3, 5.3.4
Aag.004	It is recommended that a requirement be specified that the Alarm-Aggregator function provides the capability of subscribing with it for alarm notifications sent by any NFV-MANO functional entity.	Clauses 5.2.1, 5.2.3, 5.3.4
Aag.005	It is recommended that a requirement be specified to provide the capability of the Alarm-Aggregator function to forward received alarm notifications to subscribing entities interested in those alarm notifications.	Clauses 5.2.1, 5.2.3, 5.3.4
Aag.006	It is recommended that a requirement be specified that the Alarm-Aggregator registers with all NFV-MANO functional entities to receive information about all alarms.	Clauses 5.2.1, 5.2.3, 5.3.4
Aag.007	It is recommended that a requirement be specified that the Alarm-Aggregator function has the capability of keeping the global list of active alarms in sync (i.e. up-to-date) with the individual lists of active alarms of NFV-MANO functional entities.	Clauses 5.2.1, 5.2.3, 5.3.4

6.6 Recommendations related to the MANO-Monitor

The term "MANO-Monitor" is used for the purpose to describe the functionality and to address the entity that is providing that functionality. No assumption is made about what entity can play such a role.

Table 6.6-1 provides recommendations related to the MANO-Monitor.

Table 6.6-1: Recommendations related to the MANO-Monitor

Identifier	Recommendation description	Use case reference
Mmo.001	It is recommended that a requirement be specified that enables the MANO-Monitor to inquire about any/all alarms that are active in the active alarm lists of any/all the NFV-MANO functional entities.	Clauses 5.2.1.1, 5.2.1.2, 5.2.1.3, 5.2.2.1, 5.2.2.2, 5.2.2.3, 5.2.3.1, 5.2.3.2
Mmo.002	It is recommended that a requirement be specified that enables the MANO-Monitor to correlate alarms raised separately but that are due to the same root cause.	Clause 5.2.3.1
Mmo.003	It is recommended that a requirement be specified that enables the MANO-Monitor to inform the creator of an active alarm that it has taken over the responsibility to remove the root cause of this alarm.	Clauses 5.2.1.1, 5.2.1.2, 5.2.3.1, 5.2.3.2
Mmo.004	It is recommended that a requirement be specified that an NFV-MANO functional entity stops sending notifications about an active alarm, which it has created and which has not changed, once the MANO-Monitor has taken over the responsibility of removing the root cause of the alarm.	Clauses 5.2.1.1, 5.2.1.2, 5.2.3.1, 5.2.3.2
Mmo.005	It is recommended that a requirement be specified that enables the MANO-Monitor to inform the creator of an active alarm about the potential removal of the root cause.	Clauses 5.2.1.1, 5.2.1.2, 5.2.3.1, 5.2.3.2
Mmo.006	It is recommended that a requirement be specified that an NFV-MANO functional entity confirms with the MANO-Monitor the successful removal of the root cause of an active alarm it has created.	Clauses 5.2.1.1, 5.2.1.2, 5.2.3.1, 5.2.3.2

Annex A:
Change History

Date	Version	Information about changes
November 2019	0.0.1	Early draft
April 2020	0.0.2	NFVREL(20)000001r1 Clause 2.2 Informative References, NFVREL(20)000002 Clause 3.1 Terms, NFVREL(20)000036r3 Clause 5.1.1 Use Case MANO functional entity failure, NFVREL(20)000029r1 Clause 5.1.x Use Case Detecting a failure of another NFV-MANO-functional entity, NFVREL(20)000010r3 Clause 5.1.3.1 Use Case correlation of failures of NFV-MANO functional entities
September 2020	0.0.3	NFVREL(20)000096r2 Clause 5.1.3.2 Use case communication error between NFV-MANO functional entities, NFVREL(20)000092r4 Clause 5.1.4 Failures in the interworking of NFV-MANO functional entities with other non-MANO functional blocks, NFVEVE(20)000062r1 Proposal to modify NFV003 definitions (adapt definition changes)
January 2021	0.0.4	NFVREL(20)000150r3 REL012 NFV-MANO architectural considerations NFVREL(20)000157r3 REL012 NFV-MANO load management overview NFVREL(20)000170r3 REL012 Overload handling NFVREL(20)000161r4 REL012 Discussion paper congestion use case NFVREL(20)000147r1 REL012 NFV-MANO functional entity internal failover NFVREL(20)000169r4 REL012 Clause 5.2.3 overload use case NFVREL(20)000174r1 REL012 NFV-MANO functional entity redundancy NFVREL(20)000175r1 REL012 REL012 NFV-MANO functional entity externally managed failover
April 2021	0.0.5	NFVREL(21)000014r2 REL012 Notifications delivery use case NFVREL(21)000016r4 REL012 Failover of NFV-MANO functional entities NFVREL(21)000034 REL012 Revision of clause 4 NFVREL(21)000024r3 REL012 Revision of clause 5.1.3.2 NFVREL(21)000030r4 REL012 Clause 5.1.1.1.5 Alarm handling flow modification proposal NFVREL(21)000025r4 REL012 Clause 5.1.X Alarm escalation NFVREL(21)000035r1 REL012 Revision of clause 5.1.4 NFVREL(21)000049r1 REL012 Revision of clause 5.1.1.2 NFVREL(21)000050r1 REL012 Revision of clause 5.1.2 NFVREL(21)000051r1 REL012 Revision of clause 5.1.3.1 NFVREL(21)000045r2 REL012 Clause 5.1. NFVREL(21)000060 REL012 Revision of clause 5.1.3.3
May 2021	0.0.6	NFVREL(21)000062r2 REL012 Clause 5.1.4.1 revision of flow 1 NFVREL(21)000063r1 REL012 Clause 5.1.4.1 two additional flows NFVREL(21)000065r1 REL012 Clause 5.1.4.1 one more flow NFVREL(21)000061r1 REL012 Revision of clause 5.2.4 NFVREL(21)000067r2 REL012 Clause 5.1.5 Failures caused by human errors
June 2021	0.0.7	NFVREL(21)000081r2 REL012 Structure for Clause 6 plus NFV-MANO Overload recommendations NFVREL(21)000075r2 REL012 Entity internal error handling recommendations NFVREL(21)000076r2 REL012 Entity external error handling recommendations NFVREL(21)000084r2 REL012 additional recommendations NFVREL(21)000085r1 REL012 congestion recommendations NFVREL(21)000086r1 REL012 alarm aggregator recommendations NFVREL(21)000091r2 REL012 6.1 recommendations introduction NFVREL(21)000092r1 REL012 MANO Monitor related recommendations NFVREL(21)000093r2 REL012 some more recommendations
August 2021	0.0.8	NFVREL(21)000124r1 REL012 additional alarm aggregator recommendation NFVREL(21)000126r1 REL012 5.1 Use cases introduction NFVREL(21)000122r1 REL012 Foreword

History

Document history		
V1.1.1	November 2021	Publication