# ETSI GS NFV-IFA 049 V5.4.1 (2026-01)

**GROUP SPECIFICATION**

# Network Functions Virtualisation (NFV) Release 5; Architectural Framework; VNF generic OAM functions and other PaaS Services specification

***Disclaimer***

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

Reference
RGS/NFV-IFA049ed541

Keywords
information model, interface, management, NFV, OAM, VNF

***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

***Important notice***

The present document can be downloaded from the ETSI Search & Browse Standards application.

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on ETSI deliver repository.

Users should be aware that the present document may be revised or have its status changed, this information is available in the Milestones listing.

If you find errors in the present document, please send your comments to the relevant service listed under Committee Support Staff.

If you find a security vulnerability in the present document, please report it through our Coordinated Vulnerability Disclosure (CVD) program.

***Notice of disclaimer & limitation of liability***

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

***Copyright Notification***

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.
The content of the PDF version shall not be modified without the written authorization of ETSI.
The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2026.
All rights reserved.

# 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 IPR online database.

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™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

# Foreword

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

# Modal verbs terminology

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

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

# 1 Scope

The present document specifies the VNF generic OAM functions and other PaaS Services framework, along with its PaaS Services management functions.

Regarding the VNF generic OAM functions and other PaaS Services, the present document specifies the functional requirements and the interface requirements for VNF generic OAM functions and other PaaS Services defined by the present document. Specification of the interfaces exposed by the VNF generic OAM functions and other PaaS Services, and the relevant information models are also provided, like also a description of requirements and modelling of descriptors of PaaS Services, with specific additional requirements for VNF generic OAM functions.

Regarding the PaaS Service management, the present document specifies interface requirements of interfaces produced by the PSM and PSR (as functionally specified in ETSI GS NFV-IFA 010 [i.9]), and their interface operations and associated information modelling.

# 2 References

## 2.1 Normative references

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

Referenced documents which are not found to be publicly available in the expected location might be found in the ETSI docbox.

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

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

[1] ETSI GS NFV-IFA 011: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; VNF Descriptor and Packaging Specification".

[2] ETSI GS NFV-IFA 036: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Requirements for service interfaces and object model for container cluster management and orchestration specification".

[3] ETSI GS NFV-IFA 048: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Policy Information Model Specification".

[4] ETSI GS NFV-IFA 013: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Os-Ma-nfvo reference point - Interface and Information Model Specification".

[5] ETSI GS NFV-TST 013 (V1.1.1): "Network Functions Virtualisation (NFV); Testing; Test Case Description Template Specification".

[6] ETSI GS NFV-IFA 026: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Architecture enhancement for Security Management Specification".

[7] ETSI GS NFV-IFA 027 (V5.4.1): "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Performance Measurements Specification".

## 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 may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

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

[i.2] ETSI GR NFV-EVE 019 (V5.1.1): "Network Functions Virtualisation (NFV) Release 5; Architectural Framework; Report on VNF generic OAM functions".

[i.3] ETSI GR NFV-IFA 029 (V3.3.1): "Network Functions Virtualisation (NFV) Release 3; Architecture; Report on the Enhancements of the NFV architecture towards "Cloud-native" and "PaaS"".

[i.4] ETSI GR NFV-EVE 022 (V5.1.1): "Network Functions Virtualisation (NFV) Release 5; Architectural Framework; Report on VNF configuration".

[i.5] ETSI GS NFV-IFA 008: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Ve-Vnfm reference point - Interface and Information Model Specification".

[i.6] ETSI GR NFV-IFA 041 (V4.1.1): "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Report on enabling autonomous management in NFV-MANO".

[i.7] ETSI GS NFV-IFA 047: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Management data analytics Service Interface and Information Model".

[i.8] IEEE 1588™-2019: "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems".

[i.9] ETSI GS NFV-IFA 010: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Functional requirements specification".

[i.10] ETSI GR NFV-IFA 037 (V4.1.1): "Network Functions Virtualisation (NFV) Release 4; Architectural Framework; Report on further NFV support for 5G".

[i.11] ETSI GS NFV-IFA 014: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Network Service Templates Specification".

[i.12] ETSI GR NFV-TST 012 (V1.1.1): "Network Functions Virtualisation (NFV); Testing; VIM & NFVI Control and Management Performance Evaluation".

[i.13] ETSI GS NFV-TST 010 (V4.3.1): "Network Functions Virtualisation (NFV) Release 4; Testing; API Conformance Testing Specification".

[i.14] ETSI GR NFV-TST 004 (V1.1.2): "Network Functions Virtualisation (NFV); Testing; Guidelines for Test Plan on Path Implementation through NFVI".

[i.15] ETSI GS ZSM 009-1 (V1.1.1): "Zero-touch network and Service Management (ZSM);Closed-Loop Automation; Part 1: Enablers".

[i.16] ETSI GS ZSM 009-2 (V1.1.1): "Zero-touch network and Service Management (ZSM); Closed-Loop Automation; Part 2: Solutions for automation of E2E service and network management use cases".

[i.17] ETSI GR ZSM 009-3 (V1.1.1): "Zero-touch network and Service Management (ZSM); Closed-Loop Automation; Part 3: Advanced topics".

[i.18] ETSI GR NFV-IFA 023 (V3.1.1): "Network Functions Virtualisation (NFV); Management and Orchestration; Report on Policy Management in MANO; Release 3".

[i.19] ETSI TS 128 533 (V18.2.0): "5G; Management and orchestration; Architecture framework (3GPP TS 28.533 version 18.2.0 Release 18)".

[i.20] ETSI TS 128 104: "5G; Management and orchestration; Management Data Analytics (MDA) (3GPP TS 28.104)".

[i.21] ETSI GS NFV-IFA 050: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Intent Management Service Interface and Information Model Specification".

# 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:

**VNF generic OAM function:** PaaS Service that provides in a generic form, OAM capabilities applicable to any kind of VNF

NOTE 1: These PaaS Services aim at covering diverse OAM functionality, such as provisioning, connectivity, configuration and monitoring of one or more VNFs.

NOTE 2: The "any kind of VNF" concerns aspects of supporting diverse VNF implementation approaches and diverse network functionality and services provided by the VNFs.

## 3.2 Symbols

Void.

## 3.3 Abbreviations

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

NOTE: An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in ETSI GR NFV 003 [i.1].

MCCO Managed CIS Cluster Objects
MDAF Management Data Analytics Function

# 4 Overview of the VNF generic OAM functions and PaaS Services framework

## 4.1 Basic concepts

ETSI GR NFV-EVE 019 [i.2] describes three solutions regarding the VNF generic OAM functions architectural framework and its interactions with the rest of the entities of the NFV ecosystem (i.e. NFV-MANO, NFVI, OSS/BSS and VNFs):

- In clause 6.3 of ETSI GR NFV-EVE 019 [i.2] solution A is described, introducing the set of VNF generic OAM functions as a new functional block capable to interact with NFV-MANO, OSS/BSS and the VNF instances.

- In clause 6.4 of ETSI GR NFV-EVE 019 [i.2] solution B is described (with two variations B1 and B2) considering extending existing functional blocks functionality by considering VNF generic OAM functions. For example, VNF generic OAM functions like the *Upgrade VNF* and *VNF Configuration Manager* are sorted as part of EM, while the functionality of other VNF generic OAM functions like the *log aggregator* and the *log* can interact directly with NFV-MANO.

- According to solution C described in clause 6.5 of ETSI GR NFV-EVE 019 [i.2], VNF generic OAM functions can be realized as VNF Common Services, offering services to multiple consumers. The solution considers VNF Common Services are PaaS Services consumed by VNFs, as described in clause 7.1.1.2 of ETSI GR NFV-IFA 029 [i.3].

According to the analysis made in ETSI GR NFV-EVE 019 [i.2] the final recommendations describe that the overall design of the VNF generic OAM functions framework qualify solution A and solution C for further investigation.

According to ETSI GR NFV-IFA 029 [i.3], PaaS Services can be VNF Common Services, VNF Dedicated Services or other services. General concepts of PaaS Services and use cases are documented in ETSI GR NFV-IFA 029 [i.3]; additional use cases about reusing VNF Common/Dedicated Services for the deployment of VNF and NS in the context of 5G networks are also documented in ETSI GR NFV-IFA 037 [i.10].

As analysed and documented in ETSI GR NFV-EVE 019 [i.2], there is a close mapping between VNF generic OAM functions and PaaS Services. In the present document, it is specified that a "VNF generic OAM function" is a "PaaS Service" of a specific type, i.e. a PaaS Service that provides and handles generic OAM functionality for VNFs. As a PaaS Service, a VNF generic OAM function can be either a VNF Common Service or a VNF Dedicated Service. In the following and remaining of the present document, VNF Common/Dedicated Services are also simply referred as "PaaS Services".

In the present document, a framework of VNF generic OAM functions and other PaaS Services is specified.

## 4.2 VNF generic OAM functions and other PaaS Services framework

### 4.2.1 Framework considerations and general functional requirements for VNF generic OAM functions and other PaaS Services

#### 4.2.1.1 General functional requirements for VNF generic OAM functions and other PaaS Services

The functional requirements defined in table 4.2.1.1-1 and table 4.2.1.1-2 are equally applicable to both VM-based and Container-based environments.

Table 4.2.1.1-1 provides general functional requirements for the VNF generic OAM functions and other PaaS Services framework.

**Table 4.2.1.1-1: General functional requirements related to VNF generic OAM functions and other PaaS Services framework**

| Identifier | Recommendation description |
|---|---|
| PaasGen.Fwk.001 | The VNF generic OAM functions and other PaaS Services framework shall unambiguously define basic types of VNF generic OAM functions and other PaaS Services. See note. |
| PaasGen.Fwk.002 | The VNF generic OAM functions and other PaaS Services framework shall support the capability of VNF generic OAM functions and other PaaS Services to interoperate among themselves. See note. |
| NOTE: By defining basic types of VNF generic OAM functions and other PaaS Services and identifying interoperability points, implementations can unambiguously define the functionalities offered by the solutions, even when these can perform an aggregation of the basic types of VNF generic OAM functions and/or other PaaS Services. For example, for the case of VNF generic OAM functions a "VNF metrics manager" could bundle the functionalities of the specified "VNF Metrics Aggregator" and "VNF Metrics Analyser" functions. | |

NOTE 1: From version v5.2.1 of the present document, the requirements were renamed from "VnfGenOam.Fwk.[req#]" to "PaasGen.Fwk.[req#]".

Table 4.2.1.1-2 provides generic functional requirements for the VNF generic OAM functions and other PaaS Services.

**Table 4.2.1.1-2: General functional requirements related to VNF generic OAM functions and other PaaS Services**

| Identifier | Recommendation description |
|---|---|
| PaaS.GenFunc.001 | A VNF generic OAM function or other PaaS Service shall support the capability to expose standard interfaces and operations related to its functionality. |
| PaaS.GenFunc.002 | A VNF generic OAM function or other PaaS Service shall support the capability to be managed by a managing entity. See note 1. |
| PaaS.GenFunc.003 | A VNF generic OAM function or other PaaS Service shall support the capability of being consumed by any authorized entity like VNFs, NFV-MANO entities (FBs or functions), OSS/BSS, or other authorized VNF generic OAM functions or other PaaS Services. |
| PaaS.GenFunc.004 | A VNF generic OAM function or other PaaS Service shall support the capability of being consumed/shared by one or multiple services/entities at a time (e.g. OSS and NFV-MANO). |
| PaaS.GenFunc.005 | A VNF generic OAM function or other PaaS Service shall support the capability to handle multiple entities/instances at a time (e.g. multiple VNFs). |
| PaaS.GenFunc.006 | A VNF generic OAM function or other PaaS Service shall support the capability to support requesting specific operations to NFV-MANO, VNF/VNFC instances (including the VNF's application(s)), NFVI/hosts, and other VNF generic OAM functions or other PaaS Services. |
| PaaS.GenFunc.007 | A VNF generic OAM function or other PaaS Service shall support the capability to have a lifecycle independent of any consumer of its services. |
| PaaS.GenFunc.008 | A VNF generic OAM function or other PaaS Service shall support the capability to be terminated in a graceful manner. |
| PaaS.GenFunc.009 | A VNF generic OAM function or other PaaS Service shall support the capability to be scaled. See note 2. |
| PaaS.GenFunc.010 | A VNF generic OAM function or other PaaS Service shall support the capability to send notifications and alerts to authorized consumers (see notes 3 and 4). |
| PaaS.GenFunc.011 | A PaaS Service shall support the capability to enforce the outputs from policies decisions. |

NOTE 1: This concerns to the "manageability" of the VNF generic OAM function or other PaaS Service. This requirement implies that the VNF generic OAM function or other PaaS Service supports the capability to expose standard interfaces and operations to configure it, to query information and supported functionality and capabilities, to enable the subscription to notifications provided by the VNF generic OAM function or other PaaS Service, and to expose metrics, logs and other information related to the VNF generic OAM function or other PaaS Service, including its lifecycle.
NOTE 2: Irrelevant of the realization of the VNF generic OAM function or other PaaS Service, by being scalable, the VNF generic OAM function or other PaaS Service can adapt for the increasing/decreasing demand of services from/to the VNF generic OAM function or other PaaS Service.
NOTE 3: Notifications can be used to support monitoring actions and troubleshooting, results reporting, etc.
NOTE 4: The Notifications Management Interface exposed by the Notification Manager is specified in clause 6.3.12. Other VNF generic OAM functions or PaaS Services may also support the Notifications Management interface for subscription management to notifications and to send notifications. Other interfaces exposed by VNF generic OAM functions or other PaaS Services to send notifications, if any, are not specified in the present document.

NOTE 2: From version v5.2.1 of the present document, the requirements were renamed from "VnfGenOam.GenFunc.[req#]" to "Paas.GenFunc.[req#]".

### 4.2.1.2 VNF Generic OAM functions and other PaaS Services realization forms

In the present document the following models for realizing VNF Generic OAM functions and other PaaS Services are considered as described in ETSI GR NFV-IFA 029 [i.3]:

a) A VNF Generic OAM function or other PaaS Service can be deployed as a VNF (see also solution C considered in ETSI GR NFV-EVE 019 [i.2]). PaaS Services deployed as VNFs are described in clause 7.1.1.2 of ETSI GR NFV-IFA 029 [i.3].

b) A VNF Generic OAM function or other PaaS Service can be deployed as a NFVI resource (either in the form of one or multiple virtualised resources or as one or multiple managed CIS cluster objects (MCCO)); and

c) A VNF Generic OAM function or other PaaS Service can be deployed as a new object class.

#### 4.2.1.3 Relationship between VNF Generic OAM functions and EMs

VNF generic OAM functions and other PaaS Services can interact with any VNF type and support in a generic way operations related to VNF connectivity, configuration and monitoring.

The use of VNF generic OAM functions and other PaaS Services can relax the restriction that management actions like the configuration of different VNF types is coupled with dedicated management systems (i.e. EMs).

EXAMPLE: VNF generic OAM functions and other PaaS Services can be used to perform EM related operations like VNF configuration and/or exposure of the *Indicator* and *LCM Coordination* interfaces according to ETSI GS NFV-IFA 008 [i.5] in a generic simplified way.

In more detail when it comes to VNF configuration aspects in NFV, VNF configuration is regarded to include two parts, as documented by clause 4.2 of ETSI GR NFV-EVE 022 [i.4]. The first part is about virtualisation-dependent items, and the second part of virtualisation-independent items. The latter are also referred to as "VNF's application configuration". In principle functionalities provided by EMs are targeting VNF's application configuration aspects and lack support for the virtualisation-dependent items. These tasks can be performed by VNF generic OAM functions or other PaaS Services.

Note that scenarios where EMs are used to interact with VNF generic OAM functions and other PaaS Services to deliver VNF configuration functionality are not excluded. For example, an EM can convey VNF configuration to a VNF through the VNF Configuration Manager. EMs can also interact with the Configuration Server PaaS Service see Solution #2 in clause 7.3.3.1.2.2 of ETSI GR NFV-EVE 022 [i.4] describing such interactions.

#### 4.2.1.4 Relationship between VNF Generic OAM functions, other PaaS Services and MDAF

The Management Data Analytics Function (MDAF) enables automation in NFV-MANO. The MDAF was introduced in ETSI GR NFV-IFA 041 [i.6], while the relevant interface specification is in ETSI GS NFV-IFA 047 [i.7].

In the context of the VNF generic OAM functions, data analysis mechanisms have been introduced in the VNF Metrics Analyser and the Logs Analyser functions. The VNF Metrics Analyser function (for metrics) and the Log Analyser function (for logs) expose an interface related to analytics within their scope (see clause 6.3.5 for the interface exposed by the Logs Analyser function and clause 6.3.7 for the interface exposed by the VNF Metrics Analyser). One difference between these two VNF generic OAM functions and MDAF is that the former can interact directly with the VNFs. This is not the case for MDAF which can collect VNF related data through VNFM and the relevant VNF indicators. Second, the VNF generic OAM functions perform analysis in a pre-defined and narrower scope, i.e. either metrics or logs, while the MDAF can perform analytics by collecting and handling data of many other sources and types.

Regarding interactions between MDAF and VNF generic OAM functions like also other PaaS Services as specified by the present document:

- MDAF can be a consumer of the interface exposed by a VNF generic OAM function or other PaaS Services.
  - For example, the VNF Metrics Analyser function and the Log Analyser function can be used to provide input to MDAF by performing analytics pre-processing.
- A VNF generic OAM function or other PaaS Services can be a consumer of the MDA-1 interface exposed by MDAF.
  - For example, the Policy Agent function can receive an analytics report provided by MDAF to perform closed loop control. See clause 4.2.1.7 of the present document for closed loop control in NFV.

#### 4.2.1.5 Notifications management for VNF Generic OAM functions and other PaaS Services

In an NFV environment notifications management can be a highly complex procedure since many different components can raise alerts and notifications when issues are detected from any layer of the protocol stack. From an OSS/BSS perspective, registration to all these monitoring components and handling independently the notification process can be cumbersome.

Irrelevant of the NFVI operational environment (i.e. VM-based, container-based or container on VMs) VNF generic OAM functions and other PaaS Services can send notifications related to events on the associated managed objects.

To simplify however the relevant subscription management and notifications routing from a VNF generic OAM function/PaaS Service consumer perspective, e.g. OSS/BSS a PaaS Service named Notification Manager is used. The Notification Manager offers the functionality to consumers to centralize the subscription to notifications of events produced by VNF generic OAM functions/PaaS Services, and when such events happen, then forward/route the notifications to the consumers.

The Notification Manager exposes in the northbound a generic interface supporting subscription operations and notifications management operations targeting specific or all of the VNF generic OAM functions and other PaaS Services. See clause 4.2.4.3 for more details on the Notification Manager PaaS Service.

EXAMPLE: The VNF Metrics Aggregator function detects an event and sends a notification to the Notification Manager. Then the Notification Manager forwards the notification to the subscriber.

In the PaaS Services management plane, the following operations are supported by PSM and PSR, regarding notifications:

- subscription management to notifications; and
- support notifying subscribers about events related to PaaS Services.

Notifications sent by these entities to notifications subscribers are not forwarded through the Notification Manager, rather be handled independently by PSM (about PaaS Service LCM events) and PSR (about PSD management events) respectively.

#### 4.2.1.6 Policy management for VNF Generic OAM functions and other PaaS Services

A Policy Agent is a PaaS Service offering the functionality to manage policies for other PaaS Services and VNF/VNFC instances (see clause 4.2.4.4). The Policy Agent can interact with VNF generic OAM functions and other PaaS Services for assisting on the execution and decision-making of policies within their respective OAM/service scope:

- Consumers of the Policy Agent functionality can use the interface exposed by the Policy Agent to manage a policy (i.e. create, delete, query a policy). See clause 6.3.13 for the specification of the IF-F1 interface exposed by the Policy Agent.
- For the case of VNFs/VNFCs being assisted by the Policy Agent, see clause 4.2.3 for a description of the IF-V1 and IF-V2 interfaces and clause 8 of the present document for service interaction requirements with VNFs.
- The Policy Agent is not responsible for NFV-MANO policies. Each NFV-MANO FB exposes a policy management interface and is responsible to conform to the policies received by performing the relevant PF operations documented in ETSI GR NFV-IFA 023 [i.18].

#### 4.2.1.7 Closed loop control in NFV

ETSI ZSM framework capabilities regarding closed loop control are defined in ETSI GS ZSM 009-1 [i.15], ETSI GS ZSM 009-2 [i.16] and ETSI GR ZSM 009-3 [i.17]. For example closed loop LCM is defined in ETSI GS ZSM 009-1 [i.15], while ETSI GR ZSM 009-3 [i.17] analyses issues like the design of composition and coordination of interdependent closed loops and dynamic composition of closed loops.

Following the principles described in ETSI GS ZSM 009-1 [i.15], clause 8.1.5.2 for interactions based on zero-touch policies, the Policy Agent can interact with other VNF generic OAM functions, NFVI components and CIS clusters, NFV-MANO, etc. to enable closed loop control in the VNF generic OAM function framework. The Policy Agent gathers input data to evaluate a policy and make a decision.

The Policy Agent can interact with:

- OSS/BSS:
  - It can receive policies from OSS/BSS, targeting the any other VNF generic OAM function or a VNF or sets of VNFs.
  - Report to OSS/BSS about policy decisions.

- Other VNF generic OAM functions:
  - Receive inputs from other VNF generic OAM functions for policies to be executed.
  - Provide outputs to other VNF generic OAM functions about policy decisions.
- MDAF:
  - Receive inputs sent by the MDAF regarding analytics reporting and recommended actions set.
- Intent Management function:
  - It can receive polices from the Intent Management function.

### 4.2.1.8 Testing management for VNFs

In ETSI GR NFV-EVE 019 [i.2], clause 4.3.11, the VNF Testing Manager function has been introduced as a VNF generic OAM function. The VNF Testing Manager is responsible to provide a simple mechanism to orchestrate complex test cases (e.g. VNFC-to-gateway connectivity test) in field deployments, but not only, based on a request from the operator or due to a VNF or NS LCM operation.

NOTE: It is not envisioned that the VNF Testing Manager is a replacement or alternative to tools dedicated to "API conformance testing".

The VNF Testing Manager can be used to support multilayer testing functionalities up to the application layer related to VNF and NFV-MANO operations. Furthermore, VNF testing can be related to intra or inter-NFVI-PoP testing. As a VNF generic OAM function, it can interact with NFV-MANO entities, OSS, NFVI, VNFs and other functions (e.g. MDAF) to support the execution of a test. To successfully support VNF testing, interaction with other VNF generic OAM functions and other PaaS Services is also expected, like for example with VNF Metrics Analyser and the Log Analyser.

In a virtualised and cloudified environment testing can consider the underlay network and/or the overlay network, different sending and receiving entities like VNF-to-VNF, VNFC-to-gateway, Node-to-Node etc. and different virtualisation technologies (e.g. VM-based, container-based).

A VNF Testing Manager can be responsible for testing for one or more domains. Domains can be established in various manners, such as NFV administrative domains e.g. covering also the case of multi-operator environments.

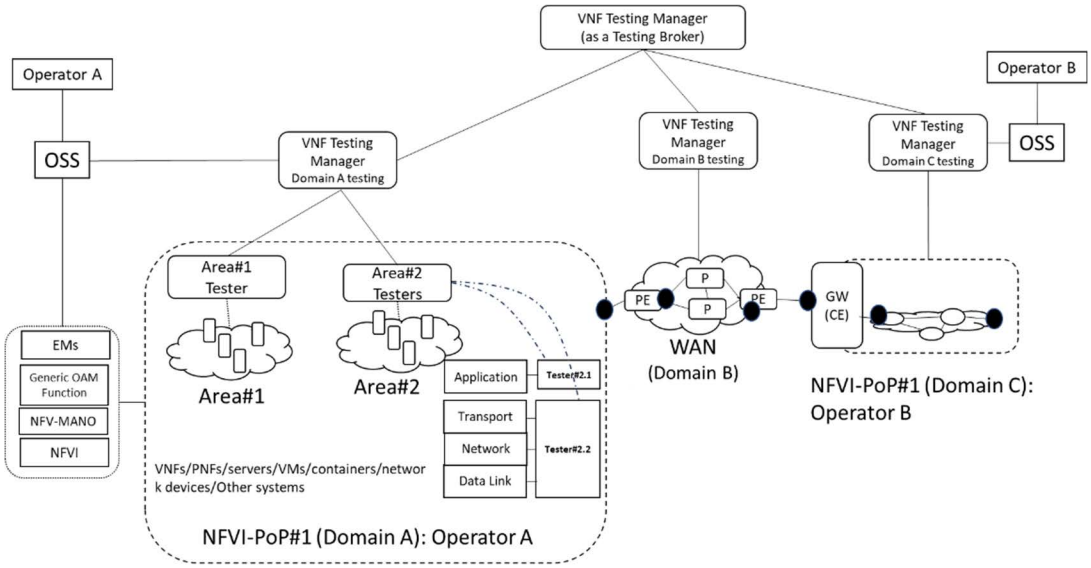


**Figure 4.2.1.8-1: VNF Testing Managers as a Domain Testers and the concept of domain areas**

Figure 4.2.1.8-1 depicts an architecture with multiple VNF Testing Managers acting as domain testers in a multi operator environment. A VNF Testing Manager acting as a domain tester can segmentize its domain into test areas and manage corresponding area testers based on the needs of the experiment.

A test area is a logical abstraction within the domain boundaries and may cover one or more layers of the protocol stack and/or cover different network segment within the domain. For example, the operator can segment an administrative domain as defined in ETSI GR NFV 003 [i.1] into test areas, each with a respective test area controller. Each test area controller controls a respective subset of communication network components and fully or parts of their functionalities.

#### 4.2.1.9 VNF configuration management

As described by ETSI GR NFV-EVE 022 [i.4], VNF configuration encompasses two dimensions: virtualisation-dependent attributes configuration (like the VNF's VNFM IP address), and virtualisation independent attributes configuration (like traffic forwarding rules for a VNF). Both dimensions need to be properly realized to bring the VNF into operation.

NOTE: VNF configuration actions can be performed during provisioning but also during operation.

A single VNF generic OAM function or a combination of VNF generic OAM functions and other PaaS services can be used to support the configuration of a VNF. For example, the Network Configuration Manager can be used to configure the relevant network connectivity aspects, while the VNF Configuration Manager function, can be used to configure application attributes of one or more VNF instances. The VNF Configuration Manager can also interact with a Configuration Server to retrieve configuration data. VNF configuration management can be performed even without VNF generic OAM functions and other PaaS Services. ETSI GR NFV-EVE 022 [i.4] details three different methods to support VNF configuration.

Annex C summarizes the key characteristics of existing VNF configuration methods and describes a potential interplay with VNF Generic OAM functions and other PaaS Services.

### 4.2.2 Framework design of VNF Generic OAM functions and other PaaS Services

VNF generic OAM functions and other PaaS Services provide capabilities to facilitate the deployment and operation of various NFV components, including VNF, NS and NFVI. They can be used to extend the functionality of baseline NFVI (e.g. adding enhancements to CIS Clusters), provide OAM capabilities to VNFs, or provide additional capabilities at NS level (e.g. adding new functionalities to NS VLs). VNF generic OAM functions are simply a subset of PaaS Services that provide OAM capabilities to VNFs or NSs. In other words, if a PaaS Service is related to OAM capabilities for VNFs or NSs it is described as a VNF generic OAM function.

A visual representation of the VNF generic OAM functions and other PaaS Services framework is depicted in figure 4.2.2-1. The design reassembles properties from both Solution A in clause 6.2 and solution C in clause 6.5 of ETSI GR NFV-EVE 019 [i.2].

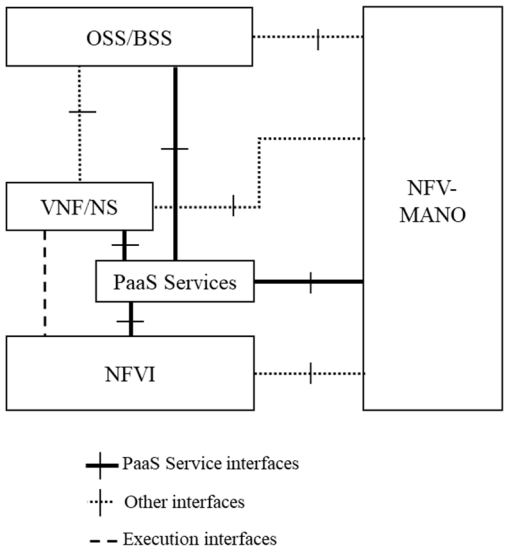


**Figure 4.2.2-1: VNF generic OAM functions and other PaaS Services framework**

NOTE: Figure 4.2.2-1 omits the depiction of interactions between NFV-MANO and external entities, as well as interactions between NFVI and external entities, for the purpose of simplification.

In more detail, a logical grouping of a set of different VNF generic OAM functions and other PaaS Services is considered. The logical set is not a managed object, rather each VNF generic OAM function or other PaaS Service can be operated and managed independently.

As depicted in figure 4.2.2-1 each VNF generic OAM function and other PaaS Service can interact with:

- NFV-MANO, e.g.:
  - in the use cases where the VNF generic OAM functions are managed by NFV-MANO; or
  - receiving a request to isolate a containerized workload; or
  - requesting to perform an operation because of a closed-loop control operation performed by the Policy agent; or
  - interacting with MDAF and final processing of analytics data.
- Operator and OSS/BSS, e.g.:
  - requesting the VNF generic OAM function to retrieve and process logs from the VNF/VNFCs, NFV-MANO and NFVI; or
  - requesting the VNF generic OAM function to retrieve and process VNF specific metrics from the VNF/VNFCs and NFV-MANO; or
  - requesting to distribute configuration to the VNF/VNFCs;
  - when interacting with MDA exposed by 3GPP according to ETSI TS 128 104 [i.20].
- VNF/VNFC/NSs, e.g.:
  - providing logs/VNF-specific metrics; or
  - receiving configuration from the VNF generic OAM function.
- NFVI/host, e.g.:
  - providing logs.
- Other PaaS Services (like other VNF generic OAM functions), e.g.:
  - the Log/VNF metrics analyser processing logs/VNF-specific metrics provided by the Log/VNF metrics aggregator; or
  - the Policy agent interacting with other VNF generic OAM functions to enable closed-loop control and decision making. See closed-loop automation use cases in ETSI TS 128 104 [i.20].

As specified ETSI GS NFV-IFA 010 [i.9], the NFV Architectural Framework can also support the management of PaaS Services. The general requirements of PaaS Services management are specified in clause 5.8 of ETSI GS NFV-IFA 010 [i.9]. The management framework of PaaS Services adds to NFV-MANO two additional management functions, the PaaS Services Management (PSM) and the PaaS Service Repository (PSR). Functional requirements for PSR function and PSM are specified in clause 17 and 18 of ETSI GS NFV-IFA 010 [i.9], respectively.

Figure 4.2.2-2 illustrates the scenario wherein PaaS Services are managed by PSM and PSR within NFV-MANO.

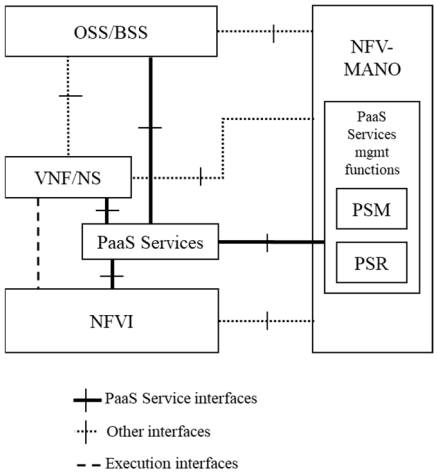


**Figure 4.2.2-2: PaaS Services managed by NFV-MANO**

See ETSI GR NFV-EVE 019 [i.2], clause 5.6 for more examples about interactions between VNF generic OAM function or other PaaS Service and other entities.

### 4.2.3 Interface and service level interactions of VNF generic OAM function and other PaaS Services

For every PaaS Service (including the specific types of VNF generic OAM functions) the following interfaces and service level interactions are considered, as depicted in figure 4.2.3-1.

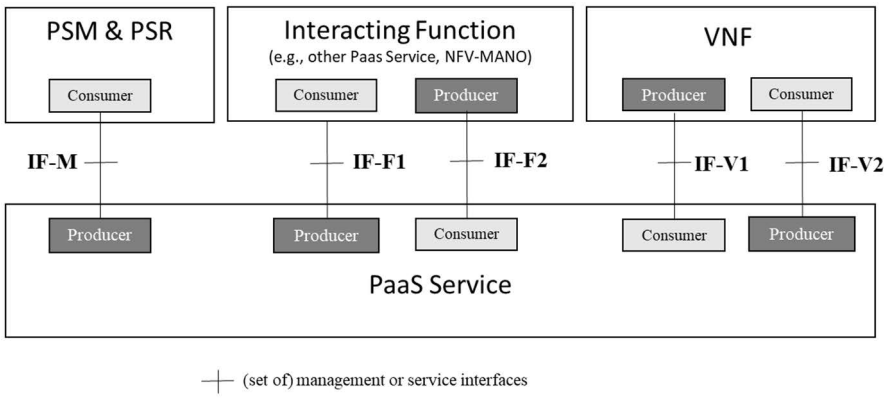


**Figure 4.2.3-1: Interfaces and service level interactions of PaaS Services**

The set of interfaces and service level interactions of a PaaS Service are categorized into two groups:

- it comprises the set of interfaces and service level interactions on which the PaaS Service primarily plays the role of a "producer" of interface and associated services.
- it comprises the set of interfaces and service level interactions on which the PaaS Service primarily plays the role of a "consumer" of interfaces and associated services produced by another entity.

In more detail the following interfaces (or set of interfaces) are considered:

- IF-F1: interface(s) exposed by the PaaS Service with specific functionality tailored to that PaaS Service. Considering the case when the PaaS Service is a VNF generic OAM function, the set of interfaces produced by each VNF generic OAM function are specified in clause 6. Interfaces produced by other PaaS Services are also specified in clause 6, where such a PaaS Service is explicitly defined.
- IF-M: interface(s) exposed by the PaaS Service towards one or multiple corresponding management entity for the purpose of managing the PaaS Service. In case the PaaS Services are managed by NFV-MANO, the consumers of these interfaces are the PSM and PSR.

NOTE: For a PaaS Service deployed as a VNF, VNF configuration methods described in ETSI GR NFV-EVE 022 [i.4] are also applicable for configuring the PaaS Service.

- IF-F2: to perform the intended functionality, a PaaS Service can interact with different function producer entities (i.e. OSS, NFV-MANO, NFVI or other PaaS Services). In the case of inter- PaaS Service interactions, the consumer PaaS Service consumes the IF-F1 interface of the producer PaaS Service. In other cases, what interfaces are consumed by the PaaS Service depend on the specific function producer entity, e.g. when interacting with NFV-MANO functional blocks and functions such as VNFM, CISM or VIM, the PaaS Service consumes the authorized standard interfaces produced by the respective entity.

- IF-V1 and IF-V2: it comprises the interface(s) used to interact with the VNF instances that are being managed by the VNF generic OAM function, and interfaces used to interact with other PaaS Services when such a PaaS Service is associated to other constructs, such as NS instances, but also VNF instances. The relevant service interaction requirements with managed VNF and NS instances are specified in clause 8.

### 4.2.3a VNF generic OAM functions

In the present document, it is specified that a "VNF generic OAM function" is a "PaaS Service" of a specific type, i.e. a PaaS Service that provides and handles generic OAM functionality for VNFs.

The PaaS Services which are qualified as VNF Generic OAM functions, and which are specified in the present document are the following:

- Traffic Enforcer function
- Network Configuration Manager function
- Upgrade VNF function
- Log Aggregator function
- Log Analyser function
- VNF Metrics Aggregator function
- VNF Metrics Analyser function
- Time function
- Configuration Manager function
- VNF Testing Manager function

The relevant use cases and concepts related to these VNF Generic OAM functions are described in ETSI GR NFV-EVE 019 [i.2].

### 4.2.4 Other PaaS Services

#### 4.2.4.1 Overview

Clause 4.2.4 introduces the PaaS Services, which are not qualified as VNF Generic OAM functions, that are further specified in the present document.

Clause 6.8.2.1 of ETSI GS NFV-IFA 014 [i.11] introduces some examples of PaaS Services and their relationship to other constructs such as VNFs and NSs, including VNF generic OAM functions (specified by the present document) and service mesh capabilities. Other PaaS Services are further specified in subsequent clauses.

#### 4.2.4.2 Configuration Server

A Configuration Server is a PaaS Service that offers a logically centralized repository for configuration management purposes. It is referred to be "logically centralized" because the Configuration Server offers a centralized point for distributing and retrieving configuration data. However, this does limit the implementation of the actual storage resources for configuration, which can be multiple and distributed, yet centrally leveraged via the Configuration Server.

A Configuration Server provides additional functionality beyond storage of configuration data such as data processing, data validation, and data format transformation.

The use cases and concepts related to centralized storage for configuration are described in clauses 5.3.4 and 7.3.3 of ETSI GR NFV-EVE 022 [i.4].

#### 4.2.4.3 Notification Manager

A Notification manager is a PaaS Service that offers the functionality to consumers to centralize the subscription to notifications of events produced by other PaaS Services, and when such events happen, then forward/route (with or without processing) the notifications to the consumers.

The present document does not specify requirements for the interfaces to be considered for VNF generic OAM functions and other PaaS Services to send notifications. The Notification manager can communicate over produced interfaces with the VNF generic OAM functions and other PaaS Services to subscribe to notifications or to collect (either on demand or automatically) notifications.

The use cases and concepts related to Notifications manager function are described in ETSI GR NFV-EVE 019 [i.2], wherein it was initially introduced as a VNF generic OAM function.

#### 4.2.4.4 Policy Agent

A Policy Agent is a PaaS Service offering the functionality to manage policies for other PaaS Services and VNF/VNFC instances to execute policies and perform the decision making of policies. The Policy Agent can receive policies for any VNF/VNFC instance and any other PaaS Service.

On the one hand, policy execution concerns to the environment provided by the Policy Agent on which policies conforming to a defined data model are running. On the other hand, policy decision making concerns to evaluating the input/events against the policy and other structured data and delivering a decision as an output. This output can then be used by other PaaS Services to perform the enforcement of the policy.

The Policy Agent does not perform the functionality to enforce the policy. Enforcement of the policy is performed by the respective VNF generic OAM function, other PaaS Service, and/or VNF/VNFC that is assisted by the Policy Agent, according to their functional scope.

The use cases and concepts related to the Policy Agent function are described in ETSI GR NFV-EVE 019 [i.2], wherein it was initially introduced as a VNF generic OAM function.

When a request is made to the Policy Agent to create or update a policy, the Policy Agent stores the policy in a repository, if not already available or updates the policy (see clause 5.11 for the Policy Agent functional requirements).

In summary, for requests made for a PaaS Service through the IF-F1 interface (see clause 4.2.3 for an overview and clause 6.3 for the specification of the IF-F1 interfaces) the PaaS Service interacts with the Policy Agent as follows.

The PaaS Service provides input data like also indicating the policy to conform against to the Policy Agent. The Policy Agent then performs policy evaluation, by evaluating the request against defined policies stored in the repository. It can access various attributes of the request to decide. Based on the evaluation, the Policy Agent makes decisions like deciding new key-value pairs, describes "allow" or "deny" decisions etc. Regarding policy enforcement, the Policy Agent provides the evaluation output to the PaaS Service which enforces the decision (i.e. executes the right action based on the input provided by the Policy Agent, which conforms to the policy).

Based on the functionalities described above, the Policy Agent can also be used to support closed loop control in NFV (see clause 4.2.1.7 of the present document).

NOTE: Which entity defines the PaaS Service policies (e.g. OSS/BSS), how policies are defined, and which language is used to describe a policy, are not specified in the present document.

# 5 Functional requirements for VNF generic OAM functions and other PaaS Services

## 5.1 Introduction

This clause defines functional requirements for VNF generic OAM functions and other PaaS Services.

## 5.2 Functional requirements for VNF generic OAM function Traffic Enforcer

Table 5.2-1 specifies functional requirements applicable to the Traffic Enforcer function based on ETSI GR NFV-EVE 019 [i.2], clause 7.2.

**Table 5.2-1: Functional requirements to the Traffic Enforcer function**

| Numbering | Requirement |
|---|---|
| TrafficEnforcer.001 | The Traffic Enforcer function shall support the capability to perform traffic isolation and traffic rerouting of one or more VNFC instances (see note). |
| NOTE: Traffic isolation can be partial or full (i.e. lowering the traffic sent to a VNFC instance) or full (i.e. blocking the traffic sent to a VNFC instance). | |

## 5.3 Functional requirements for VNF generic OAM function Network Configuration Manager

Table 5.3-1 specifies functional requirements applicable to the Network Configuration Manager function based on ETSI GR NFV-EVE 019 [i.2], clause 7.2.

**Table 5.3-1: Functional requirements to the Network Configuration Manager function**

| Numbering | Requirement |
|---|---|
| NetConfMa.001 | The Network Configuration Manager function shall support the capability to set network configuration information related to one or more VNF/VNFC instances. |
| NetConfMa.002 | The Network Configuration Manager function shall support the capability to interact with a Configuration Server to retrieve configuration data related to its functional scope. |

## 5.4 Functional requirements for VNF generic OAM function Upgrade VNF

Table 5.4-1 specifies functional requirements applicable to the Upgrade VNF function based on ETSI GR NFV-EVE 019 [i.2], clause 7.2.

NOTE: Even though the function is named "Upgrade VNF", the function does not only support the upgrade, but also update procedures of a VNF, and in general on any kind of software modification involved.

**Table 5.4-1: Functional requirements to the Upgrade VNF**

| Numbering | Requirement |
|---|---|
| UpgradeVNF.001 | The Upgrade VNF function shall support the capability to modify the software of a VNF to another version (see note 1). |
| UpgradeVNF.002 | The Upgrade VNF function shall support the capability to add an additional virtual resource to a VNFC instance during the process of VNF upgrading (see note 2). |
| UpgradeVNF.003 | The Upgrade VNF function shall support the capability to coordinate among VNFs when modifying their software to run on another version (see note 3). |
| UpgradeVNF.004 | The Upgrade VNF function shall support the capability to coordinate with other VNF generic OAM functions and/or NFV-MANO during the process of VNF software modification to address specific steps in the process (see note 4). |
| NOTE 1: Support to update software of VNF/VNFC, import new service name, import new certificate for other VNF in load balancer, setting configuration of CP in load balancer, etc.<br>NOTE 2: Support adding CPU or memory, or adding or extending volume to use by extending the storage size, etc.<br>NOTE 3: Reference to software images (VM or OS container images), database schema change, application configuration files, etc.<br>NOTE 4: Examples are: to coordinate with the Network Configuration Manager to add network connectivity to new type of VNF instance, to coordinate with the VNF Configuration Manager to configure existing or new VNFC instances. | |

## 5.5 Functional requirements for VNF generic OAM function Log Aggregator

Table 5.5-1 specifies functional requirements applicable to the Log Aggregator function based on ETSI GR NFV-EVE 019 [i.2], clause 7.2.

**Table 5.5-1: Functional requirements for the Log Aggregator function**

| Numbering | Requirement |
|---|---|
| LogAggregator.001 | The Log Aggregator function shall support the capability to collect different types of logs from different entities like the VNF instances (including the VNF's applications), NFV--MANO or NFVI (compute, storage, network resources) determined by a filter (see note 1). |
| LogAggregator.002 | The Log Aggregator function shall support the capability to pre-process the logs (see note 2). |
| LogAggregator.003 | The Log Aggregator function shall support the capability to aggregate the logs in a configurable manner (see note 3). |
| LogAggregator.004 | The Log Aggregator function shall support the capability to store historical log records (see note 4). |
| LogAggregator.005 | The Log Aggregator function shall support the capability to expose (filtered) logs to authorized consumers. |
| NOTE 1: As an example for the case of VNF/VNFC instances, the filter shall support filtering of VNF/VNFC instances by type of the VNF/VNFC, vendor, host, zone, VNF instance identifier, etc. Also, it shall be able to filter by log attributes metric/log type, severity level, etc.<br>NOTE 2: One form of pre-processing is to harmonize the format of the logs.<br>NOTE 3: Examples of configurable forms of aggregation are to aggregate all logs based on criteria of log level, different instances belonging to the same VNF, VNF instances managed by the same VNFM, etc.<br>NOTE 4: A use case to store historical log records is about using such records for further root-cause analysis. | |

## 5.6 Functional requirements for VNF generic OAM function Log Analyser

Table 5.6-1 specifies functional requirements applicable to the Log Analyser function based on ETSI GR NFV-EVE 019 [i.2], clause 7.2.

**Table 5.6-1: Functional requirements for the Log Analyser function**

| Numbering | Requirement |
|---|---|
| LogAnalyser.001 | The Log Analyser function shall support to analyse and process different types of logs based on a set of analysis functions (see note 1). |
| LogAnalyser.002 | The Log Analyser function shall support configuration of the analytics/processing to be applied (see note 2). |
| LogAnalyser.003 | The Log Analyser function shall support the capability to send notifications based on findings from the analysis of the logs. |
| LogAnalyser.004 | The Log Analyser function shall support the capability to expose analytics results to authorized consumers. |
| NOTE 1: Examples of analysis functions are abnormal behaviour detection, threshold cross, statistical processing, correlation of logs, etc.<br>NOTE 2: Examples of configuration forms of the analytics are set threshold, define the composition of the analytic function from a set of basic analytic functions, etc. | |

## 5.7 Functional requirements for VNF generic OAM function VNF Metrics Aggregator

Table 5.7-1 specifies functional requirements applicable to the VNF Metrics Aggregator function based on ETSI GR NFV-EVE 019 [i.2], clause 7.2.

**Table 5.7-1: Functional requirements for the VNF Metrics Aggregator function**

| Numbering | Requirement |
|---|---|
| VNFMetricAggregator.001 | The VNF Metrics Aggregator function shall support the capability to collect different types of metrics from entities determined by a filter (see notes 1 and 5). |
| VNFMetricAggregator.002 | The VNF Metrics Aggregator function shall support the capability to pre-process the metrics (see note 2). |
| VNFMetricAggregator.003 | The VNF Metrics Aggregator function shall support the capability to aggregate the metrics in a configurable manner (see note 3). |
| VNFMetricAggregator.004 | The VNF Metrics Aggregator function shall support the capability to store time series metrics for records (see note 4). |
| VNFMetricAggregator.005 | The VNF Metrics Aggregator function shall support the capability to expose (filtered) metrics to authorized consumers. |
| NOTE 1: The filter shall support operations like filtering of VNF/VNFC instances by type of the VNF/VNFC, vendor, host, zone, VNF instance identifier, etc. Also it shall be able to filter by metric/log type, severity level, etc.<br>NOTE 2: One form of pre-processing is to harmonize the format of the metrics.<br>NOTE 3: Examples of configurable forms of aggregations are to aggregate all metrics related to performance, aggregate metrics from different instances belonging to the same VNF, aggregate metrics of VNF instances managed by the same VNFM, etc.<br>NOTE 4: Use cases for storing time series of metrics are for instance using the stored metrics for further root-cause analysis, abnormal behaviour detection, etc.<br>NOTE 5: VNF metrics include performance metrics specified in ETSI NFV-IFA 027 [7] (e.g. VNF's network performance metrics, NFVI related metrics), other virtualisation-dependent (e.g. VNF's connectivity) as well as virtualisation-independent metrics like VNF's application metrics, etc. | |

## 5.8 Functional requirements for VNF generic OAM function VNF Metrics Analyser

Table 5.8-1 specifies functional requirements applicable to the VNF Metrics Analyser function based on ETSI GR NFV-EVE 019 [i.2], clause 7.2.

**Table 5.8-1: Functional requirements for the VNF Metrics Analyser function**

| Numbering | Requirement |
|---|---|
| VNFMetricAnalyser.001 | The VNF Metrics Analyser function shall support the capability to analyse and process different types of metrics based on a set of analysis functions (see note 1). |
| VNFMetricAnalyser.002 | The VNF Metrics Analyser function shall support the capability to provide configuration of the analytics/processing of metrics to be applied (see note 2). |
| VNFMetricAnalyser.003 | The VNF Metrics Analyser function shall support the capability to send notifications based on findings from the analysis of the metrics. |
| VNFMetricAnalyser.004 | The VNF Metrics Analyser function shall support the capability to expose the metrics analytics results to authorized consumers. |
| NOTE 1: Examples of analysis functions are abnormal behaviour detection, threshold crossing, statistical processing, etc.<br>NOTE 2: Examples of configuration forms of the analytics are set thresholds, define the composition of the analytic function from a set of basic analytic functions. | |

## 5.9 Functional requirements for VNF generic OAM function Time function

Table 5.9-1 specifies functional requirements applicable to the Time function based on ETSI GR NFV-EVE 019 [i.2], clause 7.2.

**Table 5.9-1: Functional requirements for the Time function**

| Numbering | Requirement |
|---|---|
| TimeFunction.001 | The Time function shall support the capability to manage the synchronization between the system time of VNFs and their components. |
| TimeFunction.002 | The Time function shall support the capability to configure the time protocol(s) used in the system (see note 2). |
| TimeFunction.003 | The Time function shall support the capability to provide notifications and alerts to authorized consumers (see note 3). |
| TimeFunction.004 | The Time function shall support the capability to record and provide logs to authorized consumers. |
| TimeFunction.005 | The Time function shall support the capability to interact with a Configuration Server to retrieve configuration data related to its functional scope. |
| NOTE 1: Synchronization between VNFs is about preserving the time skew within a certain boundary, for the VNFs with time synchronization requirements.<br>NOTE 2: The system is comprised by VNFs with time synchronization requirements and by other entities used to support synchronization (e.g. a master clock server). As an example, for the case of Precision Time Protocol (PTP), examples of configurable parameters are "slaveOnly", "priority1", etc.<br>NOTE 3: Alerts triggered by the Time function can be used to support administrative actions and troubleshooting. | |

## 5.10 Functional requirements for VNF generic OAM function VNF Configuration Manager function

Table 5.10-1 specifies functional requirements applicable to the VNF Configuration Manager function based on ETSI GR NFV-EVE 019 [i.2], clause 7.2.

**Table 5.10-1: Functional requirements for the VNF Configuration Manager function**

| Numbering | Requirement |
|---|---|
| VnfConfigMgmt.001 | The VNF Configuration Manager function shall support the capability to convey configuration information to one or more VNF/VNFC instances (see notes 1, 3 and 4). |
| VnfConfigMgmt.002 | The VNF Configuration Manager function shall support the capability to query configuration information of VNF/VNFC instances (see notes 2 and 3). |
| VnfConfigMgmt.003 | The VNF Configuration Manager function shall support the capability to interact with a Configuration Server to retrieve configuration data related to its functional scope. |
| VnfConfigMgmt.004 | The VNF Configuration Manager function shall support the capability to manage configuration information for one or more VNF/VNFC instances (see notes 1, 4 and 5). |
| NOTE 1: Configuration information includes virtualisation-dependent configurations (e.g. IP addresses set through NFV-MANO mechanisms) and virtualisation-independent configurations (also referred as VNF's application configuration) (e.g. thresholds related to the application load).<br>NOTE 2: A query can be related to the fetching of the value of a specific configuration parameter.<br>NOTE 3: Supporting configuration management of the whole set of configuration information (as indicated in note 1), provides a consistent way to perform any kind of configuration.<br>NOTE 4: The VNF configuration manager does not understand the semantics of the configuration that are conveyed to the VNF/VNFC instances.<br>NOTE 5: Examples of configuration management operations performed by the VNF Configuration Manager function include preparation actions (e.g. create configuration backup) and postprocessing actions (e.g. rollback running configuration). | |

## 5.11 Functional requirements for PaaS Service Policy Agent

Table 5.11-1 specifies functional requirements applicable to the Policy Agent function based on ETSI GR NFV-EVE 019 [i.2], clause 7.2.

**Table 5.11-1: Functional requirements for the Policy Agent function**

| Numbering | Requirement |
|---|---|
| PolicyAgent.001 | The Policy Agent function shall support the capability to support automated decision making to ease administration tasks. See note 1. |
| PolicyAgent.002 | The Policy Agent function shall support the capability to notify the consumers (e.g. OSS/BSS) about events related to policy managements actions. See note 2. |
| PolicyAgent.003 | The Policy Agent function shall support the capability to parse and execute VNF and VNF generic OAM functions policies. See note 3. |
| PolicyAgent.004 | The Policy Agent function shall support the capability to perform CRUD operations for policies upon request from a consumer (see note 4). |
| NOTE 1: The Policy Agent function can interact with other VNF generic OAM functions to perform automated decision making with or without interaction with NFV-MANO components. See use case description in clause 4.4.2.6 of ETSI GR NFV-EVE 019 [i.2].<br>NOTE 2: Notifications sent by the Policy Agent are forwarded to the notifications subscriber (e.g. OSS/BSS) through the Notification Manager. See clause 4.2.1.5.<br>NOTE 3: Refer to description in clause 4.2.1.6 regarding the enforcement of policies.<br>NOTE 4: The Policy Agent upon a request to create, delete or update a policy, updates accordingly a repository where the policies are stored. The policies can target VNF generic OAM functions and other PaaS Services and VNF/VNFC instances. | |

## 5.12 Functional requirements for VNF generic OAM function VNF Testing Manager

Table 5.12-1 specifies functional requirements applicable to the VNF Testing Manager function based on ETSI GR NFV-EVE 019 [i.2], clause 7.2.

**Table 5.12-1: Functional requirements for the VNF Testing Manager**

| Numbering | Requirement |
|---|---|
| TestingManager.001 | The VNF Testing Manager shall support the capability to set testing plans for NFV environments, including VNFs. See notes 1 and 2. |
| TestingManager.002 | The VNF Testing Manager shall support the capability to control the execution of a test upon request from an authorized consumer. See note 3. |
| TestingManager.003 | The VNF Testing Manager shall support the capability to collect testing results from different entities related to testing. See note 2. |
| TestingManager.004 | The VNF Testing Manager shall support the capability to report testing results from different entities related to testing. See note 2. |
| TestingManager.005 | The VNF Testing Manager shall support the capability to interact with other VNF Testing Managers to perform testing beyond the VNF Testing Manager's domain. See note 4. |
| TestingManager.006 | The VNF Testing Manager shall support the capability to analyse the outcomes from the test execution and assess the success or failure of the test. |
| TestingManager.007 | The VNF Testing Manager shall support the capability to determine for each communication network component (e.g. Core network or RAN network) a respective test controller which is able to manage the component with respect to testing; and for each determined controller determine a specification of a sub-test to be performed by this controller. See note 5, note 6 and note 7. |
| TestingManager.008 | The VNF Testing Manager shall support the capability to negotiate with other VNF Testing Managers and different test controllers (e.g. for the Core or the RAN) the sub-test to be performed by the test controller to conform with a policy of the test controller and/or to be feasible with the capabilities supported and the available resources. |
| TestingManager.009 | The VNF Testing Manager shall support the capability to be configured to create test area controllers inside each administrative domain according to the specification of the test. See note 8. |

NOTE 1: The test is about a single component or end-to-end communication testing between two or more components of the NFV communication system (e.g. between VNFs, VNFS and PNFs, etc.). The testing plan modelling shall support identifying the targets (which entities are involved in the test), and the scope of testing (which kind of test and what to test considering multilayer testing). Testing plan also considers setting and applying configuration related to testing (e.g. VNF-to-VNF connectivity testing).

NOTE 2: Configuration of the test can be facilitated by the interaction of VNF Testing Manager with other VNF generic OAM functions like the Network Configuration Manager for the actual network configuration. Similarly for results collection, VNF Testing Manager can interact with other VNF generic OAM functions like the Log Aggregator, etc.

NOTE 3: Control of the execution of a test is about starting, stopping the test, checking status of a test, e.g. to check if it is completed, running, etc.

NOTE 4: For instance, a VNF Testing Manager can be applied for testing within a certain domain or administrative boundary, such as an NFVI-PoP, and the testing plan involve other domains. Communication between VNF Testing Managers can be achieved by other means (e.g. through a testing broker service proxy/gateway).

NOTE 5: Each test controller is able to manage communication network components of a respective domain of the NFV-based communication system associated with the test controller with respect to testing (e.g. RAN domain or the Core domain).

NOTE 6: Each domain of the NFV-based communication system comprises one or more subnetworks and/or one or more layers of the protocol stack of the NFV communication system.

NOTE 7: The domains associated with different test controllers can be domains of different communication network operators (e.g. for enabling testing of functionality related to roaming).

NOTE 8: Each test area controller controls a respective subset of communication network components and fully or parts of their functionalities. For example, the operator can segment (a part of) n administrative domain as defined in ETSI NFV 003 [i.1] (e.g. the Core) into test areas (or zones), each with a respective test area controller.

## 5.13 Functional requirements for PaaS Service Notification Manager

Table 5.13-1 specifies functional requirements applicable to the Notification Manager function based on ETSI GR NFV-EVE 019 [i.2], clause 7.2.

**Table 5.13-1: Functional requirements to the Notification Manager function**

| Numbering | Requirement |
|---|---|
| NotificationManager.001 | The Notification Manager shall support the capability to receive notifications sent by other VNF generic OAM functions and other PaaS Services. |
| NotificationManager.002 | The Notification Manager shall support capability to process (e.g. group, deduplicate) received notifications. See note 1. |
| NotificationManager.003 | The Notification Manager shall support capability to route processed notification to authorized consumers. |
| NotificationManager.004 | The Notifications manager shall support the capability for a consumer to manage subscriptions to notifications about events reported by VNF generic OAM functions and other PaaS Services. See notes 1 and 2. |
| NOTE 1: Notifications sent by VNF generic OAM functions and other PaaS Services to the Notification Manager, can be sent on demand or automatically to the Notification Manager.<br>NOTE 2: Subscription management includes creation and termination of subscriptions to notifications. | |

## 5.14 Functional requirements for PaaS Service: Configuration Server

Table 5.14-1 specifies functional requirements applicable to the Configuration Server.

**Table 5.14-1: Functional requirements of Configuration Server**

| Numbering | Requirement |
|---|---|
| ConfigServer.001 | The Configuration Server shall support the capability of storing configuration data. |
| ConfigServer.002 | The Configuration Server shall support the capability of converting configuration data between different formats. |
| ConfigServer.003 | The Configuration Server should support the capability to validate configuration data based on data schemas. See note 1. |
| ConfigServer.004 | The Configuration Server shall support the capability of version control configuration data. |
| ConfigServer.005 | The Configuration Server shall support the capability of providing the configuration data to consumers that request to fetch such configuration data. |
| ConfigServer.006 | The Configuration Server shall support the capability keep information about the stored configuration data. See note 2. |
| ConfigServer.007 | The Configuration Server shall support the capability to notify about events and changes of configuration data. |
| NOTE 1: Data schemas can be onboarded onto the Configuration Server or provided during operations issued by a consumer.<br>NOTE 2: Information about the stored configuration data includes, but it is not limited to format of the data, last modification date of the data, version of the stored data. | |

# 6 Interfaces for VNF generic OAM functions and other PaaS Services

## 6.1 Introduction

This clause defines interface requirements for VNF generic OAM functions and other PaaS Services.

## 6.2 Interface requirements for VNF generic OAM functions and other PaaS Services

### 6.2.1 Interface requirements for VNF generic OAM function Traffic Enforcer

Table 6.2.1-1 specifies requirements applicable to the Traffic Enforcer function by means of exposed interfaces.

**Table 6.2.1-1: Requirements of the Traffic Enforcer function by means of exposed interfaces**

| Numbering | Requirement |
|---|---|
| TrafficEnforcerInf.001 | The Traffic Enforcer function shall support producing the traffic management Interface. |

Table 6.2.1-2 specifies requirements applicable to the *traffic management Interface* supported by the Traffic Enforcer generic OAM function.

**Table 6.2.1-2: Interface requirements of the Traffic Management Interface**

| Numbering | Requirement |
|---|---|
| TrafficEnf.Trafm.001 | The Traffic Management Interface shall support the blocking and rerouting of traffic indicating selected VNFC Instances. |

### 6.2.2 Interface requirements for VNF generic OAM function Network Configuration Manager

Table 6.2.2-1 specifies requirements applicable to the Network Configuration Manager function by means of exposed interfaces.

**Table 6.2.2-1: Requirements of the Network Configuration Manager function by means of exposed interfaces**

| Numbering | Requirement |
|---|---|
| NetConfMaInf.001 | The Network Configuration Manager function shall support producing the network configuration management Interface. |

Table 6.2.2-2 specifies requirements applicable to the *Network configuration management Interface* supported by the Network Configuration Manager generic OAM function.

**Table 6.2.2-2: Interface requirements of the Network configuration management interface**

| Numbering | Requirement |
|---|---|
| NetConfMa.NetConfm.001 | The Network configuration management interface shall support configuring the network connectivity for the VNF/VNFC instances. |

### 6.2.3 Interface requirements for VNF generic OAM function Upgrade VNF

Table 6.2.3-1 specifies requirements applicable to the Upgrade VNF function by means of exposed interfaces.

**Table 6.2.3-1: Requirements of the Upgrade VNF function by means of exposed interfaces**

| Numbering | Requirement |
|---|---|
| UpgVNFInf.001 | The Upgrade VNF function shall support producing the Upgrade VNF Management Interface. |

Table 6.2.3-2 specifies requirements applicable to the *Upgrade VNF Management Interface* supported by the Upgrade VNF generic OAM function.

**Table 6.2.3-2: Interface requirements of the Upgrade VNF Management Interface**

| Numbering | Requirement |
|---|---|
| UpgVNF.VNFUpgMa.001 | The Upgrade VNF Management Interface shall support coordinating VNFs when modifying their software or configuration (see note 1). |
| UpgVNF.VNFUpgMa.002 | The Upgrade VNF Management Interface shall support onboarding, deleting, and querying of Upgrade VNF files (see note 2). |
| NOTE 1: The coordination might include installing new file(s) to the VNFC instances and/or configuration of resources related to the VNF. Coordination actions are also about adding network connectivity to the new type of VNF instance, i.e. support to update software of VNF/VNFC, import new service name, import new certificate for other VNF in load balancer, setting configuration of CP in load balancer, or trigger the addition of an additional virtual resource to a VNFC instance during the process of VNF upgrading, etc.<br>NOTE 2: The Upgrade VNF files might include configuration files, and modification process and coordination actions related files (e.g. new database schema, additional executables for the VNF). | |

## 6.2.4 Interface requirements for VNF generic OAM function Log Aggregator

Table 6.2.4-1 specifies requirements applicable to the Log Aggregator VNF generic OAM function by means of exposed interfaces.

**Table 6.2.4-1: Requirements for the Log Aggregator function by means of exposed interfaces**

| Numbering | Requirement |
|---|---|
| VNFLogAggregator.001 | The VNF Log Aggregator function shall support producing the Log Exposure Interface (see note). |
| NOTE: Refer to the support of capabilities of log collection by the log aggregator function specified in clause 8.2. | |

Table 6.2.4-2 specifies requirements applicable to the *Log Exposure Interface* supported by the Log Aggregator generic OAM function.

**Table 6.2.4-2: Interface requirements for the Log Exposure Interface**

| Numbering | Requirement |
|---|---|
| LogAggr.Expose.001 | The Log Exposure Interface shall support exposing the logs to authorized consumers. |
| LogAggr.Expose.002 | The Log Exposure Interface shall support the capability to support filtering of the logs. |

## 6.2.5 Interface requirements for VNF generic OAM function Log Analyser

Table 6.2.5-1 specifies requirements applicable to the Log Analyser VNF generic OAM function by means of exposed interfaces.

**Table 6.2.5-1: Requirements for the Log Analyser function by means of exposed interfaces**

| Numbering | Requirement |
|---|---|
| VNFLogAnalyser.001 | The Log Analyser function shall support producing the Log Analysis Exposure Interface. |

Table 6.2.5-2 specifies requirements applicable to the *Log Analysis Exposure Interface* supported by the Log Analyser generic OAM function.

**Table 6.2.5-2: Interface requirements for the Log Analysis Exposure Interface**

| Numbering | Requirement |
|---|---|
| LogAnalyser.Expose.001 | The Log Analysis Exposure Interface shall support exposing the logs analysis results to authorized consumers. |
| LogAnalyser.Expose.002 | The Log Analysis Exposure Interface shall support configuring the processing of logs to be analysed. |

## 6.2.6 Interface requirements for VNF generic OAM function VNF Metrics Aggregator

Table 6.2.6-1 specifies requirements applicable to the Metrics aggregator VNF generic OAM function by means of exposed interfaces.

**Table 6.2.6-1: Requirements for the Metrics Aggregator function by means of exposed interfaces**

| Numbering | Requirement |
|---|---|
| VNFMetricAggregator.001 | The VNF Metrics Aggregator function shall support producing the Metrics Exposure Interface (see note). |
| NOTE: Refer to the support of capabilities of metrics collection by the metrics aggregator function specified in clause 8.2. | |

Table 6.2.6-2 specifies requirements applicable to the Metrics Exposure Interface supported by the Metrics aggregator generic OAM function.

**Table 6.2.6-2: Interface requirements for the Metrics Exposure Interface**

| Numbering | Requirement |
|---|---|
| MetricAggr.Expose.001 | The Metrics Exposure Interface shall support exposing the metrics to authorized consumers. |
| MetricAggr.Expose.002 | The Metrics Exposure Interface shall support the capability to support filtering of the metrics. |

## 6.2.7 Interface requirements for VNF generic OAM function VNF Metrics Analyser

Table 6.2.7-1 specifies requirements applicable to the Metrics analyser VNF generic OAM function by means of exposed interfaces.

**Table 6.2.7-1: Requirements for the Metrics Analyser function by means of exposed interfaces**

| Numbering | Requirement |
|---|---|
| VNFMetricAnalyser.001 | The VNF Metrics Analyser function shall support producing the Metrics Analysis Exposure Interface. |

Table 6.2.7-2 specifies requirements applicable to the *Metrics Analysis Exposure Interface* supported by the Metrics analyser VNF generic OAM function.

**Table 6.2.7-2: Interface requirements for the Metrics Analysis Exposure Interface**

| Numbering | Requirement |
|---|---|
| MetricAnalyser.Expose.001 | The Metrics Analysis Exposure Interface shall support exposing the metrics analysis results to authorized consumers. |
| MetricAnalyser.Expose.002 | The Metrics Analysis Exposure Interface shall support configuring the processing of metrics to be analysed. |

## 6.2.8 Interface requirements for VNF generic OAM function Time function

Table 6.2.8-1 specifies requirements applicable to the Time function by means of exposed interfaces.

**Table 6.2.8-1: Requirements for the Time function by means of exposed interfaces**

| Numbering | Requirement |
|---|---|
| TimeFuncInf.001 | The Time function shall support producing the Time management interface. |

Table 6.2.8-2 specifies requirements applicable to the *Time management interface* supported by the Time function.

**Table 6.2.8-2: Interface requirements for the Time management interface**

| Numbering | Requirement |
|---|---|
| TimeFunc.Mgmt.001 | The Time management interface shall support configuring the protocols used for time synchronization for VNF/VNFC instances with time synchronization requirements. |
| TimeFunc.Mgmt.002 | The Time management interface shall support setting the time for VNF/VNFC instances with time synchronization requirements (see note 1). |
| TimeFunc.Mgmt.003 | The Time management interface shall support exposing logs related to time protocol operations on VNF/VNFC instances (see note 2). |
| NOTE 1: Setting the time can involve different actions such as: providing specific time values to be set or forcing to synchronize the time to a source.<br>NOTE 2: Time function can interact with the Log Aggregator function for logs collection related to time protocols operations. | |

## 6.2.9 Interface requirements for VNF generic OAM function VNF Configuration Manager

Table 6.2.9-1 specifies requirements applicable to the VNF Configuration Manager function by means of exposed interfaces.

**Table 6.2.9-1: Requirements for the VNF Configuration Manager function by means of exposed interfaces**

| Numbering | Requirement |
|---|---|
| VnfConfigMgmtFuncInf.001 | The VNF Configuration Manager function shall support producing the VNF configuration management Interface. |

Table 6.2.9-2 specifies requirements applicable to the *VNF configuration management Interface* supported by the VNF generic OAM function VNF Configuration Manager.

**Table 6.2.9-2: Interface requirements for the VNF configuration management interface**

| Numbering | Requirement |
|---|---|
| VnfConfigMgmt.Conf.001 | The VNF configuration management interface shall support configuring VNF/VNFC instances. |
| VnfConfigMgmt.Conf.002 | The VNF configuration management interface shall support querying configuration information of VNF/VNFC instances. |
| VnfConfigMgmt.Conf.003 | The VNF configuration management interface shall support backing-up configuration information of VNF/VNFC instances. |

## 6.2.10 Interface requirements for VNF generic OAM function VNF Testing Manager

Methods and metrics for the evaluation of VIM and NFVI control and management performance are described in ETSI GR NFV-TST 012 [i.12]. API conformance regarding functionality test in an automated way for ETSI NFV APIs is specified in ETSI GS NFV-TST 010 [i.13]. Guidelines for test planning of path implementation through NFVI is in ETSI GR NFV-TST 004 [i.14].

The VNF Testing Manager can be used to provide a mechanism for the test configuration, execution and results collection. The planning phase of a test can consider the guidelines proposed by the referenced deliverables.

Table 6.2.10-1 specifies requirements applicable to the VNF Testing Manager function by means of exposed interfaces.

**Table 6.2.10-1: Requirements of the VNF Testing Manager function by means of exposed interfaces**

| Numbering | Requirement |
|---|---|
| TestManagerInf.001 | The VNF Testing Manager shall support producing the VNF Testing Management Interface. |

Table 6.2.10-2 specifies requirements applicable to the VNF Testing Management Interface supported by the VNF Testing Manager VNF generic OAM function.

**Table 6.2.10-2: Interface requirements for the VNF Testing Management Interface**

| Numbering | Requirement |
|---|---|
| VnfTest.VnfTestMa.001 | The VNF Testing Management Interface shall support setting testing plans. See note 1 and note 2. |
| VnfTest.VnfTestMa.002 | The VNF Testing Management Interface shall support requesting to control the execution of a test from authorized consumers. See notes 3 and 4. |
| VnfTest.VnfTestMa.003 | The VNF Testing Management Interface shall support querying the status of a test. |
| VnfTest.VnfTestMa.004 | The VNF Testing Management Interface shall support reporting testing results to authorized consumers. See note 5 and note 6. |
| VnfTest.VnfTestMa.005 | The VNF Testing Management Interface shall support enabling negotiation between VNF Testing Managers acting as administrative domain testers. See note 7. |
| NOTE 1: The testing plan modelling shall support identifying the targets (which entities are involved in the test), and the scope of testing (which kind of test and what to test considering multilayer testing). Testing plan also considers setting and applying configuration related to testing (e.g. VNF-to-VNF connectivity testing). | |
| NOTE 2: Entities involved in a VNF test can be VNFs, PNFs, other NFV-MANO entities, NFVI entities (e.g. switches, routers), CIS cluster components, VNF generic OAM functions, etc. | |
| NOTE 3: Example of an authorized consumer is OSS/BSS. The testing manager can also trigger the execution of a test due to a VNF or NS LCM operation, for example to conform to a policy. The relevant information modelling needs to support the mechanism to setup criteria of when to activate a test. | |
| NOTE 4: Control of execution of a test is about starting, stopping the test, checking status of a test, e.g. to check if it is completed, running, etc. | |
| NOTE 5: Testing results collection depends on the libraries, tools and the relevant APIs exposed by the entities used to support the test. | |
| NOTE 6: The VNF Testing Manager interacts with all the appropriate management systems to support configuration of the test, execution of the test and results retrieval. It also has the ability to correlate the testing results from different VNF Testing managers after the test execution. | |
| NOTE 7: The VNF Testing Manager is able to perform end-to-end test synthesis based on the testing intent specification, constraints, policies etc. through negotiation with the other VNF Testing Managers. For example, VNF Testing managers can negotiate the network demarcation points between different management components (e.g. customer edge (CE) gateway). | |

## 6.2.11 Interface requirements for PaaS Service Notification Manager

Table 6.2.11-1 specifies requirements applicable to the Notification Manager function by means of exposed interfaces.

**Table 6.2.11-1: Requirements of the Notification Manager function by means of exposed interfaces**

| Numbering | Requirement |
|---|---|
| NotificationManagerInf.001 | The Notification Manager function shall support producing the Notifications Management Interface. |

Table 6.2.11-2 specifies requirements applicable to the *Notifications Management Interface* supported by the Notification Manager.

**Table 6.2.11-2: Interface requirements for the Notifications Management Interface**

| Numbering | Requirement |
|---|---|
| Notif.Manager.Notif.Mgmt.001 | The Notifications Management Interface shall support management of subscriptions to notifications. |
| Notif.Manager.Notif.Mgmt.002 | The Notifications Management Interface shall support sending processed notifications to authorized consumers. |

## 6.2.12 Interface requirements for PaaS Service Policy Agent

Table 6.2.12-1 specifies requirements applicable to the Policy Agent function by means of exposed interfaces.

NOTE: The present document version does not specify interface requirements and interface operations regarding the execution of the policies.

**Table 6.2.12-1: Requirements of the Policy Agent function by means of exposed interfaces**

| Numbering | Requirement |
|---|---|
| VnfPolicyAgentInf.001 | The Policy Agent function shall support producing the policy management Interface. |

Table 6.2.12-2 specifies requirements applicable to the *Policy Management Interface* supported by the Policy Agent.

**Table 6.2.12-2: Interface requirements for the Policy Management Interface**

| Numbering | Requirement |
|---|---|
| PolicyAgent.PolicyMgmt.001 | The Policy Management Interface shall support to manage VNFs, VNF generic OAM functions and other PaaS Service policies through the Policy Agent. See note. |
| NOTE: Management of policies includes creation, deletion and query information of created policies. | |

## 6.2.13 Interface requirements for PaaS Service Configuration Server

Table 6.2.13-1 specifies requirements applicable to the Configuration Server by means of exposed interfaces.

**Table 6.2.13-1: Requirements of the Configuration Server by means of exposed interfaces**

| Numbering | Requirement |
|---|---|
| ConfigServerInf.001 | The Configuration Server shall produce the Configuration Data Management interface. |

Table 6.2.13-2 specifies requirements applicable to the Configuration Data Management interface produced by the Configuration Server.

**Table 6.2.13-2: Interface requirements of the Configuration Data Management interface**

| Numbering | Requirement |
|---|---|
| ConfigServer.Cdm.001 | The Configuration Data Management interface shall support transferring to the Configuration Server configuration data to be stored. |
| ConfigServer.Cdm.002 | The Configuration Data Management interface shall support updating stored configuration data. |
| ConfigServer.Cdm.003 | The Configuration Data Management interface shall support deleting stored configuration data. |
| ConfigServer.Cmd.004 | The Configuration Data Management interface shall support fetching stored configuration data. |
| ConfigServer.Cmd.005 | The Configuration Data Management interface shall support querying information about the stored configuration data. |
| ConfigServer.Cmd.006 | The Configuration Data Management interface shall support requesting to convert stored configuration data to a different format. |
| ConfigServer.Cmd.007 | The Configuration Data Management interface should support requesting to validate the configuration data based on referenced data schemas. |
| ConfigServer.Cmd.008 | The Configuration Data Management interface shall support subscribing to notifications about configuration data management events and changes of configuration data. |
| ConfigServer.Cmd.009 | The Configuration Data Management interface shall support notifying subscribers about events related to configuration data management events and changes of configuration data. |

# 6.3 Interface operations

## 6.3.1 Traffic Management Interface

### 6.3.1.1 Description

This interface enables a consumer to manage traffic of one or more VNFC instances.

### 6.3.1.2 Traffic Management

#### 6.3.1.2.1 Description

This operation enables the consumer to request to isolate from and reconnect traffic towards selected VNFC instances.

Table 6.3.1.2.1-1 lists the information flow exchanged between the consumer, e.g. another VNF Generic OAM function, NFV-MANO, or other management entities and the producer, that is the Traffic Enforcer.

**Table 6.3.1.2.1-1: Traffic Management operation**

| Message | Requirement | Direction |
|---|---|---|
| TrafMaRequest | Mandatory | Consumer → Traffic Enforcer |
| TrafMaResponse | Mandatory | Traffic Enforcer → Consumer |

#### 6.3.1.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.1.2.2-1.

**Table 6.3.1.2.2-1: Traffic management operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| vnfcInstanceId | M | 1..N | Identifier | Identifier of the VNFC instance(s) requiring traffic management. |
| targetAction | M | 1 | Enum | Specifies the traffic control action.<br>VALUES:<br>• FULL ISOLATE: Isolate specified VNFC instances from full incoming traffic<br>• PARTIAL ISOLATE: Isolate specified VNFC instances from partial incoming traffic (see note)<br>• RECONNECT: Reconnect specified VNFC instances for incoming traffic |
| NOTE: Traffic isolation can be partial (i.e. blocking traffic destined to specific ports) or full (i.e. blocking the traffic sent to a VNFC instance). | | | | |

#### 6.3.1.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.1.2.3-1.

**Table 6.3.1.2.3-1: Traffic management operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| operationalResult | M | 1 | Not specified | Information about the traffic isolation and reconnection. |

#### 6.3.1.2.4 Operation results

After successful operation, the traffic of affected VNFC instances has been successfully controlled, e.g. the indicated VNFC instances have been traffic isolated or reconnected, dependent on the requested target action. The result of the operation indicates if it has been successful or not with a standard success/error result.

## 6.3.2 Network Configuration Management Interface

### 6.3.2.1 Description

This interface enables a consumer to set network configuration information to one or more VNF/VNFC instances.

### 6.3.2.2 Network configuration management

#### 6.3.2.2.1 Description

This operation enables the generic OAM function to configure the network connectivity for the VNF/VNFC instances.

Table 6.3.2.2.1-1 lists the information flow exchanged between the consumer, e.g. another VNF Generic OAM function, NFV-MANO, or other management entities and the producer, that is the Network Configuration Manager.

**Table 6.3.2.2.1-1: Network configuration management operation**

| Message | Requirement | Direction |
|---|---|---|
| NetConfMa Request | Mandatory | Consumer → Network Configuration Manager |
| NetConfMa Response | Mandatory | Network Configuration Manager → Consumer |

#### 6.3.2.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.2.2.2-1.

**Table 6.3.2.2.2-1: Network configuration management input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| vnfInstanceId | M | 0..1 | Identifier | Identifier of the VNF instance for which the network connectivity is to be configured (see notes 1 and 2). |
| vnfcInstanceId | M | 0..1 | Identifier | Identifier of the VNFC instance for which the network connectivity is to be configured (see notes 1 and 2). |
| cpConfig | M | 1..N | CpConfigInfo | The external CP instance(s) of the VNF/VNFC instance and the configuration information to be applied. |
| meshConfig | M | 0..1 | Not specified | This attribute contains parameters necessary for configuring the behaviour and features for a VNF/VNFC within a service mesh network environment. |
| NOTE 1: Only one of vnfInstanceId and vnfcInstanceId shall be present.<br>NOTE 2: Whether the Identifier specifies a set of VNFs/VNFCs will be investigated as part of stage 3 activities. | | | | |

#### 6.3.2.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.2.2.3-1.

**Table 6.3.2.2.3-1: Network configuration management output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| operationalResult | M | 1 | Not specified | Information about the network configuring. |

#### 6.3.2.2.4 Operation results

After successful operation, Network Configuration Manager notifies to consumer about completion of network configuration. The result of the operation indicates if it has been successful or not with a standard success/error result.

## 6.3.3 Upgrade VNF Management Interface

### 6.3.3.1 Description

This interface shall allow the consumer to request Upgrade VNF operations to be performed by the Upgrade VNF function.

The Upgrade VNF management interface shall support the following operations:

- Upgrade VNF management operation
- Onboarding Upgrade VNF files
- Deleting Upgrade VNF files
- Querying Upgrade VNF files

### 6.3.3.2 Upgrade VNF management operation

#### 6.3.3.2.1 Description

This operation enables the consumer to request to upgrade VNF/VNFC instances with new software version, virtualised resources, configurations, etc.

Table 6.3.3.2.1-1 lists the information flow exchanged between the consumer, e.g. another VNF Generic OAM function, NFV-MANO, or other management entities and the producer, that is the Upgrade VNF function.

**Table 6.3.3.2.1-1: Upgrade VNF management operation**

| Message | Requirement | Direction |
|---|---|---|
| upgradeVNFRequest | Mandatory | Consumer → Upgrade VNF function |
| upgradeVNFResponse | Mandatory | Upgrade VNF function → Consumer |

#### 6.3.3.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.2.2-1.

**Table 6.3.3.2.2-1: Upgrade VNF management input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| vnfInstanceId | M | 0..1 | Identifier | Identifier of the VNF instance to be upgraded (see note 1). |
| vnfcInstanceId | M | 0..1 | Identifier | Identifier of the VNFC instance to be upgraded (see note 1). |
| configurableProperties | M | 0..N | KeyValuePair | Updated values for configurable properties (see note 2). |
| fileInfoId | M | 1..N | Identifier | Identifier of the Upgrade VNF file define by the file provider. |
| NOTE 1: Only one of vnfInstanceId and vnfcInstanceId shall be present.<br>NOTE 2: During the upgrade process changes of connectivity can be expected and the appropriate input parameters can be considered. | | | | |

#### 6.3.3.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.3.2.3-1.

**Table 6.3.3.2.3-1: Upgrade VNF management output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| UpgradeResult | M | 1 | Not specified | Information about the upgrade result. |

#### 6.3.3.2.4 Operation results

After successful operation, the Upgrade VNF function notifies to consumer about completion of the upgrade. The result of the operation indicates if it has been successful or not with a standard success/error result.

### 6.3.3.3 Onboarding Upgrade VNF files

#### 6.3.3.3.1 Description

This operation enables the consumer to request to onboard Upgrade VNF files for the VNF/VNFC instances.

Table 6.3.3.3.1-1 lists the information flow exchanged between the consumer, e.g. another Generic OAM function, NFV-MANO, or other management entities and the producer, that is the Upgrade VNF function.

**Table 6.3.3.3.1-1: Onboarding Upgrade VNF files operation**

| Message | Requirement | Direction |
|---|---|---|
| OnboardVNFUpgFileRequest | Mandatory | Consumer → Upgrade VNF function |
| OnboardVNFUpgFileResponse | Mandatory | Upgrade VNF function → Consumer |

#### 6.3.3.3.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.3.2-1.

**Table 6.3.3.3.2-1: Onboarding Upgrade VNF files input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| upgradeFileInfo | M | 1..N | UpgFileData | Information about Upgrade VNF file(s). |

#### 6.3.3.3.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.3.3.3-1.

**Table 6.3.3.3.3-1: Onboarding Upgrade VNF files output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| fileInfoId | M | 0..N | Identifier | Identifier(s) of the on-boarded Upgrade VNF file(s) |

#### 6.3.3.3.4 Operation results

After successful operation, the Upgrade VNF function notifies to consumer about completion of onboarding Upgrade VNF files. The result of the operation indicates if it has been successful or not with a standard success/error result.

### 6.3.3.4 Deleting Upgrade VNF files

#### 6.3.3.4.1 Description

This operation enables the consumer to delete Upgrade VNF files for the VNF/VNFC instances.

Table 6.3.3.4.1-1 lists the information flow exchanged between the consumer, e.g. another Generic OAM function, NFV-MANO, or other management entities and the producer, that is the Upgrade VNF function.

**Table 6.3.3.4.1-1: Deleting Upgrade VNF files operation**

| Message | Requirement | Direction |
|---|---|---|
| DeleteVNFUpgFileRequest | Mandatory | Consumer → Upgrade VNF function |
| DeleteVNFUpgFileResponse | Mandatory | Upgrade VNF function → Consumer |

#### 6.3.3.4.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.4.2-1.

**Table 6.3.3.4.2-1: Deleting Upgrade VNF files input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| fileInfoId | M | 1..N | Identifier | Identifier(s) for the Upgrade VNF file(s). |

#### 6.3.3.4.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.3.4.3-1.

**Table 6.3.3.4.3-1: Deleting Upgrade VNF files output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| deletedFileInfoId | M | 0..N | Identifier | Identifier(s) of the deleted Upgrade VNF file. |

#### 6.3.3.4.4 Operation results

After successful operation, the Upgrade VNF function notifies to consumer about completion of deleting Upgrade VNF files. The result of the operation indicates if it has been successful or not with a standard success/error result.

### 6.3.3.5 Querying Upgrade VNF files

#### 6.3.3.5.1 Description

This operation enables consumer to query the Upgrade VNF function about Upgrade VNF files for the VNF/VNFC instances.

Table 6.3.3.5.1-1 lists the information flow exchanged between the consumer, e.g. another Generic OAM function, NFV-MANO, or other management entities and the producer, that is the Upgrade VNF function.

**Table 6.3.3.5.1-1: Querying Upgrade VNF files operation**

| Message | Requirement | Direction |
|---|---|---|
| QueryVNFUpgFIleRequest | Mandatory | Consumer → Upgrade VNF function |
| QueryVNFUpgFIleResponse | Mandatory | Upgrade VNF function → Consumer |

#### 6.3.3.5.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.5.2-1.

**Table 6.3.3.5.2-1: Querying Upgrade VNF files input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| filter | M | 1 | Not specified | Filtering criteria to select one or more Upgrade VNF file(s) information (see note). |
| NOTE: Specification of filtering mechanism is part of the protocol design. | | | | |

#### 6.3.3.5.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.3.5.3-1.

**Table 6.3.3.5.3-1: Querying Upgrade VNF files output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| queryFileInfoResult | M | 0..N | FileInfo | Information about the Upgrade VNF file(s) matching the query. |
| NOTE: The lower cardinality is 0 since there may be no matches to the provided filter. | | | | |

#### 6.3.3.5.4 Operation results

After successful operation, the consumer has queried the Upgrade VNF file querying results. The result of the operation indicates if it has been successful or not with a standard success/error result. For a particular query, information about the Upgrade VNF file querying result that are matching with the filter shall be returned.

### 6.3.4 Log Aggregator Exposure Interface

#### 6.3.4.1 Description

This interface enables a consumer to access log aggregator results.

#### 6.3.4.2 Exposing Log Aggregator results

##### 6.3.4.2.1 Description

This operation enables the consumer to request the Log Aggregator to expose the aggregated log results.

Table 6.3.4.2.1-1 lists the information flow exchanged between the consumer, e.g. another VNF generic OAM function, NFV-MANO, or other management entities and the producer, that is the VNF log aggregator.

**Table 6.3.4.2.1-1: Exposing Log Aggregator result operation**

| Message | Requirement | Direction |
|---|---|---|
| LogRequest | Mandatory | Consumer → Log Aggregator |
| LogResponse | Mandatory | Log Aggregator → Consumer |

##### 6.3.4.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.4.2.2-1.

**Table 6.3.4.2.2-1: Exposing Log Aggregator result operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| Filter | M | 1 | Not specified | Filtering criteria to select one or a set of logs (see note) |
| NOTE: Specification of filtering mechanism is part of the protocol design. | | | | |

##### 6.3.4.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.4.2.3-1.

**Table 6.3.4.2.3-1: Exposing Log Aggregator result operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| queryResult | M | 0..N | LogAggregateOutput | Information about the log(s) matching the query. |

#### 6.3.4.2.4 Output parameters

After successful operation, the consumer has queried the log aggregator results. The result of the operation indicates if it has been successful or not with a standard success/error result. For a particular query, information about the log aggregator result that are matching the filter shall be returned.

## 6.3.5 Log Analysis Exposure Interface

### 6.3.5.1 Description

This interface enables a consumer to access log analysis results.

### 6.3.5.2 Exposing Log Analysis results

#### 6.3.5.2.1 Description

This operation enables the consumer to request the log analysis results from the Log Analyser.

Table 6.3.5.2.1-1 lists the information flow exchanged between the consumer, e.g. another VNF generic OAM function, NFV-MANO, or other management entities and the producer, that is the Log Analyser.

**Table 6.3.5.2.1-1: Exposing Log analysis result operation**

| Message | Requirement | Direction |
|---|---|---|
| LogInfoRequest | Mandatory | Consumer → Log Analyser |
| LogInfoResponse | Mandatory | Log Analyser → Consumer |

#### 6.3.5.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.5.2.2-1.

**Table 6.3.5.2.2-1: Exposing Log analysis result operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| Filter | M | 1 | Not specified | Filtering criteria to select one or a set of log analysis result (see note). |
| LogAnalysisConfig | M | 1 | Not specified | The configuration for the log analysis. The configuration supports the analysis functions and processing rules. |
| NOTE: Specification of filtering mechanism is part of the protocol design. | | | | |

#### 6.3.5.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.5.2.3-1.

**Table 6.3.5.2.3-1: Exposing Log analysis result operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| queryResult | M | 0..N | LogAnalysisOutput | Information about the logs matching the query. |

#### 6.3.5.2.4 Operation results

After successful operation, the consumer has queried the log analyser results. The result of the operation indicates if it has been successful or not with a standard success/error result. For a particular query, information about the log analysis result that are matching with the filter shall be returned.

## 6.3.6 Metrics Exposure Interface

### 6.3.6.1 Description

This interface enables a consumer to access metrics aggregator results.

### 6.3.6.2 Exposing Metrics aggregator results

#### 6.3.6.2.1 Description

This operation enables VNF Metrics Aggregator to expose the metrics aggregator results to an authorized consumer.

Table 6.3.6.2.1-1 lists the information flow exchanged between the consumer, e.g. another VNF generic OAM function, NFV-MANO, or other management entities and the producer, that is the VNF Metrics Aggregator.

**Table 6.3.6.2.1-1: Exposing Metrics aggregator result operation**

| Message | Requirement | Direction |
|---|---|---|
| MetricRequest | Mandatory | Consumer → VNF Metrics Aggregator |
| MetricResponse | Mandatory | VNF Metrics Aggregator → Consumer |

#### 6.3.6.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.6.2.2-1.

**Table 6.3.6.2.2-1: Exposing Metrics aggregator result operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| Filter | M | 1 | Not specified | Filtering criteria to select one or a set of metrics.<br>The filter shall support to specify which vnf instances the metrics information is requested to be collected, and to be capable to aggregate all metrics related to performance, aggregate metrics from different instances belonging to the same VNF, aggregate metrics of VNF instances managed by the same VNFM, etc. (see note). |
| NOTE: Specification of filtering mechanism is part of the protocol design. | | | | |

#### 6.3.6.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.6.2.3-1.

**Table 6.3.6.2.3-1: Exposing Metrics aggregator result operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| queryResult | M | 0..N | Not specified | Information about the metrics matching the query. |

#### 6.3.6.2.4 Operation results

After successful operation, the consumer has queried the metrics aggregator result. The result of the operation indicates if it has been successful or not with a standard success/error result. For a particular query, information about the metrics aggregator result that are matching the filter shall be returned.

## 6.3.7 Metrics Analysis Exposure Interface

### 6.3.7.1 Description

This interface enables a consumer to access metrics analysis results.

### 6.3.7.2 Exposing Metrics Analysis results

#### 6.3.7.2.1 Description

This operation enables the consumer to request the VNF Metrics Analyser function to expose the metrics analysis results to an authorized consumer.

Table 6.3.7.2.1-1 lists the information flow exchanged between the consumer, e.g. another VNF generic OAM function, NFV-MANO, or other management entities and the producer, that is the VNF Metrics Analyser.

**Table 6.3.7.2.1-1: Exposing Metrics analysis result operation**

| Message | Requirement | Direction |
|---|---|---|
| MetricInfoRequest | Mandatory | Consumer → VNF Metrics Analyser |
| MetricInfoResponse | Mandatory | VNF Metrics Analyser → Consumer |

#### 6.3.7.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.7.2.2-1.

**Table 6.3.7.2.2-1: Exposing Metrics analysis result operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| Filter | M | 1 | Not specified | Filtering criteria to select one or a set of metrics analysis result.<br>The filter to specify on which vnf instances the metrics information is requested to be collected (see note). |
| MetricsAnalysisConfig | M | 1 | Not specified | The configuration for the metrics analysis. The configuration shall support specifying the metrics to be analysed, the analysis functions, and metrics thresholds. |
| NOTE: Specification of filtering mechanism is part of the protocol design. | | | | |

#### 6.3.7.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.7.2.3-1.

**Table 6.3.7.2.3-1: Exposing Metrics analysis result operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| queryResult | M | 0..N | MetricsAnalysisOutput | Information about the metrics matching the query. |

#### 6.3.7.2.4 Operation results

After successful operation, the consumer has queried the metric analyser results. The result of the operation indicates if it has been successful or not with a standard success/error result. For a particular query, information about the metrics analysis result that are matching with the filter shall be returned.

## 6.3.8 Time Management Interface

### 6.3.8.1 Description

The Time management interface enables the consumer to request the Time function to configure parameters relate to time synchronization to the VNF/VNFC instances.

### 6.3.8.2 Time function configuration

#### 6.3.8.2.1 Description

This operation enables the consumer to request to configure time protocols and/or setting the time function for VNF/VNFC instances with time synchronization requirements.

Table 6.3.8.2.1-1 lists the information flow exchanged between the Time function and a consumer of the interface, e.g. OSS/BSS, another Generic OAM function, NFV-MANO, or other management entities.

**Table 6.3.8.2.1-1: Time function configuration operation**

| Message | Requirement | Direction |
|---|---|---|
| TimeConfigRequest | Mandatory | Consumer → Time function |
| TimeConfigResponse | Mandatory | Time function → Consumer |

#### 6.3.8.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.8.2.2-1.

**Table 6.3.8.2.2-1: Time function configuration operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| vnfInstanceId | M | 1 | Identifier | Identifier of the VNF instance to be configured. |
| vnfcInstanceId | M | 0..1 | Identifier | Identifier of the VNFC instance to be configured, in case the configuration concerns to a specific VNFC. |
| timeConfig | M | 0..1 | Not specified | The timeConfig provides values for the time protocol configuration e.g. specifies clock mode, clock priority, etc. Examples of time protocols are IEEE 1588-2019 [i.8], etc. |

#### 6.3.8.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.8.2.3-1.

**Table 6.3.8.2.3-1: Time function configuration operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| operationalResult | M | 1 | Not specified | Information about the Time function configuring. |

#### 6.3.8.2.4 Output results

After successful operation, time function configuration notifies to consumer about completion of time function configuration. The result of the operation indicates if it has been successful or not with a standard success/error result.

## 6.3.9 VNF Configuration Management Interface

### 6.3.9.1 Description

The VNF configuration management interface shall support the following operations:

- Set VNF configuration
- Query VNF configuration information
- Backup VNF configuration information

### 6.3.9.2 Set VNF configuration operation

#### 6.3.9.2.1 Description

This operation enables the consumer to request to configure VNF/VNFC instances. The VNF configuration operation can configure both the virtualisation-dependent items of VNF/VNFC instances, as managed by NFV-MANO, as well as the virtualisation-independent items (also referred as VNF's application configuration) of the VNF/VNFC instances.

Table 6.3.9.2.1-1 lists the information flow exchanged between the consumer, e.g. another VNF generic OAM function, NFV-MANO, or other management entities and the producer, that is the VNF Configuration Manager.

**Table 6.3.9.2.1-1: VNF Configuration operation**

| Message | Requirement | Direction |
|---|---|---|
| SetConfigRequest | Mandatory | Consumer → VNF Configuration Manager |
| SetConfigResponse | Mandatory | VNF Configuration Manager → Consumer |

#### 6.3.9.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.9.2.2-1.

**Table 6.3.9.2.2-1: VNF Configuration operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| vnfInstanceId | M | 1 | Identifier | Identifier of the VNF instance to be configured. |
| vnfcInstanceId | M | 0..1 | Identifier | Identifier of the VNFC instance to be configured, in case the configuration concerns to a specific VNFC. |
| configuration | M | 1 | Not specified | Information about the configuration that needs to be applied in the VNF/VNFC instances. |
| configActions | M | 0..1 | Not specified | Specific actions indicated by the consumer upon setting up the configuration such as enabling the capability on rolling back on error. |

#### 6.3.9.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.9.2.3-1.

**Table 6.3.9.2.3-1: VNF Configuration operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| operationalResult | M | 1 | Not specified | Information about the success and failure of configuration. |

#### 6.3.9.2.4 Output results

After successful operation, VNF Configuration Manager function notifies the consumer about completion of the VNF/VNFC configuration. The result of the operation indicates if it has been successful or not with a standard success/error result. In the case of error during the setting of the configuration, automatic rollback actions can be performed by the VNF Configuration Manager.

### 6.3.9.3 Query VNF configuration information operation

#### 6.3.9.3.1 Description

This operation enables the consumer to be able to query VNF configuration of VNF/VNFC instances. See description in clause 6.3.9.2.1 for the appliable items of VNF configuration.

Table 6.3.9.3.1-1 lists the information flow exchanged between the consumer, e.g. another VNF generic OAM function, NFV-MANO, or other management entities and the producer, that is the VNF Configuration Manager.

**Table 6.3.9.3.1-1: Query VNF Configuration operation**

| Message | Requirement | Direction |
|---|---|---|
| QueryConfigRequestI | Mandatory | Consumer → VNF Configuration Manager |
| QueryConfigResponse | Mandatory | VNF Configuration Manager → Consumer |

#### 6.3.9.3.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.9.3.2-1.

**Table 6.3.9.3.2-1: Query VNF configuration operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| filter | M | 1 | Not specified | Filtering criteria to select one or a set of configuration information (see note).<br>The filter to specify on which VNF/VNFC instances the configuration information is requested to be collected (see note). |
| NOTE: Specification of filtering mechanism is part of the protocol design. | | | | |

#### 6.3.9.3.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.9.3.3-1.

**Table 6.3.9.3.3-1: Querying VNF configuration operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| queryResult | M | 0..N | Not specified | Information about the VNF configuration matching the query. |

#### 6.3.9.3.4 Output results

After successful operation, the consumer has received from the VNF Configuration Manager querying results. The result of the operation indicates if it has been successful or not with a standard success/error result. For a particular query, information about the VNF configuration querying result that are matching with the filter shall be returned.

### 6.3.9.4 Backup VNF configuration information operation

#### 6.3.9.4.1 Description

This operation enables the consumer to request the backup of configuration information of VNF/VNFC instances. The Backup VNF configuration information operation can be used to backup the configuration information of both virtualisation-dependent items of VNF/VNFC instances, as well as virtualisation-independent items (also referred as VNF's application configuration) of the VNF/VNFC instances.

Table 6.3.9.4.1-1 lists the information flow exchanged between the consumer, e.g. another VNF generic OAM function, other PaaS Services, NFV-MANO, or other management entities and the producer, that is the VNF Configuration Manager.

**Table 6.3.9.4.1-1: Backup VNF configuration operation**

| Message | Requirement | Direction |
|---|---|---|
| BackupConfigRequest | Mandatory | Consumer → VNF Configuration Manager |
| BackupConfigResponse | Mandatory | VNF Configuration Manager → Consumer |

#### 6.3.9.4.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.9.4.2-1.

**Table 6.3.9.4.2-1: Backup VNF configuration operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| vnfInstanceId | M | 1 | Identifier | Identifier of the VNF instance for which the configuration data is to be backed up. |
| vnfcInstanceId | M | 0..1 | Identifier | Identifier of the VNFC instance for which the configuration data is to be backed up, in case the configuration concerns to a specific VNFC. |
| filter | M | 1 | Not specified | Filtering criteria to select one or a set of configuration data. |
| backupLocation | M | 0..1 | Not specified | Indication on where to backup the configuration data. For example, the Configuration Server PaaS Service can be used to provide the repository for configuration management purposes. In case of cardinality 0, a default location can be considered. |

#### 6.3.9.4.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.9.4.3-1.

**Table 6.3.9.4.3-1: Backup VNF Configuration operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| operationalResult | M | 1 | Not specified | Information about the success or failure of the operation. |

#### 6.3.9.4.4 Output results

After successful operation, VNF Configuration Manager function notifies the consumer about completion of the Backup VNF configuration information operation. The result of the operation indicates if it has been successful or not with a standard success/error result.

## 6.3.10 VNF Testing Management Interface

### 6.3.10.1 Description

This interface enables a consumer to manage a VNF test concerning one or more VNFC instances and applications hosted by the VNF. The VNF test also considers the configuration and involvement of other components in the test, such as NFVI nodes, CIS cluster nodes, CIS cluster storage, NFVI network elements, NFVI resources, NFVI components, etc. residing within the same or different NFVI-PoPs where the VNF instance is placed.

ETSI GS NFV-TST 013 [5] specifies a test case description information model and the relevant data model. The requirements for the test case description and test case inputs as described in ETSI GS NFV-TST 013 [5] are also applicable for the case of the VNF testing management interface.

ETSI GS NFV-IFA 050 [i.21] specifies an Intent management interface. The Intent description can be also related to testing.

### 6.3.10.2 Test configuration operation

#### 6.3.10.2.1 Description

This operation enables the consumer to request to configure a test.

Table 6.3.10.2.1-1 lists the information flow exchanged between the consumer, e.g. another VNF generic OAM function, NFV-MANO, or other management entities and the producer, that is the VNF Testing Manager.

**Table 6.3.10.2.1-1: Test configuration operation**

| Message | Requirement | Direction |
|---|---|---|
| TestConfRequest | Mandatory | Consumer → VNF Testing Manager |
| TestConfResponse | Mandatory | VNF Testing Manager → Consumer |

#### 6.3.10.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.10.2.2-1.

**Table 6.3.10.2.2-1: Test configuration operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| testDescriptor | M | 1 | TestCaseDescriptor | As specified in ETSI GS NFV-TST 013 [5], clause 6.2.2. |
| allowNegotiation | M | 1 | Not specified | It can be used to enable that the VNF Testing Manager processing the test is allowed to negotiate testing parameters with other entities. |
| allowDelegation | M | 1 | Note specified | It can be used to enable that the VNF Testing Manager is allowed to delegate tasks related to the execution of the test to other entities (e.g. other VNF Testing Managers) controlling a respective sub-set of communication network components and/or functionalities. |
| NOTE: Network topology information is part of the TestCaseDescriptor. | | | | |

#### 6.3.10.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.10.2.3-1.

**Table 6.3.10.2.3-1: Test configuration operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| testConfigId | M | 1 | Identifier | Identifier of the test configuration that has been set. |

### 6.3.10.2.4 Operation results

After successful operation, the test configuration has been set and it can be used for subsequent execution. The result of the operation indicates if it has been successful or not with a standard success/error result.

The VNF Testing management interface can be called by the Intent Management system described in ETSI GS NFV-IFA 050 [i.21], which will be used to translate the high-level intent to the TestCaseDescriptor. If still the TestCaseDescriptor is not complete (e.g. not all the configuration is provided), the VNF Testing manager translates a received (sub-)test specification to actions (e.g. tools/ to use/ configuration) or other sub-tests for end-to-end testing.

## 6.3.10.3 Control test execution operation

### 6.3.10.3.1 Description

This operation enables the consumer to setup the time and/or criteria of when the test is to be executed. Consumer can also indicate the time to stop the test or alternatively depending on the test type, testing can run until completion or it can be stopped by requesting to change the status of the test.

Table 6.3.10.3.1-1 lists the information flow exchanged between the consumer, e.g. another VNF generic OAM function, OSS/BSS, NFV-MANO, or other management entities and the producer, that is the VNF Testing Manager.

**Table 6.3.10.3.1-1: Control test execution operation**

| Message | Requirement | Direction |
|---|---|---|
| ControlTestExecutionRequest | Mandatory | Consumer → VNF Testing Manager |
| ControlTestExecutionResponse | Mandatory | VNF Testing Manager → Consumer |

### 6.3.10.3.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.10.3.2-1.

**Table 6.3.10.3.2-1: Control test execution operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| testConfigId | M | 1 | Identifier | Identifier of the testing configuration to execute. |
| startTime | M | 0..1 | DateTime | Timestamp of the time to start the test. See note 1. |
| endTime | M | 0..1 | DateTime | Timestamp the time to finish the test. See note 2. |
| executionCriteria | M | 0..1 | Not specified | Other criteria to trigger the execution of the testing that is not time-based. The parameter shall support setting up criteria based on events such as VNF LCM. See note 1. |
| stopped | M | 0..1 | Boolean | Indication to stop the test. |
| NOTE 1: If neither "startTime", nor "executionCriteria" is provided, the operation request is to execute the test immediately. | | | | |
| NOTE 2: When endTime equals zero, there is no specific period for which the test needs to run. The test runs until completion. | | | | |

### 6.3.10.3.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.10.3.3-1.

**Table 6.3.10.3.3-1: Control test execution operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| testId | M | 1 | Identifier | Identifier of the test. |

### 6.3.10.3.4 Operation results

After successful operation, the test has been created and setup for execution, and depending on the input criteria, it has started. The result of the operation indicates if it has been successful or not with a standard success/error result.

## 6.3.10.4 Query test report operation

### 6.3.10.4.1 Description

This operation enables an authorized consumer (e.g. OSS/BSS) to query and fetch existing test reports from the VNF Testing Manager.

Table 6.3.10.4.1-1 lists the information flow exchanged between the consumer, e.g. another VNF generic OAM function, OSS/BSS, NFV-MANO, or other management entities and the producer, that is the VNF Testing Manager.

**Table 6.3.10.4.1-1: Query test report operation**

| Message | Requirement | Direction |
|---|---|---|
| QueryTestReportRequest | Mandatory | Consumer → VNF Testing Manager |
| QueryTestReportResponse | Mandatory | VNF Testing Manager → Consumer |

### 6.3.10.4.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.10.4.2-1.

**Table 6.3.10.4.2-1: Query test report operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| reportFilter | M | 0..1 | Not Specified | Filtering criteria to select one or a set of reports. If absent, the information related to all available testing reports are returned. |

### 6.3.10.4.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.10.4.3-1.

**Table 6.3.10.4.3-1: Query test report operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| queryResult | M | 0..N | Not Specified | The testing reports matching the query. |

### 6.3.10.4.4 Operation results

If the operation succeeds, the testing reports that matches the filter (if present) are returned. The result of the operation indicates if it has been successful or not with a standard success/error result.

## 6.3.10.5 Query test status operation

### 6.3.10.5.1 Description

This operation enables an authorized consumer (e.g. OSS/BSS) to query the status of the test from the VNF Testing Manager.

Table 6.3.10.5.1-1 lists the information flow exchanged between the consumer, e.g. another VNF generic OAM function, OSS/BSS, NFV-MANO, or other management entities and the producer, that is the VNF Testing Manager.

**Table 6.3.10.5.1-1: Query test report operation**

| Message | Requirement | Direction |
|---|---|---|
| QueryTestStatus | Mandatory | Consumer → VNF Testing Manager |
| QueryTestStatus | Mandatory | VNF Testing Manager → Consumer |

#### 6.3.10.5.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.10.5.2-1.

**Table 6.3.10.5.2-1: Query test status operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| testId | M | 1 | Identifier | Identifier of the test for which the status is requested. |

#### 6.3.10.5.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.10.5.3-1.

**Table 6.3.10.5.3-1: Query test report operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| status | M | 1 | Enumeration | The testing status information. |

#### 6.3.10.5.4 Operation results

If the operation succeeds, the testing status information is returned. The result of the operation indicates if it has been successful or not with a standard success/error result.

## 6.3.11 Configuration Data Management Interface

### 6.3.11.1 Description

The Configuration Data Management interface enables a consumer to store, update, delete and fetch sets of configuration data from a logically centralized data repository.

### 6.3.11.2 Transfer Configuration Data

#### 6.3.11.2.1 Description

This operation enables a consumer to transfer configuration data to the Configuration Server to be stored. The configuration data is transferred and stored as a "file", hereafter referred as "data configuration set".

Table 6.3.11.2.1-1 lists the information flow exchanged between the Configuration Server and a consumer of the interface.

**Table 6.3.11.2.1-1: Transfer Configuration Data operation**

| Message | Requirement | Direction |
|---|---|---|
| TransferConfigurationDataRequest | Mandatory | Consumer → Configuration Server |
| TransferConfigurationDataResponse | Mandatory | Configuration Server → Consumer |

#### 6.3.11.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.11.2.2-1.

**Table 6.3.11.2.2-1: Transfer Configuration Data operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| configDataSet | M | 1 | Not specified | The set of configuration data to be stored. This can be a text file formatted according to some schema and data format, or a binary file. |
| configDataSetDescription | M | 0..1 | String | A description of the configuration data set. |

#### 6.3.11.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.11.2.3-1.

**Table 6.3.11.2.3-1: Transfer Configuration Data operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| configDataSetId | M | 1 | Identifier | Identifier of the configuration data set that has been stored. |

#### 6.3.11.2.4 Output results

After successful operation, the Configuration Server has stored the configuration data set, created an information element associated to the configuration data set with metadata, and identified the configuration data set. The result of the operation indicates if it has been successful or not with a standard success/error result.

### 6.3.11.3 Delete Configuration Data

#### 6.3.11.3.1 Description

This operation enables a consumer to delete a stored configuration data set.

Table 6.3.11.3.1-1 lists the information flow exchanged between the Configuration Server and a consumer of the interface.

**Table 6.3.11.3.1-1: Delete Configuration Data operation**

| Message | Requirement | Direction |
|---|---|---|
| DeleteConfigurationDataRequest | Mandatory | Consumer → Configuration Server |
| DeleteConfigurationDataResponse | Mandatory | Configuration Server → Consumer |

#### 6.3.11.3.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.11.3.2-1.

**Table 6.3.11.3.2-1: Delete Configuration Data operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| configDataSetId | M | 1..N | Identifier | The identifier of the configuration data set to delete. |
| NOTE: It is up to the protocol design stage to determine whether this operation will be modelled as a "bulk" operation that supports deleting multiple configuration data sets in one request, or as a series of requests that delete one configuration data set at a time. | | | | |

#### 6.3.11.3.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.11.3.3-1.

**Table 6.3.11.3.3-1: Delete Configuration Data operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| configDataSetId | M | 1..N | Identifier | Identifier of the configuration data set that has been deleted. |
| NOTE: It is up to the protocol design stage to determine whether this operation will be modelled as a "bulk" operation that supports deleting multiple configuration data sets in one request, or as a series of requests that delete one configuration data set at a time. | | | | |

#### 6.3.11.3.4 Output results

After successful operation, the Configuration Server has deleted the configuration data set(s) and deleted the associated configuration data set metadata. The result of the operation indicates if it has been successful or not with a standard success/error result.

### 6.3.11.4 Update Configuration Data

#### 6.3.11.4.1 Description

This operation enables a consumer to update a stored configuration data set.

Table 6.3.11.4.1-1 lists the information flow exchanged between the Configuration Server and a consumer of the interface.

**Table 6.3.11.4.1-1: Update Configuration Data operation**

| Message | Requirement | Direction |
|---|---|---|
| UpdateConfigurationDataRequest | Mandatory | Consumer → Configuration Server |
| UpdateConfigurationDataResponse | Mandatory | Configuration Server → Consumer |

#### 6.3.11.4.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.11.4.2-1.

**Table 6.3.11.4.2-1: Update Configuration Data operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| configDataSetId | M | 1 | Identifier | The identifier of the configuration data set to be updated. |
| updatedConfigDataSet | M | 1 | Not specified | The set of configuration data with the updated configuration data. This can be a text file formatted according to some schema and data format, or a binary file. |

#### 6.3.11.4.3 Output parameters

None.

#### 6.3.11.4.4 Output results

After successful operation, the Configuration Server has updated the identified configuration data set with the updated configuration data values, and updated the metadata associated to the configuration data set. The result of the operation indicates if it has been successful or not with a standard success/error result.

### 6.3.11.5 Get Configuration Data

#### 6.3.11.5.1 Description

This operation enables a consumer to get (fetch) a stored configuration data set.

Table 6.3.11.5.1-1 lists the information flow exchanged between the Configuration Server and a consumer of the interface.

**Table 6.3.11.5.1-1: Get Configuration Data operation**

| Message | Requirement | Direction |
|---|---|---|
| GetConfigurationDataRequest | Mandatory | Consumer → Configuration Server |
| GetConfigurationDataResponse | Mandatory | Configuration Server → Consumer |

#### 6.3.11.5.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.11.5.2-1.

**Table 6.3.11.5.2-1: Get Configuration Data operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| configDataSetId | M | 1 | Identifier | The identifier of the configuration data set to fetch. |

#### 6.3.11.5.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.11.5.3-1.

**Table 6.3.11.5.3-1: Get Configuration Data operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| configDataSet | M | 1 | Not specified | The configuration data set. This can be a text file formatted according to some schema and data format, or a binary file. |

#### 6.3.11.5.4 Output results

After successful operation, the Configuration Server has transferred to the consumer the configuration data set. The result of the operation indicates if it has been successful or not with a standard success/error result.

### 6.3.11.6 Query Configuration Data Information

#### 6.3.11.6.1 Description

This operation enables a consumer to query information about a stored configuration data set.

Table 6.3.11.6.1-1 lists the information flow exchanged between the Configuration Server and a consumer of the interface.

**Table 6.3.11.6.1-1: Query Configuration Data Information operation**

| Message | Requirement | Direction |
|---|---|---|
| QueryConfigurationDataInfoRequest | Mandatory | Consumer → Configuration Server |
| QueryConfigurationDataInfoResponse | Mandatory | Configuration Server → Consumer |

#### 6.3.11.6.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.11.6.2-1.

**Table 6.3.11.6.2-1: Query Configuration Data Information operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| Filter | M | 1 | Filter | Filter defining the configuration data set information on which the query applies. The filter can be a single identifier, multiple identifiers of configuration data sets, or a wildcard. |

#### 6.3.11.6.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.11.6.3-1.

**Table 6.3.11.6.3-1: Query Configuration Data Information operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| configDataSetInfo | M | 0..N | ConfigDataSetInfo | Information of the configuration data sets matching the input filter. |

#### 6.3.11.6.4 Output results

After successful operation, the Configuration Server has provided to the consumer information about the configuration data sets according to the input filter. The result of the operation indicates if it has been successful or not with a standard success/error result.

### 6.3.11.7 Convert Configuration Data

#### 6.3.11.7.1 Description

This operation enables a consumer to request the Configuration Server to convert a configuration data set from into a different data format. The original configuration data set remains as is, and a new configuration data set is created with the data format requested.

Table 6.3.11.7.1-1 lists the information flow exchanged between the Configuration Server and a consumer of the interface.

**Table 6.3.11.7.1-1: Convert Configuration Data operation**

| Message | Requirement | Direction |
|---|---|---|
| ConvertConfigurationDataRequest | Mandatory | Consumer → Configuration Server |
| ConvertConfigurationDataResponse | Mandatory | Configuration Server → Consumer |

#### 6.3.11.7.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.11.7.2-1.

**Table 6.3.11.7.2-1: Convert Configuration Data operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| configDataSetId | M | 1 | Identifier | The identifier of the configuration data set to convert. |
| dataFormat | M | 1 | String | The destination data format for the conversion. |
| dataSchema | M | 0..1 | Not specified | A data schema to be used during the conversion, in case some specific rules need to be considered for the conversion. |

#### 6.3.11.7.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.11.7.3-1.

**Table 6.3.11.7.3-1: Convert Configuration Data operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| newConfigDataSetId | M | 1 | Identifier | Identifier of the configuration data set that has been created from the conversion. |

#### 6.3.11.7.4 Output results

After successful operation, the Configuration Server has created a new configuration data set by converting the source configuration data set. The result of the operation indicates if it has been successful or not with a standard success/error result.

### 6.3.11.8 Validate Configuration Data

#### 6.3.11.8.1 Description

This operation enables a consumer to request the Configuration Server to validate a configuration data set according to a reference data schema.

Table 6.3.11.8.1-1 lists the information flow exchanged between the Configuration Server and a consumer of the interface.

**Table 6.3.11.8.1-1: Validate Configuration Data operation**

| Message | Requirement | Direction |
|---|---|---|
| ValidateConfigurationDataRequest | Mandatory | Consumer → Configuration Server |
| ValidateConfigurationDataResponse | Mandatory | Configuration Server → Consumer |

#### 6.3.11.8.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.11.8.2-1.

**Table 6.3.11.8.2-1: Validate Configuration Data operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| configDataSetId | M | 1 | Identifier | The identifier of the configuration data set to validate. |
| dataSchema | M | 1 | Not specified | A data schema to be used during the validation. The data schema may be passed by reference or value. |

#### 6.3.11.8.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.11.8.3-1.

**Table 6.3.11.8.3-1: Validate Configuration Data operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| validationResult | M | 1 | Not specified | Result of the validation with information of which parts of the configuration data set are not conformant to the reference data schema, if errors are found. |

#### 6.3.11.8.4 Output results

After successful operation, the Configuration Server has validated the configuration data set based on the data schema to use for the validation. The result of the operation indicates if it has been successful or not with a standard success/error result.

## 6.3.12 Notifications Management Interface

### 6.3.12.1 Description

The Notifications Management Interface shall support management of subscriptions and a notify operation.

### 6.3.12.2 Subscribe operation

#### 6.3.12.2.1 Description

This operation enables the consumer to subscribe with a filter for the notifications related to events performed by corresponding VNF generic OAM functions.

NOTE: Specification of the filtering mechanism is part of the protocol design.

Table 6.3.12.2.1-1 lists the information flow exchanged between the Notification Manager and a consumer of the interface, e.g. OSS/BSS.

**Table 6.3.12.2.1-1: Notification Manager subscribe operation**

| Message | Requirement | Direction |
|---|---|---|
| SubscribeRequest | Mandatory | Consumer → Notification Manager |
| SubscribeResponse | Mandatory | Notification Manager → Consumer |

#### 6.3.12.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.12.2.2-1.

**Table 6.3.12.2.2-1: Subscribe operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| filter | M | 1 | Filter | Input filter for selecting the corresponding VNF generic OAM functions and the related change notifications to subscribe. This filter can contain information about specific types of changes to subscribe to, or attributes of the generic OAM functions, etc. |

#### 6.3.12.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.12.2.3-1.

**Table 6.3.12.2.3-1: Subscribe operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| subscriptionId | M | 1 | Identifier | Identifier of the subscription realized. |

#### 6.3.12.2.4 Output results

After successful subscription, the consumer is registered to receive notifications about events performed by corresponding VNF generic OAM functions. The result of the operation shall indicate if the subscription has been successful or not with a standard success/error result. For a particular subscription, only notifications matching the filter will be delivered to the consumer.

### 6.3.12.3 Notify operation

#### 6.3.12.3.1 Description

This operation distributes notifications to subscribers related to events performed by corresponding VNF generic OAM functions and other PaaS Services. It is a one-way operation issued by the Notification Manager that cannot be invoked as an operation by the consumer (e.g. OSS/BSS).

To receive notifications, the consumer shall have a subscription.

Table 6.3.12.3.1-1 lists the information flow exchanged between the consumer and the Notification Manager.

**Table 6.3.12.3.1-1: Notify operation**

| Message | Requirement | Direction |
|---|---|---|
| Notify | Mandatory | Notification Manager → consumer |

The following notifications can be sent by this operation:

- PaasServiceNotification (see clause 7.10.2) (see note).

NOTE: In case the notification is sent by VNF generic OAM function or other PaaS Service other than the Notification Manager, the notification includes only the payload of it, without the PaaSServiceNotification "wrapping".

## 6.3.13 Policy Management Interface

### 6.3.13.1 Description

This interface enables a consumer of the Policy Agent (e.g. the OSS/BSS) to invoke management operations of policies about VNF, VNF generic OAM functions and other PaaS Services, hereafter all such kinds of policies referred generically as "PaaS policy". The following policy management operations are defined for this interface:

- Create Policy.
- Delete Policy.
- Query Policy.

NOTE: In the context of the PaaS framework in NFV, for the management of subscriptions to notifications sent by the Policy Agent which inform about changes of a policy and about any detected policy conflicts, the interface exposed by the Notification Manager can be also considered.

### 6.3.13.2 Create Policy operation

#### 6.3.13.2.1 Description

This operation enables a consumer of the Policy Agent (e.g. the OSS/BSS) to create a PaaS policy through the Policy Agent. Table 6.3.13.2.1-1 lists the information flow exchanged between the consumer and the Policy Agent.

**Table 6.3.13.2.1-1: Create Policy operation**

| Message | Requirement | Direction |
|---|---|---|
| CreatePolicyRequest | Mandatory | Consumer → Policy Agent |
| CreatePolicyResponse | Mandatory | Policy Agent → Consumer |

#### 6.3.13.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.13.2.2-1.

**Table 6.3.13.2.2-1: Create Policy operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| designer | M | 1 | String | Human readable name of designer of the policy. |
| name | M | 1 | String | Human readable name of the policy. |
| version | M | 1 | Version | Version of the policy. For example, in case the policy information model specified in ETSI GS NFV-IFA 048 [3] is used, this can refer to the "policyVersion" attribute. |
| enforcementFunctionId | M | 0..1 | Not specified | Identifier of the function which enforces the policy. See note 1. |
| policy | M | 1 | Not specified | Specifies the policy to be created. See notes 2 and 3. In case the information model specified in ETSI GS NFV-IFA 048 [3] is used the policy can refer to the "Policy" information element specified in clause 5.2 of ETSI GS NFV-IFA 048 [3]. |
| NOTE 1: The Policy Agent uses the enforcementFunctionId attribute to identify the function enforcing the policy.<br>NOTE 2: An identifier for uniquely identifying the policy is included in the policy.<br>NOTE 3: The consumer (e.g. OSS/BSS) may use this operation to update an existing policy with a new version. For example, in case the policy information model specified in ETSI GS NFV-IFA 048 [3] is used, different policy versions share the same internal identifier of the policy but having different PolicyInfo instances. The design of different policy versions and their business logic is out of the scope of the present document. | | | | |

#### 6.3.13.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.13.2.3-1.

**Table 6.3.13.2.3-1: Create Policy operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| policyInfoId | M | 1 | Identifier | Identifier of the created policy information. |

#### 6.3.13.2.4 Operation results

In case of success, the PaaS policy is transferred to the Policy Agent and corresponding policy information is created by the Policy Agent. In case of failure, appropriate error information is returned.

### 6.3.13.3 Delete Policy operation

#### 6.3.13.3.1 Description

This operation enables the consumer (e.g. OSS/BSS) to delete one or multiple PaaS policy(ies) from the Policy Agent. Table 6.3.13.3.1-1 lists the information flow exchanged between the consumer and the Policy Agent.

**Table 6.3.13.3.1-1: Delete Policy operation**

| Message | Requirement | Direction |
|---|---|---|
| DeletePolicyRequest | Mandatory | Consumer → Policy Agent |
| DeletePolicyResponse | Mandatory | Policy Agent → Consumer |

#### 6.3.13.3.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.13.3.2-1.

**Table 6.3.13.3.2-1: Delete Policy operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| policyInfoId | M | 1..N | Identifier (Reference to PolicyInfo) | Identifier(s) of policy information. |
| NOTE: It is part of the protocol design whether this operation is modelled as a "bulk" operation that allows to delete multiple policies in one request, or as a series of requests that delete one policy at a time. | | | | |

#### 6.3.13.3.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.13.3.3-1.

**Table 6.3.13.3.3-1: Delete Policy operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| deletedPolicyInfoId | M | 0..N | Identifier (Reference to PolicyInfo) | Identifier(s) of the deleted policy information. |

#### 6.3.13.3.4 Operation results

In case of success, the PaaS policy(ies) is (are) deleted from the Policy Agent, and a success indicator is returned to the consumer (e.g. OSS/BSS). In case of failure, appropriate error information is returned.

### 6.3.13.4 Query Policy operation

#### 6.3.13.4.1 Description

This operation enables the consumer (e.g. OSS/BSS) to query the information from the Policy Agent on one or multiple PaaS policy(ies). Table 6.3.13.4.1-1 lists the information flow exchanged between the consumer and the Policy Agent.

**Table 6.3.13.4.1-1: Query Policy operation**

| Message | Requirement | Direction |
|---|---|---|
| QueryPolicyRequest | Mandatory | Consumer → Policy Agent |
| QueryPolicyResponse | Mandatory | Policy Agent → Consumer |

#### 6.3.13.4.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.13.4.2-1.

**Table 6.3.13.4.2-1: Query Policy operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| filter | M | 1 | Filter | Filter defining the policy information on which the query applies, based on attributes of policy information.<br>It can also be used to specify one or more policy(ies) information to be queried by providing their identifiers. |
| attributeSelector | M | 0..N | String | Provides a list of attribute names of policy information. If present, only these attributes are returned for the policy information matching the filter.<br>If absent, the complete policy information is returned. |

#### 6.3.13.4.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.13.4.3-1.

**Table 6.3.13.4.3-1: Query Policy operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| queryPolicyInfoResult | M | 0..N | PolicyInfo | Policy information matching the input filter. If attributeSelector is present, only the attributes listed in attributeSelector are returned for the selected policy information. See note. |
| NOTE: The lower cardinality is 0 since there may be no matches to the provided filter. | | | | |

#### 6.3.13.4.4 Operation results

After success operation, the Policy Agent has queried the internal PaaS policy information. The result of the operation indicates whether it has been successful or not with a standard success/error result. For a particular query, policy information that is matching the filter shall be returned.

# 7 Information elements exchanged

## 7.1 Introduction

This clause defines, or references, definitions of information elements used in the interfaces defined in the present document.

The specification of the following information elements is part of the protocol design:

- String.
- Integer.
- Identifier.
- Filter.
- DateTime.
- Value.
- Version.
- KeyValuePair.

# 7.2 Information elements related to Network Configuration Management interface

## 7.2.1 Introduction

This clause defines information elements related to the Network Configuration Management Interface.

## 7.2.2 CpConfigInfo information element

### 7.2.2.1 Description

This data type provides the list of attributes for the CpConfigInfo information element.

### 7.2.2.2 Attributes

The CpConfigInfo information element shall follow the indications provided in table 7.2.2.2-1.

**Table 7.2.2.2-1: Attributes of the CpConfigInfo information element**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| cpInstanceId | M | 1 | Identifier | Identifier of the external CP instance(s) of the VNF/VNFC instance to be configured. |
| cpConfig | M | 1 | Not specified | Configuration information of the external CP instance(s) to be configured, including the information of the CP address, and the protocol that use for CP connection. |

# 7.3 Information elements related to Upgrade VNF Management interface

## 7.3.1 Introduction

This clause defines information elements related to the Upgrade VNF Management interface.

## 7.3.2 UpgFileData information element

### 7.3.2.1 Description

This data type provides the list of attributes for the UpgFileData information element.

### 7.3.2.2 Attributes

The UpgFileData information element shall follow the indications provided in table 7.3.2.2-1.

**Table 7.3.2.2-1: Attributes of the UpgFileData information element**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| fileInfoId | M | 1 | Identifier | Identifier of the Upgrade VNF file define by the file provider. |
| upgFileDescription | M | 1 | String | Human-readable description of the Upgrade VNF file. |
| fileAccessLocation | M | 1 | String | Information about the Upgrade VNF file access address. |
| installationScriptLocation | M | 0..1 | String | Information about the access address of the installation script used for installing the Upgrade VNF file. |
| version | M | 1 | Version | Version of the Upgrade VNF file. |

## 7.3.3 FileInfo information element

### 7.3.3.1 Description

This data type provides the list of attributes for the FileInfo information element.

### 7.3.3.2 Attributes

The FileInfo information element shall follow the indications provided in table 7.3.3.2-1.

**Table 7.3.3.2-1: Attributes of the FileInfo information element**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| fileInfoId | M | 1 | Identifier | Identifier of the Upgrade VNF file defined by the file provider. |
| fileDescription | M | 1 | String | Human-readable description of the Upgrade VNF file. |
| fileAccessLocation | M | 1 | String | Information about the Upgrade VNF file access address. |
| installationScriptLocation | M | 0..1 | String | Information about the access address of the installation script used for installing the Upgrade VNF file. |
| version | M | 1 | Version | Version of the Upgrade VNF file. |

# 7.4 Information elements related to the Log Aggregator Exposure interface

## 7.4.1 Introduction

This clause defines information elements related to the Log Aggregator.

## 7.4.2 LogAggregateOutput information element

### 7.4.2.1 Description

This data type provides the list of attributes for the LogAggregateOutput information element.

### 7.4.2.2 Attributes

The LogAggregateOutput information element shall follow the indications provided in table 7.4.2.2-1.

**Table 7.4.2.2-1: Attributes of the LogAggregateOutput information element**

| Attribute | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| logAggregateId | M | 1 | Identifier | Identifier of the log aggregator output. |
| aggregateOutputGenerationTime | M | 1 | DateTime | The time when the log aggregator output is generated. |
| typeSpecificOutput | M | 1 | Not specified | The output information specific to the type of log aggregate. |

# 7.5 Information elements related to the Log Analysis exposure interface

## 7.5.1 Introduction

This clause defines information elements related to the Metrics Exposure Interface.

## 7.5.2 LogAnalysisOutput information element

### 7.5.2.1 Description

This data type provides the list of attributes for the LogAnalysisOutput information element.

### 7.5.2.2 Attributes

The LogAnalysisOutput information element shall follow the indications provided in table 7.5.2.2-1.

**Table 7.5.2.2-1: Attributes of the LogAnalysisOutput information element**

| Attribute | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| logAnalysisType | M | 1 | Not specified | The type of a log analysis type (e.g. root cause analysis of VNF faults). |
| logAnalysisId | M | 1 | Identifier | Identifier of the Log analysis output. |
| logAnalysisOutputGenerationTime | M | 1 | DateTime | The time when the log analysis output is generated. |
| typeSpecificOutput | M | 1 | Not specified | The output information specific to the log analysis type. |

# 7.6 Information elements related to the VNF Metrics Analysis Exposure interface

## 7.6.1 Introduction

This clause defines information elements related to the VNF Metrics analysis generic OAM function exposure interface.

## 7.6.2 MetricsAnalysisOutput information element

### 7.6.2.1 Description

This data type provides the list of attributes for the MetricsAnalysisOutput information element.

### 7.6.2.2 Attributes

The MetricsAnalysisOutput information element shall follow the indications provided in table 7.6.2.2-1.

**Table 7.6.2.2-1: Attributes of the MetricsAnalysisOutput information element**

| Attribute | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| analyticsType | M | 1 | String | The type of a metrics analytics process (e.g. "Network service alarm incident analysis", "Network service health analysis", "Network service resource utilization analysis", etc.). |
| analyticsId | M | 1 | Identifier | Identifier of the metrics analytics output. |
| analyticsOutputGenerationTime | M | 1 | DateTime | The time when the metrics analytics output is generated. |
| typeSpecificOutput | M | 1 | Not specified | The output information specific to the type of metrics analytics. |
| recommendedActions | M | 0..1 | Not specified | Recommended actions to follow up according to the output of data analytics. In case there are no recommended actions described, then these can be decided by a consumer of the interface. |

# 7.7 Information elements related to PSM produced management interfaces

## 7.7.1 Introduction

This clause defines information elements and notifications related to the management interfaces produced by the PSM.

## 7.7.2 PaasServiceLifecycleNotification

### 7.7.2.1 Description

This notification informs the receiver of lifecycle events of a PaaS Service instance.

### 7.7.2.2 Trigger conditions

This notification is produced when there is a change in the lifecycle of the PaaS Service cause by a PaaS Service lifecycle management operation, including:

- Instantiation of a PaaS Service;
- Termination of a PaaS Service instance;
- Scaling of a PaaS Service instance.

Notifications shall be issued at the start of the lifecycle operation, and when the operation concludes, including the case that it concludes unsuccessfully, i.e. when some error occurs and the operation cannot continue.

### 7.7.2.3 Attributes

The attributes of the notification shall follow the indications provided in table 7.7.2.3-1.

**Table 7.7.2.3-1: Attributes of the PaasServiceLifecycleNotification**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| notificationStatus | M | 1 | Enum | Indicates whether this notification informs about the start or result of a lifecycle operation.<br>VALUES:<br>• START: informs about the start of the lifecycle operation<br>• RESULT: informs about the final or intermediate result of the lifecycle operation |
| paasServiceId | M | 1 | Identifier | Identifier of the affected PaaS Service instance. |
| operation | M | 1 | String | Name of the lifecycle operation. |
| operationStatus | M | 1 | Not specified | Indicates the operation status, including any related error information is the lifecycle operation concludes unsuccessfully. |
| lifecycleOperationOccurrenceId | M | 1 | Identifier | Identifier of the lifecycle operation occurrence associated to the notification. |
| deploymentLifecycleOperationOccurrenceId | M | 0..1 | Identifier | Identifier of the lifecycle operation occurrence associated to the deployment resource changes as exposed by the respective management function responsible for the resource fulfilment of the PaaS Service instance. See note. |
| changedInfo | M | 0..1 | Not specified | Additional information of the PaaS Service affected by the lifecycle operation. It can include information related to PaaS Service (see PaasServiceEntry). |
| NOTE: For instance, in the case of the PaaS Service being deployed as a VNF instance, this can refer to a VNF lifecycle management operation occurrence. | | | | |

# 7.8 Information elements related to PSR produced management interfaces

## 7.8.1 Introduction

This clause defines information elements and notifications related to the management interfaces produced by the PSR.

## 7.8.2 PsdOnboardingNotification

### 7.8.2.1 Description

This notification informs the receiver of events about onboarding of PSDs. The notification is only issued after all the onboarding steps (i.e. uploading and validation of the PSD) are done.

### 7.8.2.2 Trigger conditions

This notification is produced when:

- New PSD is onboarded.

### 7.8.2.3 Attributes

The attributes of the notification shall follow the indications provided in table 7.8.2.3-1.

**Table 7.8.2.3-1: Attributes of the PsdOnboardingNotification**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| psdInfoObjectId | M | 1 | Identifier | Identifier of the PSD information object. |
| psdId | M | 1 | Identifier | Identifier of the onboarded PSD. This identifies the PSD in a globally unique way. This information is copied from the PSD file(s). |
| versionTag | M | 1 | Not specified | Version tag of the PSD. |

## 7.8.3 PsdChangeNotification

### 7.8.3.1 Description

This notification informs the receiver of events about PSD management and changes of state of an onboarded PSD.

### 7.8.3.2 Trigger conditions

This notification is produced when:

- An onboarded PSD is deleted.
- The operational state of a specific version of a PSD is changed.

### 7.8.3.3 Attributes

The attributes of the notification shall follow the indications provided in table 7.8.3.3-1.

**Table 7.8.3.3-1: Attributes of the PsdChangeNotification**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| psdInfoObjectId | M | 1 | Identifier | Identifier of the PSD information object. |
| changeType | M | 1 | Enum | The type of change on the PSD.<br>VALUES:<br>• OP_STATE_CHANGE: change of operational state of an onboarded PSD<br>• DELETE: deletion of a PSD |
| psdId | M | 0..1 | Identifier | Identifier of the onboarded PSD.<br>Shall be present if "changeType = OP_STATE_CHANGE". |
| operationalState | M | 0..1 | Enum | Current operational state of the PSD information object and corresponding onboarded PSD.<br>VALUES:<br>• ENABLED<br>• DISABLED<br>Shall be present if "changeType = OP_STATE_CHANGE". |
| changedData | M | 0..1 | Not specified | Additional information related to the PSD change. |

## 7.8.4 PsdInfoObject information element

### 7.8.4.1 Description

This information element represents a PSD information object.

### 7.8.4.2 Attributes

The PsdInfoObject information element shall follow the indications provided in table 7.8.4.2-1.

**Table 7.8.4.2-1: Attributes of the PsdInfoObject information element**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| psdInfoObjectId | M | 1 | Identifier | Identifier of the PSD information object. |
| psdInfoObjectVersion | M | 1..N | Not specified | Versions of the PSD information object keeping track of the modifications performed to the PSD information object, including uploads of PSDs and update of the PSD information object. |
| psdInfo | M | 0..N | PsdInfo | PSD versions associated to the PSD information object. |

## 7.8.5 PsdInfo information element

### 7.8.5.1 Description

This information element represents the information of a specific version of a PSD onboarded.

### 7.8.5.2 Attributes

The PsdInfo information element shall follow the indications provided in table 7.8.5.2-1.

**Table 7.8.5.2-1: Attributes of the PsdInfo information element**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| psdFiles | M | 1 | Not specified | Files conforming the PSD. |
| psdId | M | 1 | Identifier | Identifier of the PSD. This identifies the PSD in a globally unique way. This information is copied from the PSD file(s). |
| versionTag | M | 1 | Not specified | Version tag of the PSD. |
| paasServiceVersion | M | 0..1 | Version | Version of the PaaS Service. It shall be present if the PaaS Service is versioned. |
| operationalState | M | 1 | Enum | Operational state of the PSD.<br>VALUES:<br>• ENABLED<br>• DISABLED |
| userDefinedData | M | 0..N | KeyValuePair | User defined data. |
| checksum | M | 1 | Not specified | Checksum of the PSD. |

## 7.8.6 PaasServiceEntry information element

### 7.8.6.1 Description

This information element represents an entry in the registry of registered PaaS Service instance with additional registration information, such as association to PaaS Service Consumers and runtime deployment information.

### 7.8.6.2 Attributes

The PaasServiceEntry information element shall follow the indications provided in table 7.8.6.2-1.

**Table 7.8.6.2-1: Attributes of the PaasServiceEntry information element**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| paasServiceId | M | 1 | Identifier | Identifier of the registered PaaS Service instance. |
| paasServiceInfo | M | 1 | PaasServiceInfo | Information of the registered PaaS Service instance. |
| usageFormat | M | 1 | Enum | The usage format of the PaaS Service instance.<br>VALUES:<br>• VNF_COMMON: to use the PaaS Service as VNF Common Service<br>• VNF_DEDICATED: to use the PaaS Service as VNF Dedicated Service<br>• NS: to use the PaaS Service for any NS constituent other than a VNF<br>• UNDEFINED: no specific usage defined |
| location | M | 1 | Not specified | Information about the location(s) where the PaaS Service is instantiated. This can be in the form of some identification information of the NFVI-PoP or CIS cluster, or geographical location information. |
| deploymentHandle | M | 1 | Not specified | Information about the deployment of the PaaS Service and its form. For instance, if a PaaS Service is deployed as a VNF, a reference to a VNF instance, or if a PaaS Service is deployed as an MCCO, a reference to an MCCO instance. |
| paasServiceHandle | M | 1 | Not specified | A handle enabling the access and use of the PaaS Service instance. This can include, for instance, certain interface endpoint URI together with necessary credentials to access it. The type and format of the handle depends on the form that the PaaS Service is formed. |
| scaleInfo | M | 1 | Not specified | Information about the scale/capacity of the PaaS Service instance. |
| consumerId | M | 0..N | Identifier | Identifiers of PaaS Service consumers associated to, i.e. making use, the PaaS Service instance. This can be identifiers of VNF instances and NS instances. |

## 7.8.7 PaasServiceInfo information element

### 7.8.7.1 Description

This information element represents design-time information of a registered PaaS Service instance.

### 7.8.7.2 Attributes

The PaasServiceInfo information element shall follow the indications provided in table 7.8.7.2-1.

**Table 7.8.7.2-1: Attributes of the PaasServiceInfo information element**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| psdId | M | 1 | Identifier | Identifier of the PSD from which the PaaS Service instance is based on. |
| paasServiceType | M | 1 | String | The type of PaaS Service. |
| paasServiceVersion | M | 0..1 | Version | Version of the PaaS Service. It shall be present if the PaaS Service is versioned. |
| additionalInfo | M | 0..1 | Not specified | Additional information which is specific to the PaaS Service and its type. |

# 7.9 Information elements related to Configuration Data Management interface

## 7.9.1 Introduction

This clause defines information elements related to the Configuration Data Management interface.

## 7.9.2 ConfigDataSetInfo information element

### 7.9.2.1 Description

This information element provides information about a data configuration set managed by the PaaS Service Configuration Server.

### 7.9.2.2 Attributes

The ConfigDataSetInfo information element shall follow the indications provided in table 7.9.2.2-1.

**Table 7.9.2.2-1: Attributes of the ConfigDataSetInfo information element**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| configDataSetId | M | 1 | Identifier | The identifier of the configuration data set. |
| configDataSet | M | 1 | Not specified | The stored configuration data set. This can be a text file formatted according to some schema and data format, or a binary file. |
| configDataSetDescription | M | 0..1 | String | A description of the configuration data set.<br><br>The attribute and a value may be absent. |
| dataFormat | M | 1 | String | The data format of the data configuration set. |
| dataSchema | M | 0..1 | Not specified | The data schema to which the data configuration set conforms to. This can be an actual schema file, or a reference to a schema.<br><br>The attribute and a value shall be present if the file has been validated against some data schema, or if the configuration data set has been created during a conversion of configuration data and an input data schema was provided. |
| fileSize | M | 1 | Integer | Size of the configuration data set file. |

# 7.10 Information elements related to Notifications management interface

## 7.10.1 Introduction

This clause defines information elements related to the Notification Manager.

## 7.10.2 PaasServiceNotification

### 7.10.2.1 Description

This notification informs the receiver of an event produced by a PaaS Service, including specific VNF generic OAM functions.

### 7.10.2.2 Trigger conditions

This notification is produced when the Notification Manager has processed (e.g. including potential grouping and deduplication) incoming events and notifications produced by another PaaS Service and determines the need to forward the notification to a subscriber.

### 7.10.2.3 Attributes

The attributes of the notification shall follow the indications provided in table 7.10.2.3-1.

**Table 7.10.2.3-1: Attributes of the PaasServiceNotification**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| sourcePaasServiceId | M | 1 | Identifier | Identifier of the PaaS Service instance that generates the source notification. |
| sourcePaasServiceType | M | 1 | Enum | The type of PaaS Service that originates the source notification.<br>VALUES:<br>• LOG-ANALYSER<br>• VNF-METRICS-ANALYSER<br>• TIME-MANAGER<br>• POLICY-AGENT<br>• CONFIGURATION-SERVER<br>See note. |
| notification | M | 1..N | NotificationPayload | Forwarded notification(s) raised by the identified generating PaaS Service. |
| NOTE: The values capture the list of PaaS Services that provide notifications (as per other specified requirements) according to the present document version. | | | | |

## 7.10.3 NotificationPayload information element

### 7.10.3.1 Description

This information element provides the payload of a notification with additional metadata.

### 7.10.3.2 Attributes

The NotificationPayload information element shall follow the indications provided in table 7.10.3.2-1.

**Table 7.10.3.2-1: Attributes of the NotificationPayload information element**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| receptionTime | M | 1 | DateTime | Timestamp of the reception time the source notification was received and/or collected. |
| payload | M | 1 | Not specified | Payload of the notification that is forwarded, as originally received. |

# 8 VNF generic OAM functions and other PaaS Services interactions with VNFs

## 8.1 Introduction

As specified in clause 4.2.3 interfaces IF-V1 and IF-V2 comprise a set of interfaces used to interact with the VNF and NS instances that make use of the VNF generic OAM functions and other PaaS Services. Clause 8.2 specifies service interaction requirements for the interaction between the VNF generic OAM functions and other PaaS Services with the VNFs, where applicable. Clause 8.3 describes different deployment examples regarding interactions between the VNF generic OAM functions and other PaaS Services.

NOTE: The present document does not specify the modelling regarding IF-V1 and IF-V2 interfaces.

## 8.2 Service interaction requirements on IF-V for VNF generic OAM functions and other PaaS Services

Table 8.2-1 provides service interaction requirements applicable to the IF-V interfaces.

**Table 8.2-1: Service interaction requirements on IF-V**

| Identifier | Service interaction requirement | Additional description and relevant functional requirements |
|---|---|---|
| sb-v.001 | The IF-V interface shall support the capability to support interactions between VNF generic OAM functions or other PaaS Services and VNFs for management purposes. | This requirement also expresses that VNF supports one or multiple means of management offered by the VNF generic OAM functions or other PaaS Services. How the connectivity is established between VNFs and VNF generic OAM functions or other PaaS Services is out of scope of the present document.<br><br>Reference requirements:<br>• VnfGenOam.GenFunc.003 |
| sb-v.002 | For configuration purposes of the VNF, the IF-V shall support the capabilities to convey configuration data via push mechanisms. | See note 1.<br><br>Push mechanism refers to the methods by which the VNF generic OAM function provides the configuration data into the VNF instance.<br><br>Reference requirements:<br>• TrafficEnforcer.001<br>• NetConfMa.001<br>• UpgradeVNF.001<br>• TimeFunc.Mgmt.001<br>• VnfConfigMgmt.Conf.001 |

| Identifier | Service interaction requirement | Additional description and relevant functional requirements |
|---|---|---|
| sb-v.003 | For configuration purposes of the VNF, the IF-V shall support the capabilities to convey configuration data via pull mechanisms. | Pull mechanism refers to the methods by the VNF instance can retrieve the configuration data from the VNF generic OAM function or another PaaS Service like the Configuration Server.<br><br>Reference requirements:<br>• TrafficEnforcer.001<br>• NetConfMa.001<br>• UpgradeVNF.001<br>• TimeFunc.Mgmt.001<br>• VnfConfigMgmt.Conf.001<br>• ConfigServer.005 |
| sb-v.004 | For monitoring purposes of the VNF, the IF-V shall support the capability to transfer PM/FM metrics to a VNF generic OAM function or other PaaS Services via streaming mechanisms. See note 2. | See note 2.<br><br>Reference requirements:<br>• VNFMetricAggregator.001 |
| sb-v.005 | For monitoring purposes of the VNF, the IF-V shall support the capability to transfer PM/FM metrics to a VNF generic OAM function or other PaaS Services via file-based mechanisms. See note 2. | See note 2.<br><br>Reference requirements:<br>• VNFMetricAggregator.001 |
| sb-v.006 | For logging purposes of the VNF, the IF-V shall support the capability to retrieve and pull logging data from a VNF instance. See note 3. | See note 3.<br><br>Reference requirements:<br>• LogAggregator.001 |
| sb-v.007 | For logging purposes of the VNF, the IF-V shall support the capability to transfer and collect events (or messages) with logging data from a VNF instance. See note 3. | See note 3.<br><br>Reference requirements:<br>• LogAggregator.001 |
| sb-v.008 | The IF-V interface shall support the capability to support interactions between PaaS Services and VNFs for configuration data management and control purposes. | See note 4. |
| sb-v.009 | For policy execution purposes, the IF-V shall support the capability of VNF/VNFC instances to interact with the Policy Agent. | See note 5. |

NOTE 1: This requirement is applicable to the case of VNF Configuration Manager, the Network Configuration Manager, the Traffic Enforcer, the Time function and the Upgrade VNF function.
NOTE 2: This is applicable to the case of the VNF Metrics Aggregator.
NOTE 3: This is applicable to the case of the Log Aggregator.
NOTE 4: This is applicable to the case of the Configuration Server.
NOTE 5: VNF generic OAM functions or other PaaS Services may enforce policies at VNF/VNFC-level corresponding to their scope of management, in which case there need not be an explicit interaction between the VNF/VNFC instances and the Policy Agent over the IF-V.

## 8.3 Example interactions between VNFs and VNF generic OAM functions and other PaaS Services

A VNF generic OAM function or other any other PaaS Service can realize a VNF Common/Dedicated Service as described in ETSI GR NFV-IFA 029 [i.3] and can serve one or more VNF instances.

- **Option 1:** VNF generic OAM function or other any other PaaS service as a VNF Dedicated Service.

According to ETSI GR NFV-IFA 029 [i.3], a VNF Dedicated Service is a modular service or a function with a lifecycle dependent on its consumers and that can only be consumed by a specific set of applications or services. From a deployment point of view, the following cases can be considered as depicted in figure 8.3-1.

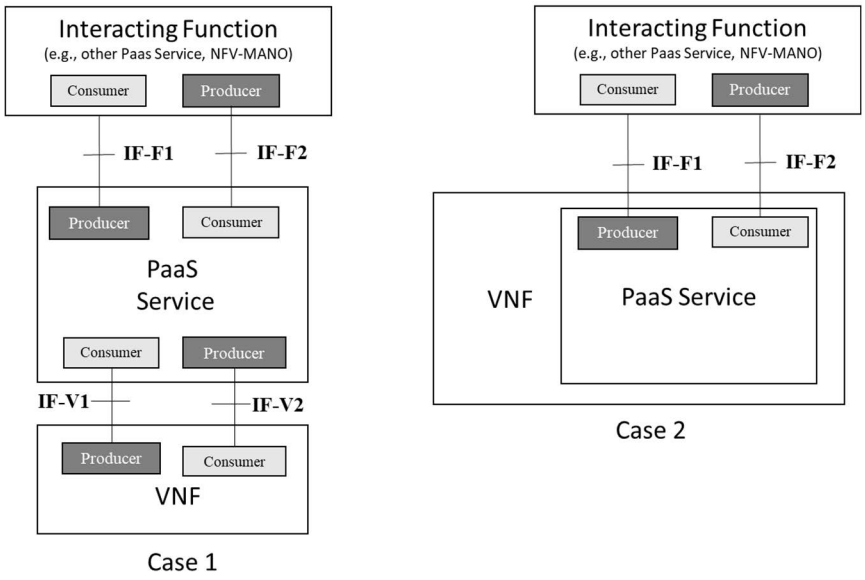


**Figure 8.3-1: Different deployment options for a dedicated PaaS Service**

In Case 1, the VNF is interacting with the PaaS Service (the VNF Generic OAM Function or any other PaaS Service) through interfaces IF-V1 and/or IF-V2 based on, for example, some standard solution. In Case 2, the PaaS service (the VNF Generic OAM Function and any other PaaS Service) is considered part of the VNF and can be also co-located with the rest of the components conforming the VNF instance. In this case, the PaaS Service is typically created during the VNF creation and has a lifecycle associated to the lifecycle of the VNF instance. In that case interactions through IF-V1 or IF-V2 are internal to the VNF, not precluding the use of standard solutions for such interactions.

In both cases, interactions between the PaaS Service and Interacting Functions (e.g. other PaaS Service, NFV MANO) are through IF-F1 and IF-F2. In both cases the lifecycle of the PaaS Service depends on the lifecycle of the VNF, since the PaaS Service is a VNF PaaS Service according to ETSI GR NFV-IFA 029 [i.3].

- **Option 2:** PaaS Service (VNF generic OAM function or any other PaaS service) as a VNF Common Service

According to ETSI GR NFV-IFA 029 [i.3], a VNF Common Service is a modular service or a function with a lifecycle independent from its consumers and that is consumable by either one or multiple services.

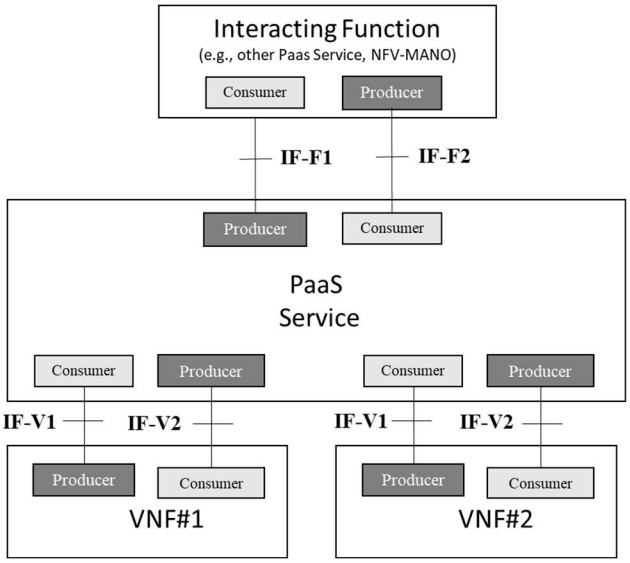


**Figure 8.3-2: PaaS Service as a VNF Common Service**

A PaaS Service (e.g. a VNF Generic OAM function) can serve at the same time one or more VNFs of different types (see figure 8.3-2). The same IF-V1 and IF-V2 interfaces can be used for different VNFs. In this case, the VNF and the PaaS Service (i.e. a VNF Generic OAM Function or any other PaaS Service) are interacting through interfaces IF-V1 and/or IF-V2 based on, for example, some standard solution.

# 9 Descriptors for VNF generic OAM functions and other PaaS Services

## 9.1 Introduction

As studied by ETSI GR NFV-IFA 029 [i.3], PaaS Services can be deployed as VNFs or can be realized as NFVI resources or as a new managed object class.

A PaaS Service Descriptor (PSD) is a template that describes a PaaS Service in terms of its characteristics, deployment and operational behaviour. Due to the different means and possibilities of deployment and the specific characteristics of PaaS Services, the files archive for a PaaS Service description is split into two main parts:

- Deployment and lifecycle management of the PaaS Service (see clause 9.2). These are further referred as "deployment and lifecycle descriptors of a PaaS Service".
- Description of the characteristics and capabilities of the PaaS Service (see clause 9.4). Furthermore, in the case of a VNF generic OAM function, additional characteristics and capabilities can be described (see clause 9.3). These are referred as "descriptor of PaaS Service characteristics" and "descriptor of VNF generic OAM function characteristics", respectively. These are commonly collected under the umbrella of the term PSD.

Figure 9.1-1 illustrates a high-level overview of the PaaS Service description files archive. A PaaS Service description files archive can be composed of multiple files to convey the necessary parts concerning the deployment and lifecycle, and the characteristics of the PaaS Service.

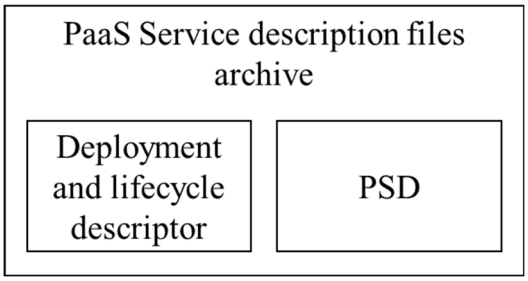


**Figure 9.1-1: PSD composition**

NOTE: Even though figure 9.1-1 does only depict single descriptors, i.e. one for deployment and lifecycle, one for PSD, individual descriptors can be implemented by various artifacts and files.

Management of the PSDs is performed via the PSR through the PaaS Services descriptor management interface (see clause 10.3.2). Regarding the "deployment and lifecycle descriptors of a PaaS Service", these are managed according to the interfaces and/or capabilities of other NFV-MANO functional blocks and functions supporting respective forms of deployment of the PaaS Service (see clause 9.2). For a successful deployment and operation of a PaaS Service, both PSD and the "deployment and lifecycle descriptor" shall be onboarded and enabled.

The PSD of a PaaS Service supports the registration, inventory and discovery processes of PaaS Services by NFV-MANO. By being able to express the capability of and requirement on specific PaaS Services, NFV-MANO can use such information to determine various aspects regarding the PaaS Services such as when:

- Onboarding of PaaS Services, i.e. cataloguing PaaS Services that are ready for deployment. The information in the descriptors can be consumed directly by NFV-MANO.

- Registering information about capabilities and other properties of PaaS Services into the inventory runtime information about instantiated PaaS Services. The information in the descriptors is consumed directly by NFV-MANO when performing the registration.
- Discovering the capabilities and properties of instantiated PaaS Services from the registry. The information in the descriptors is consumed indirectly by NFV-MANO based on what has previously been registered.
- "Binding" PaaS Services to its consumers, e.g. binding the VNF generic OAM functions to its managed VNF instances by establishing the association between which VNF instances make use of which VNF generic OAM functions. The information in the descriptors is consumed indirectly by NFV-MANO based on what has previously been registered.

Irrelevant of the form of deployment adopted, the specification of the descriptors for PaaS Services shall consider:

- The general functional requirements as described in tables 4.2.1.1-1 and 4.2.1.1-2 in clause 4.2.1.1.
- A unique descriptor irrelevant of the PaaS Service type (e.g. in case of VNF generic OAM functions such as Traffic Enforcer, Metrics Aggregator, etc.).
- Reusing the corresponding standardized deployment descriptors and packaging formats.

## 9.2 Deployment and lifecycle descriptors for PaaS Services

### 9.2.1 Overview

As introduced in clause 9.1, PaaS Services can be deployed, and their lifecycle be managed using different forms. Based on the concept of VNF Common/Dedicated Service as described in ETSI GR NFV-IFA 029 [i.3] PaaS Services can be deployed as VNFs or can be realized as NFVI resources or as a new object class.

Clause 9.2 specifies the requirements regarding the deployment and lifecycle descriptors for PaaS Services.

For a successful lifecycle management of the PaaS Services, the corresponding deployment and lifecycle descriptors of the PaaS Service shall be onboarded and available to management and orchestration functions responsible for the lifecycle of deployments (the NFV-MANO in the case of the NFV-MANO architectural framework), at the latest, before the lifecycle management of the PaaS Service is to be executed.

### 9.2.2 PaaS Service deployed as a "VNF"

In the case a PaaS Service is deployed and managed as a VNF (VM-based and/or container-based), the requirements and information modelling specified in ETSI GS NFV-IFA 011 [1] shall apply regarding the deployment and lifecycle descriptors and packaging of the PaaS Service.

When the PaaS Service is deployed as a VNF, the VNF Package (as specified in ETSI GS NFV-IFA 011 [1]) may, but need not, contain the PSD as an additional VNF Package artifact. In case the PSD is contained in the VNF Package, the VNF Package metadata artifacts shall identify in a standard format the PSD.

Requirements for the management of VNF Packages, as specified in ETSI GS NFV-IFA 013 [4], shall apply.

When the PaaS Service is deployed as a VNF, certificate management of the PaaS Service shall be carried out according to the VNF certificate management framework specified in ETSI GS NFV-IFA 026 [6]. In case of Delegation mode of VNFC certificate management, certificate related information shall be provided in the VNFD as specified in ETSI GS NFV-IFA 011 [1].

### 9.2.3 PaaS Service deployed as an "NFVI resource"

NOTE: The specification of descriptors for deployment of PaaS Services as one or multiple virtualised resources offered as a new type of NFVI resource is not completed. Therefore, the present document version does not specify related requirements.

### 9.2.4 PaaS Service deployed as a "managed CIS cluster object"

In the case a PaaS Service is deployed and managed as an MCCO, the requirements and information modelling specified in clause 6.3 of ETSI GS NFV-IFA 036 [2] shall apply regarding the descriptors of PaaS Services.

When the PaaS Service is deployed as an MCCO, the archive file of the MCCO declarative descriptor may, but need not, contain the PSD as an additional artifact or description. In case the PSD is contained in the archive file of the MCCO declarative descriptor, the MCCO declarative descriptor metadata artifacts shall identify in a standard format the PSD.

## 9.3 Characteristics description of a VNF generic OAM function

### 9.3.1 Characteristics descriptor requirements

Table 9.3.1-1 specifies general requirements for the description of characteristics and capabilities of VNF generic OAM functions.

**Table 9.3.1-1: Requirements related to the description of the characteristics and capabilities of VNF generic OAM functions**

| Identifier | Requirement |
|---|---|
| VnfGenOam.Desc.001 | Replaced by requirement Psd.001 in clause 9.4. |
| VnfGenOam.Desc.002 | Replaced by requirement Psd.002 in clause 9.4. |
| VnfGenOam.Desc.003 | The characteristics descriptor of a VNF generic OAM function shall support describing the type of a VNF generic OAM function. |
| VnfGenOam.Desc.004 | The characteristics descriptor of a VNF generic OAM function shall support describing the characteristics of a VNF generic OAM function according to its type. See note 1. |
| VnfGenOam.Desc.005 | Replaced by requirement Psd.005 in clause 9.4. |
| VnfGenOam.Desc.006 | Replaced by requirement Psd.006 in clause 9.4. |
| VnfGenOam.Desc.007 | The characteristics descriptor of a PaaS Service shall support describing the supported IF-F1 interfaces exposed by the PaaS Service and their version. |
| VnfGenOam.Desc.008 | The characteristics descriptor of a VNF generic OAM function shall support describing the supported interactions with VNFs. |
| VnfGenOam.Desc.009 | Replaced by requirement Psd.009 in clause 9.4. |
| NOTE 1: Examples of characteristics of a VNF generic OAM function according to its type, are data models used for configuration for the case of VNF Configuration Manager.<br>NOTE 2: Void.<br>NOTE 3: See clause 4.2.3 regarding the IF-F2 interfaces consumed by the VNF generic OAM function. | |

## 9.4 General requirements for PaaS Services description

Table 9.4-1 specifies general requirements for PaaS Services descriptors.

**Table 9.4-1: Requirements related to PaaS Services Descriptors**

| Identifier | Requirement |
|---|---|
| Psd.001 | The PSD shall support uniquely identifying a PaaS Service. |
| Psd.002 | The PSD shall support specifying the following metadata of a PaaS Service: version, provider, name and product description of the PaaS Service. |
| Psd.003 | The PSD shall support describing the type of a PaaS Service. |
| Psd.004 | The PSD shall support describing the characteristics of a PaaS Service according to its type. See note. |
| Psd.005 | The PSD shall support describing the form of deployment and lifecycle management of a PaaS Service. |
| Psd.006 | The characteristics descriptor of a PaaS Service shall support describing the supported management interfaces IF-M exposed by the PaaS Service and their version. |
| Psd.007 | The characteristics descriptor of a PaaS Service shall support describing the supportedIF-F1 interfaces exposed by the PaaS Service and their version. |
| Psd.008 | The characteristics descriptor of a PaaS Service shall support describing the supported interactions in the IF-V interface with VNF and NS constituents. |
| Psd.009 | The characteristics descriptor of a PaaS Service shall support describing the supported interactions in the IF-F2 with other entities or other PaaS Services. |

| Identifier | Requirement |
|---|---|
| Psd.010 | The characteristics descriptor of a PaaS Service shall support describing the pre-defined (if applicable) usage format of the PaaS Service. |
| NOTE: Examples of characteristics of a PaaS Service according to its type are specific properties of the PaaS Service. | |

# 9.5 PaaS Service Descriptor

## 9.5.1 Overview

Clause 9.5 specifies a high-level information model of PSD fulfilling the requirements specified in clause 9.4.

## 9.5.2 Psd information element

### 9.5.2.1 Description

A PSD is a template that describes a PaaS Service in terms of its characteristics, deployment and operational behaviour.

### 9.5.2.2 Attributes

The attributes of the Psd information element shall follow the indications provided in table 9.5.2.2-1.

**Table 9.5.2.2-1: Attributes of the Psd information element**

| Attribute | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| psdId | M | 1 | Identifier | Identifier of the PSD. This attribute shall be globally unique. |
| version | M | 0..1 | Version | Version of the PaaS Service. It shall be present if the PaaS Service is versioned. |
| provider | M | 1 | String | Provider of the PaaS Service. |
| name | M | 1 | String | Name of the PaaS Service. |
| description | M | 0..1 | String | Human readable name of the PaaS Service. |
| type | M | 1 | Enum | Type of PaaS Service.<br>VALUES:<br>• VNF_GENERIC_OAM_FUNCTION: a PaaS Service of type "VNF generic OAM function"<br>• CONFIGURATION_SERVER: a PaaS Service of type "Configuration Server"<br>• UNDEFINED: reserved for undefined types of PaaS Services |
| subType | M | 0..N | String | Sub-type of PaaS Service. It is used to further categorize the type of PaaS Service depending on the value of the attribute "paasServiceType". Multiple sub-types of PaaS Service can be specified.<br>It shall be present when "type = VNF_GENERIC_OAM_FUNCTION" or "type = UNDEFINED".<br>See note 1. |
| paasServiceProperties | M | 1 | Not specified | Properties of the PaaS Service which characterize the PaaS Service according to its type. |
| deploymentFormat | M | 1 | Enum | Format of deployment of the PaaS Service.<br>VALUES:<br>• VNF: when deployed as a VNF<br>• NFVI_RESOURCE: when deployed as an NFVI resource<br>• MCCO: when deployed as an MCCO |

| Attribute | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| deploymentDescriptor | M | 1 | Identifier | Identifier of the descriptor(s) for the deployment and lifecycle of the PaaS Service. See note 2. |
| deploymentFlavour | M | 0..N | Identifier | Identifier of the deployment flavours defined in the descriptor(s) for the deployment and lifecycle of the PaaS Service. See note 3. |
| usageFormat | M | 1 | Enum | The pre-defined usage format of the PaaS Service.<br>VALUES:<br>• VNF_COMMON: to use the PaaS Service as VNF Common Service<br>• VNF_DEDICATED: to use the PaaS Service as VNF Dedicated Service<br>• NS: to use the PaaS Service for any NS constituent other than a VNF<br>• UNDEFINED: no specific usage defined |
| interface | M | 1..N | PaasServiceInterface | Interfaces of the PaaS Service. |

NOTE 1: Multiple sub-types of PaaS Service can be specified to convey the possible grouping of several granular functions, e.g. a PaaS Service that is a grouping of two or more VNF generic OAM functions.
NOTE 2: For instance, this can be a reference to a VNFD identifier, an MCCO descriptor identifier, etc.
NOTE 3: For instance, in the case of deploying the PaaS Service as a VNF and the VNF is described by a VNFD, this can refer to the attribute "flavourId" of the "VnfDf" information element defined in clause 7.1.8.2 of ETSI GS NFV-IFA 011 [1].

## 9.5.3 PaasServiceInterface information element

### 9.5.3.1 Description

A PaasServiceInterface specifies information about an interface produced by the PaaS Service, or the requirements for interaction with other entities.

### 9.5.3.2 Attributes

The attributes of the PaasServiceInterface information element shall follow the indications provided in table 9.5.3.2-1.

**Table 9.5.3.2-1: Attributes of the PaasServiceInterface information element**

| Attribute | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| interfaceType | M | 1 | Enum | Type of interface according to the PaaS Services framework.<br>VALUES:<br>• IF-M<br>• IF-F1<br>• IF-F2<br>• IF-V |
| interfaceInfo | M | 1 | Not specified | Properties of the interface, such as protocols and data models used over the interface. When applicable, information about the version of the interface shall also be provided. |
| handleInfo | M | 0..1 | Not specified | Information to declare a handle according to this interface. It shall be present if the interface is used as a handle to access and use the PaaS Service. |

# 10 PaaS Services management functions interfaces

## 10.1 Introduction

For the management of PaaS Services (including VNF generic OAM functions), two management functions are defined: the PaaS Services Management (PSM) and the PaaS Service Repository (PSR). Functional requirements for PSR function and PSM are specified in clauses 17 and 18 of ETSI GS NFV-IFA 010 [i.9], respectively.

The functional capabilities of the two management functions are exposed to consumers via interfaces, whose requirements and operations are further specified in the subsequent clauses.

## 10.2 Interface requirements

### 10.2.1 Interface requirements for PSM

#### 10.2.1.1 Introduction

Clause 10.2.1 specifies the requirements of interfaces exposed by the PSM.

#### 10.2.1.2 PSM service requirements

Table 10.2.1.2-1 specifies requirements applicable to the services provided by the PSM.

**Table 10.2.1.2-1: PSM services requirements**

| Identifier | Requirement |
|---|---|
| Psm.Svc.001 | The PSM shall provide a PaaS Services lifecycle management service. |

#### 10.2.1.3 PaaS Services lifecycle management interface requirements

Table 10.2.1.3-1 specifies the requirements applicable to the PaaS Services lifecycle management interface produced by the PSM.

**Table 10.2.1.3-1: PaaS Services lifecycle management interface requirements**

| Identifier | Requirement |
|---|---|
| Psm.Svc.Psslcm.001 | The PaaS Services lifecycle management interface produced by the PSM shall support instantiating a PaaS Service. See note 1. |
| Psm.Svc.Pssicm.002 | The PaaS Services lifecycle management interface produced by the PSM shall support terminating a PaaS Service instance. See note 1. |
| Psm.Svc.Pssicm.003 | The PaaS Services lifecycle management interface produced by the PSM shall support scaling out/in an existing PaaS Service instance. See note 1. |
| Psm.Svc.Pssicm.004 | The PaaS Services lifecycle management interface produced by the PSM shall support subscription management to notifications about lifecycle events of PaaS Services. See note 2. |
| Psm.Svc.Pssicm.005 | The PaaS Services lifecycle management interface produced by the PSM shall support notifying subscribers about events related to the lifecycle of PaaS Services. |
| NOTE 1: The PSM interacts with responsible NFV-MANO functional blocks and functions for the execution of the corresponding lifecycle management. See clause 5.8 of ETSI GS NFV-IFA 010 [i.9].<br>NOTE 2: Subscription management includes: creation and termination of subscriptions to notifications. | |

### 10.2.2 Interface requirements for PSR

#### 10.2.2.1 Introduction

Clause 10.2.2 specifies the requirements of interfaces exposed by the PSR.

### 10.2.2.2 PSR service requirements

Table 10.2.2.2-1 specifies requirements applicable to the services provided by the PSR.

**Table 10.2.2.2-1: PSR services requirements**

| Identifier | Requirement |
|---|---|
| Psr.Svc.001 | The PSR shall provide a PaaS Services Descriptor (PSD) management service. |
| Psr.Svc.002 | The PSR shall provide a PaaS Services registration management service. |

### 10.2.2.3 PaaS Services descriptor management interface requirements

Table 10.2.2.3-1 specifies the requirements applicable to the interface of PaaS Services descriptor management produced by the PSR.

**Table 10.2.2.3-1: PaaS Services descriptor management interface requirements**

| Identifier | Requirement |
|---|---|
| Psr.Svc.Psdm.001 | The PaaS Services descriptor management interface produced by the PSR shall support onboarding a PSD to the repository. See notes 1 and 2. |
| Psr.Svc.Psdm.002 | The PaaS Services descriptor management interface produced by the PSR shall support deleting an onboarded PSD form the repository. |
| Psr.Svc.Psdm.003 | The PaaS Services descriptor management interface produced by the PSR shall support querying information about an onboarded PSD. |
| Psr.Svc.Psdm.004 | The PaaS Services descriptor management interface produced by the PSR shall support updating information about an onboarded PSD. See note 2. |
| Psr.Svc.Psdm.005 | The PaaS Services descriptor management interface produced by the PSR shall support fetching an onboarded PSD. |
| Psr.Svc.Psdm.006 | The PaaS Services descriptor management interface produced by the PSR shall support changing the state of an onboarded PSD. See note 3. |
| Psr.Svc.Psdm.007 | The PaaS Services descriptor management interface produced by the PSR shall support subscription management to notifications about PSD management events and changes. See note 4. |
| Psr.Svc.Psdm.008 | The PaaS Services descriptor management interface produced by the PSR shall support notifying subscribers about events related to PSD management events and changes. |
| NOTE 1: A PSD is managed as an archive unit including one or more PaaS Service descriptor files, and zero or more artifacts such as image files.<br>NOTE 2: The related operations shall support features of version control including identifying new versions of a PSD, keeping track of modifications to the information about an onboarded PSD, tracking alternative versions (e.g. under development), etc.<br>NOTE 3: Changing the state includes enabling/disabling an onboarded PSD.<br>NOTE 4: Subscription management includes: creation and termination of subscriptions to notifications. | |

### 10.2.2.4 PaaS Services registration management interface requirements

Table 10.2.2.4-1 specifies the requirements applicable to the interface of PaaS Services registration management produced by the PSR.

**Table 10.2.2.4-1: PaaS Services registration management interface requirements**

| Identifier | Requirement |
|---|---|
| Psr.Svc.Psrm.001 | The PaaS Services registration management interface produced by the PSR shall support registering to the PSR a consumable PaaS Service and PaaS Services instances upon request. |
| Psr.Svc.Psrm.002 | The PaaS Services registration management interface produced by the PSR shall support querying information about consumable PaaS Services and PaaS Services instances. |
| Psr.Svc.Psrm.003 | The PaaS Services registration management interface produced by the PSR shall support updating information about PaaS Services instances and the association of the PaaS Services instances to PaaS Services consumers. |
| Psr.Svc.Psrm.004 | The PaaS Services registration management interface produced by the PSR shall support requesting the creation of an identifier for a PaaS Service instance. |

# 10.3 Interfaces

## 10.3.1 PaaS Services lifecycle management interface

### 10.3.1.1 Description

The PaaS Services lifecycle management interface enables a consumer to request lifecycle of PaaS Services.

The following operations are defined for the present interface:

- Instantiate PaaS Service;
- Terminate PaaS Service;
- Scale PaaS Service;
- Subscribe;
- Terminate Subscriptions; and
- Notify.

PaaS Services lifecycle operations that imply resource fulfilment, the PSM interacts with the responsible NFV-MANO functional blocks and functions as specified in clause 5.8 of ETSI GS NFV-IFA 010 [i.9].

For a successful lifecycle of a PaaS Service, both PSD and the "deployment and lifecycle descriptor" shall be onboarded and enabled.

### 10.3.1.2 Instantiate PaaS Service operation

#### 10.3.1.2.1 Description

This operation instantiates a PaaS Service.

Table 10.3.1.2.1-1 lists the information flow exchanged between the consumer and the PSM, as producer of the interface.

**Table 10.3.1.2.1-1: Instantiate PaaS Service operation**

| Message | Requirement | Direction |
|---|---|---|
| InstantiatePaasServiceRequest | Mandatory | Consumer → PSM |
| InstantiatePaasServiceResponse | Mandatory | PSM → Consumer |

#### 10.3.1.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 10.3.1.2.2-1.

**Table 10.3.1.2.2-1: Instantiate PaaS Service operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| psdId | M | 1 | Identifier (Reference to Psd) | Identifier of the PSD which defines the PaaS Service to be instantiated. The identifier is provided by the PSD and known to the consumer via its exposure on the PaaS Services descriptor management service. |

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| flavourId | M | 0..1 | Identifier | Identifier of a specific deployment flavour of the PaaS Service to be instantiated. A value need not be provided in case there is an identified default deployment flavour of the PaaS Service, or there are no multiple deployment flavours defined in the PSD. |
| targetScale | M | 0..1 | Not specified | Information about the desired scale/capacity for the PaaS Service. |
| additionalParam | M | 0..N | KeyValuePair | Additional parameters passed as input to the instantiation process, specific to the PaaS Service being instantiated as declared in the PSD. |

#### 10.3.1.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 10.3.1.2.3-1.

**Table 10.3.1.2.3-1: Instantiate PaaS Service operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| paasServiceId | M | 1 | Identifier | Identifier of the PaaS Service instance. |
| lifecycleOperationOccurrenceId | M | 1 | Identifier | Identifier of the lifecycle operation occurrence. |

#### 10.3.1.2.4 Output results

In case of success, the PaaS Service instance has been instantiated, including the fulfilment of the resources as described in clause 10.3.1.1. In this case, an identifier has been assigned by the PSR upon request by the PSM. Also, the associated instance information has been registered into the PSR upon request by the PSM. In case of failure, appropriate error information shall be provided back to the consumer, either or both, via a response message or additional lifecycle event notifications.

If the lifecycle operation is processed and executed as a longer-running task, the PSM shall return identification information about the lifecycle operation that is being executed.

### 10.3.1.3 Terminate PaaS Service operation

#### 10.3.1.3.1 Description

This operation terminates an existing PaaS Service instance.

Table 10.3.1.3.1-1 lists the information flow exchanged between the consumer and the PSM, as producer of the interface.

**Table 10.3.1.3.1-1: Terminate PaaS Service operation**

| Message | Requirement | Direction |
|---|---|---|
| TerminatePaasServiceRequest | Mandatory | Consumer → PSM |
| TerminatePaasServiceResponse | Mandatory | PSM → Consumer |

#### 10.3.1.3.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 10.3.1.3.2-1.

**Table 10.3.1.3.2-1: Terminate PaaS Service operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| paasServiceId | M | 1 | Identifier | Identifier of the PaaS Service instance to terminate. |
| additionalParam | M | 0..N | KeyValuePair | Additional parameters passed as input to the termination process, specific to the PaaS Service being terminated as declared in the PSD. |

#### 10.3.1.3.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 10.3.1.3.3-1.

**Table 10.3.1.3.3-1: Terminate PaaS Service operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| lifecycleOperationOccurrenceId | M | 1 | Identifier | Identifier of the lifecycle operation occurrence. |

#### 10.3.1.3.4 Output results

In case of success, the PaaS Service instance has been terminated and all its resources released as described in clause 10.3.1.1. Also, the associated instance information has been de-registered into the PSR upon request by the PSM. In case of failure, appropriate error information shall be provided back to the consumer, either or both, via a response message or additional lifecycle event notifications.

If the lifecycle operation is processed and executed as a longer-running task, the PSM shall return identification information about the lifecycle operation that is being executed.

### 10.3.1.4 Scale PaaS Service operation

#### 10.3.1.4.1 Description

This operation scales a PaaS Service in its supported forms by the PaaS Service, such as scaling out/in or scaling up/down.

Table 10.3.1.4.1-1 lists the information flow exchanged between the consumer and the PSM, as producer of the interface.

**Table 10.3.1.4.1-1: Scale PaaS Service operation**

| Message | Requirement | Direction |
|---|---|---|
| ScalePaasServiceRequest | Mandatory | Consumer → PSM |
| ScalePaasServiceResponse | Mandatory | PSM → Consumer |

#### 10.3.1.4.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 10.3.1.4.2-1.

**Table 10.3.1.4.2-1: Scale PaaS Service operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| paasServiceId | M | 1 | Identifier | Identifier of the PaaS Service instance to scale. |
| targetScale | M | 1 | Not specified | Information about the desired new scale/capacity for the PaaS Service. |
| additionalParam | M | 0..N | KeyValuePair | Additional parameters passed as input to the scaling process, specific to the PaaS Service being scaled as declared in the PSD. |

#### 10.3.1.4.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 10.3.1.4.3-1.

**Table 10.3.1.4.3-1: Scale PaaS Service operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| lifecycleOperationOccurrenceId | M | 1 | Identifier | Identifier of the lifecycle operation occurrence. |

#### 10.3.1.4.4 Output results

In case of success, the PaaS Service instance has been scaled, including the resources fulfilment as described in clause 10.3.1.1. Also, the associated instance information has been registered into the PSR upon request by the PSM. In case of failure, appropriate error information shall be provided back to the consumer, either or both, via a response message or additional lifecycle event notifications.

If the lifecycle operation is processed and executed as a longer-running task, the PSM shall return identification information about the lifecycle operation that is being executed.

### 10.3.1.5 Subscribe operation

#### 10.3.1.5.1 Description

This operation enables the consumer to subscribe, with a filter, to notifications sent by the PSM on lifecycle events of PaaS Services.

Table 10.3.1.5.1-1 lists the information flow exchanged between the consumer and the PSM, as producer of the interface.

**Table 10.3.1.5.1-1: Subscribe operation**

| Message | Requirement | Direction |
|---|---|---|
| SubscribeRequest | Mandatory | Consumer → PSM |
| SubscribeResponse | Mandatory | PSM → Consumer |

#### 10.3.1.5.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 10.3.1.5.2-1.

**Table 10.3.1.5.2-1: Subscribe operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| filter | M | 1 | Filter | Input filter for selecting PaaS Services of interest and the specific types of lifecycle events to be notified about. |

#### 10.3.1.5.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 10.3.1.5.3-1.

**Table 10.3.1.5.3-1: Subscribe operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| subscriptionId | M | 1 | Identifier | Identifier of the subscription realized. |

#### 10.3.1.5.4 Output results

In case of success, the consumer is registered to receive notifications.

The results of the operation shall indicate if the subscription has been successful or not with a standard success/error result.

For a particular subscription, only notifications matching the filter will be delivered to the consumer.

### 10.3.1.6 Terminate Subscription operation

#### 10.3.1.6.1 Description

This operation enables the consumer to terminate a particular subscription.

Table 10.3.1.6.1-1 lists the information flow exchanged between the consumer and the PSM, as producer of the interface.

**Table 10.3.1.6.1-1: Terminate Subscription operation**

| Message | Requirement | Direction |
|---|---|---|
| TerminateSubscriptionRequest | Mandatory | Consumer → PSM |
| TerminateSubscriptionResponse | Mandatory | PSM → Consumer |

#### 10.3.1.6.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 10.3.1.6.2-1.

**Table 10.3.1.6.2-1: Terminate Subscription operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| subscriptionId | M | 1 | Identifier | Identifier of the subscription to terminate. |

#### 10.3.1.6.3 Output parameters

None.

#### 10.3.1.6.4 Output results

In case of success, the identified subscription has been terminated and does not exist anymore, and the consumer will not receive notifications related to the terminated subscription any longer.

The results of the operation shall indicate if the subscription termination has been successful or not with a standard success/error result.

#### 10.3.1.7 Notify operation

##### 10.3.1.7.1 Description

This operation notifies a subscriber about events related to PaaS Services lifecycle.

This operation distributes notifications to subscribers. It is a one-way operation issued by the producer (PSM) that cannot be invoked as an operation by a consumer. In order to receive notifications, the consumer has to perform an explicit "Subscribe Notifications" operation beforehand.

Table 10.3.1.7.1-1 lists the information flow exchanged between the consumer and the PSM, as producer of the interface.

**Table 10.3.1.7.1-1: Notify operation**

| Message | Requirement | Direction |
|---|---|---|
| Notify | Mandatory | PSM → Consumer |

The following notifications can be sent by this operation:

- PaasServiceLifecycleNotification (see clause 7.7.2).

### 10.3.2 PaaS Services descriptor management interface

#### 10.3.2.1 Description

The PaaS Services descriptor management interface enables a consumer to request management of PaaS Services Descriptors (PSD).

The following operations are defined for the present interface:

- Create PSD Info;
- Upload PSD;
- Delete PSD;
- Query PSD Info;
- Update PSD Info;
- Fetch PSD;
- Subscribe;
- Terminate Subscription; and
- Notify.

#### 10.3.2.2 Create PSD Info operation

##### 10.3.2.2.1 Description

This operation creates a PSD information object in the PSR for one or more versions of a PSD to be uploaded.

A PSD information object enables version control, i.e. any changes to a PSD information object with upload and update of PSDs are versioned.

Table 10.3.2.2.1-1 lists the information flow exchanged between the consumer and the PSR, as producer of the interface.

**Table 10.3.2.2.1-1: Create PSD Info operation**

| Message | Requirement | Direction |
|---|---|---|
| CreatePsdInfoRequest | Mandatory | Consumer → PSR |
| CreatePsdInfoResponse | Mandatory | PSR → Consumer |

#### 10.3.2.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 10.3.2.2.2-1.

**Table 10.3.2.2.2-1: Create PSD Info operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| userDefinedData | M | 0..N | KeyValuePair | User defined data to be added to the PSD information object. |

#### 10.3.2.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 10.3.2.2.3-1.

**Table 10.3.2.2.3-1: Create PSD Info operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| psdInfoObjectId | M | 1 | Identifier | Identifier of the PSD information object created by the PSR. |

#### 10.3.2.2.4 Output results

In case of success, the PSD information object has been created.

The psdInfoObjectId is only returned when the operation has been successful.

In case of failure, appropriate error information shall be provided back to the consumer.

### 10.3.2.3 Upload PSD operation

#### 10.3.2.3.1 Description

This operation uploads PSD file(s) to the PSR. The process of uploading the PSD and validating it is regarded as the "onboarding of a PSD".

This operation also enables the case of uploading new versions (e.g. with modifications) of PSDs.

Table 10.3.2.3.1-1 lists the information flow exchanged between the consumer and the PSR, as producer of the interface.

**Table 10.3.2.3.1-1: Upload PSD operation**

| Message | Requirement | Direction |
|---|---|---|
| UploadPsdRequest | Mandatory | Consumer → PSR |
| UploadPsdResponse | Mandatory | PSR → Consumer |

#### 10.3.2.3.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 10.3.2.3.2-1.

**Table 10.3.2.3.2-1: Upload PSD operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| psdInfoObjectId | M | 1 | Identifier | Identifier of the PSD information object to associate the PSD file(s). |
| psdFiles | M | 1 | Not specified | The PSD file(s) to be uploaded. |
| versionTag | M | 1 | String | A version tag for the PSD being uploaded. |
| userDefinedData | M | 0..N | KeyValuePair | User defined data to be added to the PSD information. |

#### 10.3.2.3.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 10.3.2.3.3-1.

**Table 10.3.2.3.3-1: Upload PSD operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| psdInfoObjectVersion | M | 1 | Not specified | Version of the PSD information object resulted from its modification, in this case by uploading the PSD. |

#### 10.3.2.3.4 Output results

In case of success, the PSD file(s) have been uploaded to the PSR. The PSD files have been checked and summed, and the PSD information object versioned. After uploading, the state of the specified version of PSD in the PSD information object is "ENABLED", enabling the PSD to be used for deploying corresponding PaaS Services.

In case of failure, appropriate error information shall be provided back to the consumer.

### 10.3.2.4 Delete PSD operation

#### 10.3.2.4.1 Description

This operation deletes one or more PSD(s). The associated PSD information objects are also deleted.

A PSD can only be deleted when there is no PaaS Service instance created based on the PSD.

NOTE: It is part of the protocol design whether this operation is modelled as a "bulk" operation that enables to delete multiple PSDs in one request, or a series of request that delete one PSD at a time.

Table 10.3.2.4.1-1 lists the information flow exchanged between the consumer and the PSR, as producer of the interface.

**Table 10.3.2.4.1-1: Delete PSD operation**

| Message | Requirement | Direction |
|---|---|---|
| DeletePsdRequest | Mandatory | Consumer → PSR |
| DeletePsdResponse | Mandatory | PSR → Consumer |

#### 10.3.2.4.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 10.3.2.4.2-1.

**Table 10.3.2.4.2-1: Delete PSD operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| psdInfoObjectId | M | 1..N | Identifier | Identifier of the PSD information objects to be deleted. |

#### 10.3.2.4.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 10.3.2.4.3-1.

**Table 10.3.2.4.3-1: Delete PSD operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| deletedPsdInfoObjectId | M | 1..N | Identifier | Identifier of the PSD information objects of the associated deleted PSDs. |

#### 10.3.2.4.4 Output results

In case of success, the PSDs and their corresponding information objects have been deleted from the PSR.

In case of failure, appropriate error information shall be provided back to the consumer.

### 10.3.2.5 Query PSD Info operation

#### 10.3.2.5.1 Description

This operation enables a consumer to query the PSR information concerning one or more PSD.

Table 10.3.2.5.1-1 lists the information flow exchanged between the consumer and the PSR, as producer of the interface.

**Table 10.3.2.5.1-1: Query PSD Info operation**

| Message | Requirement | Direction |
|---|---|---|
| QueryPsdInfoRequest | Mandatory | Consumer → PSR |
| QueryPsdInfoResponse | Mandatory | PSR → Consumer |

#### 10.3.2.5.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 10.3.2.5.2-1.

**Table 10.3.2.5.2-1: Query PSD Info operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| filter | M | 1 | Filter | Filter defining the PSD information objects on which the query applies, based on attributes of the PSD information objects. It can also be used to specify one or more PSD information objects to be queried by providing their identifiers and versions. |
| attributeSelector | M | 0..N | String | List of attribute names of the PSD information objects. If present, only these attributes are returned for the PSD information objects matching the filter. If absent, the complete PSD information objects are returned. |

#### 10.3.2.5.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 10.3.2.5.3-1.

**Table 10.3.2.5.3-1: Query PSD Info operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| queryResult | M | 0..N | PsdInfoObject | Data of the PSD information objects matching the input filter, and attribute selection if present. |

### 10.3.2.5.4 Output results

In case of success, the consumer has queried the PSD information objects from the PSR. For a particular query, the PSD information objects that the consumer has access to and that match the filter are returned.

In case of failure, appropriate error information shall be provided back to the consumer.

## 10.3.2.6 Update PSD Info operation

### 10.3.2.6.1 Description

This operation enables a consumer to modify user defined data and/or the operational state of an existing PSD information object.

Table 10.3.2.6.1-1 lists the information flow exchanged between the consumer and the PSR, as producer of the interface.

**Table 10.3.2.6.1-1: Update PSD Info operation**

| Message | Requirement | Direction |
|---|---|---|
| UpdatePsdInfoRequest | Mandatory | Consumer → PSR |
| UpdatePsdInfoResponse | Mandatory | PSR → Consumer |

### 10.3.2.6.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 10.3.2.6.2-1.

**Table 10.3.2.6.2-1: Update PSD Info operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| psdInfoObjectId | M | 1 | Identifier | Identifier of the PSD information object to be updated. |
| versionTag | M | 1 | Not specified | Version tag to which the update concerns. |
| userDefinedData | M | 0..N | KeyValuePair | User defined data to be updated. See note. |
| operationalState | M | 0..1 | Enum | Operational state of the on-boarded PSD.<br>VALUES:<br>• ENABLED<br>• DISABLED<br>See note. |
| NOTE: At least one of these parameters shall be present. | | | | |

### 10.3.2.6.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 10.3.2.6.3-1.

**Table 10.3.2.6.3-1: Update PSD Info operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| psdInfoObjectVersion | M | 1 | Not specified | Version of the PSD information object resulted from its modification, in this case by updating the PSD. |

#### 10.3.2.6.4 Output results

In case of success, the PSD information object has been updated according to the input request.

In case of failure, appropriate error information shall be provided back to the consumer.

### 10.3.2.7 Fetch PSD operation

#### 10.3.2.7.1 Description

This operation enables a consumer to fetch a specific PSD version from the PSR.

Table 10.3.2.7.1-1 lists the information flow exchanged between the consumer and the PSR, as producer of the interface.

**Table 10.3.2.7.1-1: Fetch PSD operation**

| Message | Requirement | Direction |
|---|---|---|
| FetchPsdRequest | Mandatory | Consumer → PSR |
| FetchPsdResponse | Mandatory | PSR → Consumer |

#### 10.3.2.7.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 10.3.2.7.2-1.

**Table 10.3.2.7.2-1: Fetch PSD operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| psdInfoObjectId | M | 1 | Identifier | Identifier of the PSD information object associated to the PSD to be fetched. |
| versionTag | M | 1 | Not specified | Version tag of the PSD to be fetched. |

#### 10.3.2.7.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 10.3.2.7.3-1.

**Table 10.3.2.7.3-1: Fetch PSD operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| psdFiles | M | 1 | Not specified | The fetched PSD files. |

#### 10.3.2.7.4 Output results

In case of success, the consumer has fetched the PSD from the PSR.

In case of failure, appropriate error information shall be provided back to the consumer.

### 10.3.2.8 Subscribe operation

#### 10.3.2.8.1 Description

This operation enables the consumer to subscribe, with a filter, to notifications sent by the PSR about PSD management events and changes.

Table 10.3.2.8.1-1 lists the information flow exchanged between the consumer and the PSR, as producer of the interface.

**Table 10.3.2.8.1-1: Subscribe operation**

| Message | Requirement | Direction |
|---|---|---|
| SubscribeRequest | Mandatory | Consumer → PSR |
| SubscribeResponse | Mandatory | PSR → Consumer |

#### 10.3.2.8.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 10.3.2.8.2-1.

**Table 10.3.2.8.2-1: Subscribe operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| filter | M | 1 | Filter | Input filter for selecting PSDs of interest and the specific types of events to be notified about. |

#### 10.3.2.8.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 10.3.2.8.3-1.

**Table 10.3.2.8.3-1: Subscribe operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| subscriptionId | M | 1 | Identifier | Identifier of the subscription realized. |

#### 10.3.2.8.4 Output results

In case of success, the consumer is registered to receive notifications.

The results of the operation shall indicate if the subscription has been successful or not with a standard success/error result.

For a particular subscription, only notifications matching the filter will be delivered to the consumer.

### 10.3.2.9 Terminate Subscription operation

#### 10.3.2.9.1 Description

This operation enables the consumer to terminate a particular subscription.

Table 10.3.2.9.1-1 lists the information flow exchanged between the consumer and the PSR, as producer of the interface.

**Table 10.3.2.9.1-1: Terminate Subscription operation**

| Message | Requirement | Direction |
|---|---|---|
| TerminateSubscriptionRequest | Mandatory | Consumer → PSR |
| TerminateSubscriptionResponse | Mandatory | PSR → Consumer |

#### 10.3.2.9.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 10.3.2.9.2-1.

**Table 10.3.2.9.2-1: Terminate Subscription operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| subscriptionId | M | 1 | Identifier | Identifier of the subscription to terminate. |

#### 10.3.2.9.3 Output parameters

None.

#### 10.3.2.9.4 Output results

In case of success, the identified subscription has been terminated and does not exist anymore, and the consumer will not receive notifications related to the terminated subscription any longer.

The results of the operation shall indicate if the subscription termination has been successful or not with a standard success/error result.

### 10.3.2.10 Notify operation

#### 10.3.2.10.1 Description

This operation notifies a subscriber about events related to PSD management and changes.

This operation distributes notifications to subscribers. It is a one-way operation issued by the producer (PSR) that cannot be invoked as an operation by a consumer. In order to receive notifications, the consumer has to perform an explicit "Subscribe Notifications" operation beforehand.

Table 10.3.2.10.1-1 lists the information flow exchanged between the consumer and the PSR, as producer of the interface.

**Table 10.3.2.10.1-1: Notify operation**

| Message | Requirement | Direction |
|---|---|---|
| Notify | Mandatory | PSR → Consumer |

The following notifications can be sent by this operation:

- PsdOnboardingNotification (see clause 7.8.2).
- PsdChangeNotification (see clause 7.8.3).

## 10.3.3 PaaS Services registration management interface

### 10.3.3.1 Description

The PaaS Services registration management interface enables a consumer to perform management tasks regarding the registration of PaaS Services instances. This includes consumable PaaS Service instances, which are deployed and available for consumption, and inventory of PaaS Service instances that have been deployed and are associated to PaaS Services consumers, i.e. that are in use.

The following operations are defined for the present interface:

- Register PaaS Service;
- Query Info Registered PaaS Service;
- Update PaaS Service Registration; and
- Create PaaS Service Identifier.

### 10.3.3.2 Register PaaS Service operation

#### 10.3.3.2.1 Description

This operation enables a consumer to register a PaaS Service instance to the PSR.

Table 10.3.3.2.1-1 lists the information flow exchanged between the consumer and the PSR, as producer of the interface.

**Table 10.3.3.2.1-1: Register PaaS Service operation**

| Message | Requirement | Direction |
|---|---|---|
| RegisterPaasServiceRequest | Mandatory | Consumer → PSR |
| RegisterPaasServiceResponse | Mandatory | PSR → Consumer |

#### 10.3.3.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 10.3.3.2.2-1.

**Table 10.3.3.2.2-1: Register PaaS Service operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| paasServiceId | M | 1 | Identifier | Identifier of the PaaS Service instance to be registered. |
| paasServiceInfo | M | 1 | PaasServiceInfo | Information of the PaaS Service instance to be registered. |
| usageFormat | M | 1 | Enum | The usage format of the PaaS Service instance.<br>VALUES:<br>• VNF_COMMON: to use the PaaS Service as VNF Common Service<br>• VNF_DEDICATED: to use the PaaS Service as VNF Dedicated Service<br>• NS: to use the PaaS Service for any NS constituent other than a VNF<br>• UNDEFINED: no specific usage defined |
| location | M | 1 | Not specified | Information about the location(s) where the PaaS Service is instantiated. This can be in the form of some identification information of the NFVI-PoP or CIS cluster, or geographical location information. |
| deploymentHandle | M | 1 | Not specified | Information about the deployment of the PaaS Service and its form. |
| paasServiceHandle | M | 1 | Not specified | A handle enabling the access and use of the PaaS Service instance. |
| scaleInfo | M | 1 | Not specified | Information about the scale/capacity of the PaaS Service instance. |

#### 10.3.3.2.3 Output parameters

None.

#### 10.3.3.2.4 Output results

In case of success, the PaaS Service instance has been registered into the PSR.

In case of failure, appropriate error information shall be provided back to the consumer.

### 10.3.3.3 Query Info Registered PaaS Service operation

#### 10.3.3.3.1 Description

This operation enables a consumer to query the PSR information about registered PaaS Service instances.

Table 10.3.3.3.1-1 lists the information flow exchanged between the consumer and the PSR, as producer of the interface.

**Table 10.3.3.3.1-1: Query Info Registered PaaS Service operation**

| Message | Requirement | Direction |
|---|---|---|
| QueryInfoRegisteredPaasServiceRequest | Mandatory | Consumer → PSR |
| QueryInfoRegisteredPaasServiceResponse | Mandatory | PSR → Consumer |

#### 10.3.3.3.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 10.3.3.3.2-1.

**Table 10.3.3.3.2-1: Query Info Registered PaaS Service operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| filter | M | 1 | Filter | Filter defining the registered PaaS Service instances on which the query applies, based on attributes of the PaasServiceEntry information objects. It can also be used to specify one or more entries to be queried by providing their identifiers. |
| attributeSelector | M | 0..N | String | List of attribute names of the PaasServiceEntry information objects. If present, only these attributes are returned for the registered PaaS Service instances matching the filter. If absent, the complete PaasServiceEntry information objects are returned. |

#### 10.3.3.3.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 10.3.3.3.3-1.

**Table 10.3.3.3.3-1: Query Info Registered PaaS Service operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| queryResult | M | 0..N | PaasServiceEntry | Information of the registered PaaS Service instance. |

#### 10.3.3.3.4 Output results

In case of success, the consumer has queried the registered PaaS Service entries from the PSR. For a particular query, the registered PaaS Service entries that the consumer has access to and that match the filter are returned.

In case of failure, appropriate error information shall be provided back to the consumer.

### 10.3.3.4 Update Registered PaaS Service operation

#### 10.3.3.4.1 Description

This operation enables a consumer to update information about a registered PaaS Service instance to the PSR.

Table 10.3.3.4.1-1 lists the information flow exchanged between the consumer and the PSR, as producer of the interface.

**Table 10.3.3.4.1-1: Update Registered PaaS Service operation**

| Message | Requirement | Direction |
|---|---|---|
| UpdateRegisteredPaasServiceRequest | Mandatory | Consumer → PSR |
| UpdateRegisteredPaasServiceResponse | Mandatory | PSR → Consumer |

### 10.3.3.4.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 10.3.3.4.2-1.

**Table 10.3.3.4.2-1: Update Registered PaaS Service operation input parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| paasServiceId | M | 1 | Identifier | Identifier of the registered PaaS Service instance to be updated. |
| usageFormat | M | 0..1 | Enum | New value for the usage format of the PaaS Service instance.<br>VALUES:<br>• VNF_COMMON: to use the PaaS Service as VNF Common Service<br>• VNF_DEDICATED: to use the PaaS Service as VNF Dedicated Service<br>• NS: to use the PaaS Service for any NS constituent other than a VNF<br>• UNDEFINED: no specific usage defined<br>See note. |
| scaleInfo | M | 0..1 | Not specified | New values of information about the scale/capacity of the PaaS Service instance.<br>See note. |
| paasServiceHandle | M | 0..1 | Not specified | New values for the PaaS Service handle. The type and format of the handle depends on the form that the PaaS Service is formed.<br>See note. |
| additionalInfo | M | 0..1 | Not specified | New values for the additional information which is specific to the PaaS Service and its type.<br>See note. |
| consumerId | M | 0..N | Identifier | New values of identifiers of PaaS Service consumers. The attribute indicates consumers to be removed and/or added.<br>See note. |
| NOTE: At least one of the attributes shall be present in a request. | | | | |

### 10.3.3.4.3 Output parameters

None.

### 10.3.3.4.4 Output results

In case of success, the PaaS Service instance has been registration has been updated in the PSR.

In case of failure, appropriate error information shall be provided back to the consumer.

### 10.3.3.5 Create PaaS Service Identifier operation

#### 10.3.3.5.1 Description

This operation enables a consumer to create and retrieve from the PSR an identifier for a PaaS Service instance.

Table 10.3.3.5.1-1 lists the information flow exchanged between the consumer and the PSR, as producer of the interface.

**Table 10.3.3.5.1-1: Create PaaS Service Identifier operation**

| Message | Requirement | Direction |
|---|---|---|
| CreatePaasServiceIdentifierRequest | Mandatory | Consumer → PSR |
| CreatePaasServiceIdentifierResponse | Mandatory | PSR → Consumer |

#### 10.3.3.5.2 Input parameters

None.

#### 10.3.3.5.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 10.3.3.5.3-1.

**Table 10.3.3.5.3-1: Create PaaS Service Identifier operation output parameters**

| Parameter | Qualifier | Cardinality | Content | Description |
|---|---|---|---|---|
| paasServiceId | M | 1 | Identifier | Identifier of the PaaS Service instance allocated by the PSR. |

#### 10.3.3.5.4 Output results

In case of success, a PaaS Service identifier has been created by the PSR and it is kept by the PSR for subsequent registration procedures.

In case of failure, appropriate error information shall be provided back to the consumer.

# Annex A (informative): Aspects of PaaS Services management

## A.1 Procedures related to PaaS Services management

### A.1.1 Introduction

The present clause describes procedures related to PaaS Services management. The purpose is to illustrate the interactions involving PSM, PSR, and other NFV-MANO functional blocks and functions.

### A.1.2 Instantiation and registration of a PaaS Service

Figure A.1.2-1 illustrates the high-level procedure of instantiation and registration of a new PaaS Service.

As a pre-condition for this procedure, the PSD related to the PaaS Service to instantiate has been onboarded and is available to the PSR.

The NFVO triggers the procedure when determining that a new PaaS Service is to be instantiated. The source of the trigger could be some internal orchestration and management procedure or an explicit request by an external consumer, such as the OSS/BSS.

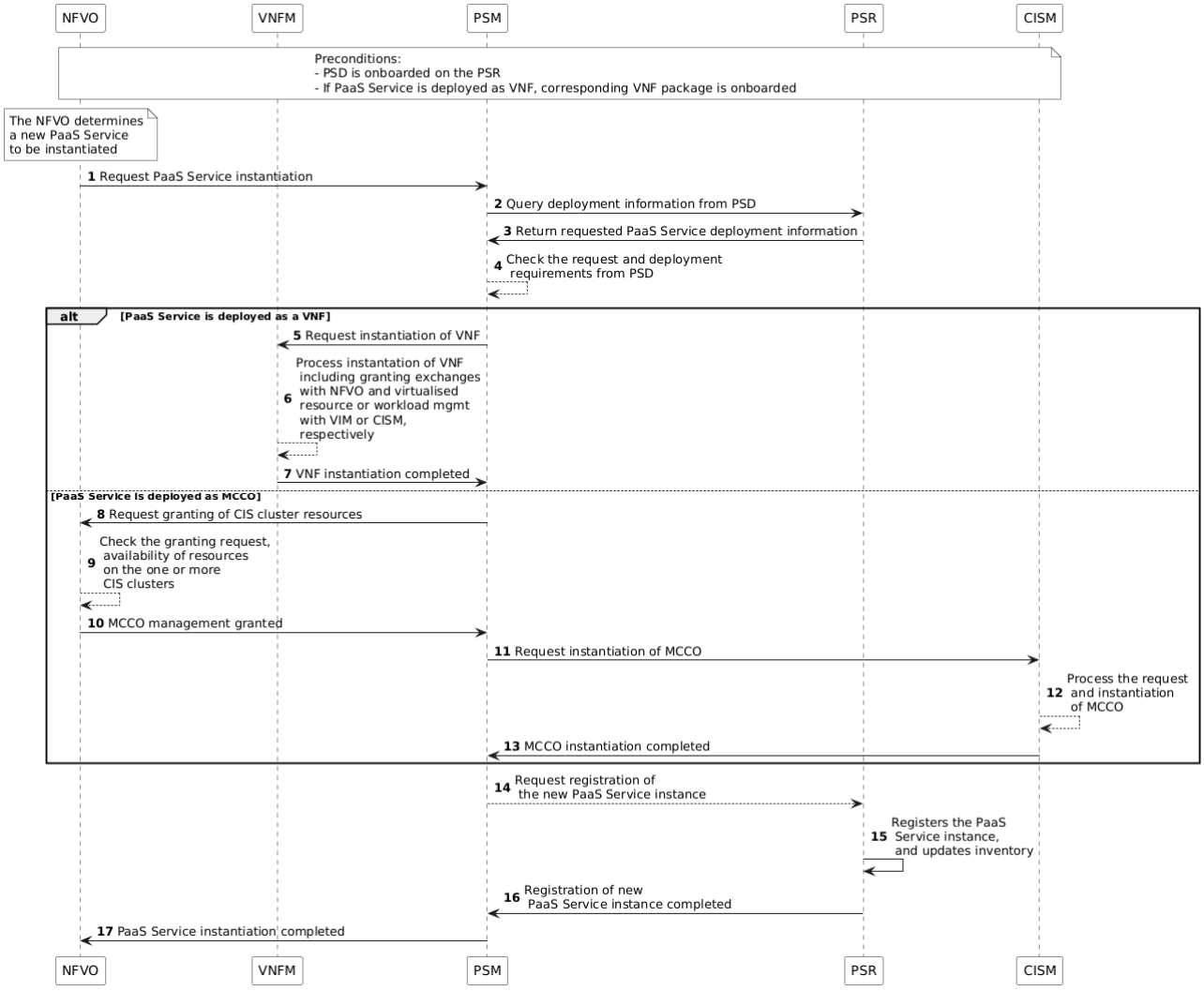


**Figure A.1.2-1: High-level procedure of instantiation and registration of a new PaaS Service**

The procedure of instantiation and registration of a new PaaS Service comprises the following steps:

1) The NFVO requests the PSM to instantiate a new PaaS Service. The request includes information about the type of PaaS Service.

2) The PSM queries the PSR for deployment information contained in the PSD for the PaaS Service to instantiate.

3) The PSR returns to the PSM the PaaS Service deployment information. This can contain deployment requirements and artefacts necessary for the deployment of the PaaS Service.

4) The PSM processes the deployment information and determines the form of deployment of the PaaS Service.

Depending on the form of deployment of the PaaS Service:

5) In case the PaaS Service is to be deployed as a VNF, the PSM requests the VNFM to instantiate the corresponding VNF.

6) The VNFM processes the instantiation request. The VNFM interacts with the NFVO for the necessary granting exchanges, and if the granting is successful, then with the VIM or CISM for the deployment of the virtualised resources or workloads, respectively.

7) Once the instantiation of the VNF completes, the VNFM confirms to the PSM that the VNF instantiation realizing the PaaS Service is complete.

8) In case the PaaS Service is to be deployed as a set of one or more MCCO, first the PSM requests granting of the operation and resource to the NFVO.

9) The NFVO process the granting request, checks the availability of resources on the CIS cluster(s) and determines the placement requirements.

10) The NFVO confirms to the PSM the granting of CIS cluster resources.

11) The PSM requests to the CISM the instantiation of MCCOs.

12) The CISM processes the instantiation request. The CISM allocates the resources in the CIS cluster and the creation of the corresponding MCCO.

13) The CISM confirms to the PSM the instantiation of the MCCO whose workloads realize the PaaS Service.

14) The PSM requests the PSR to register the new consumable PaaS Service instance.

15) The PSR registers the new consumable PaaS Service instance and updates its inventory.

16) The PSR confirms to the PSM the PaaS Service registration.

17) The PSM confirms to the NFVO the instantiation of the PaaS Service requested.

## A.1.3 Create PSD Information

Figure A.1.3-1 illustrates the high-level procedure of create PSD information in PSR.

The source of the trigger for the create PSD information could be some management procedure or an explicit request by an external consumer to NFVO, such as the OSS/BSS.

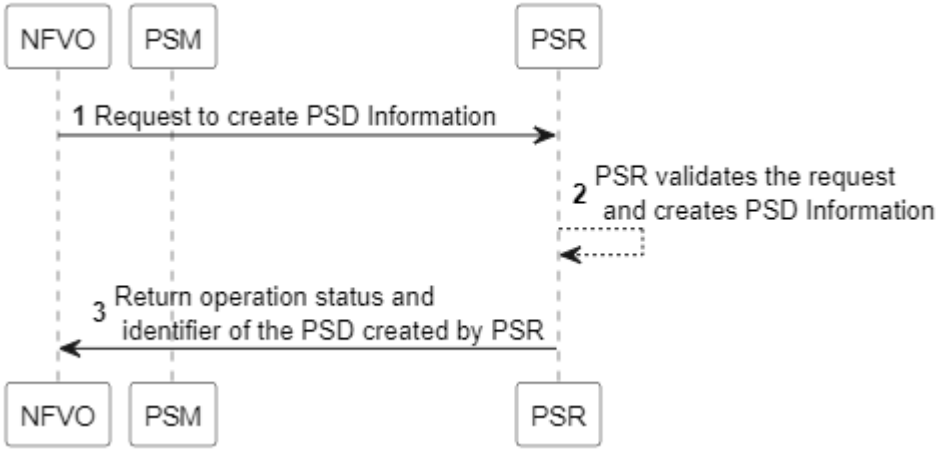


**Figure A.1.3-1: High-level procedure of create PSD information**

The procedure of create PSD information comprises the following steps:

1) The NFVO requests the PSR to create PSD information. The request includes information about the PSD such as user defined data to be added to the PSD information.

2) The PSR validates the request and creates PSD information.

3) The PSR returns the identifier of the PSD information created by PSR to NFVO.

## A.1.4 Upload PSD

Figure A.1.4-1 illustrates the high-level procedure of upload PSD information in PSR. As a pre-condition for this procedure, the PSD related to the upload has been created and is available to the PSR. The source of the trigger for the upload PSD information could be some management procedure or an explicit request by an external consumer to NFVO, such as the OSS/BSS.

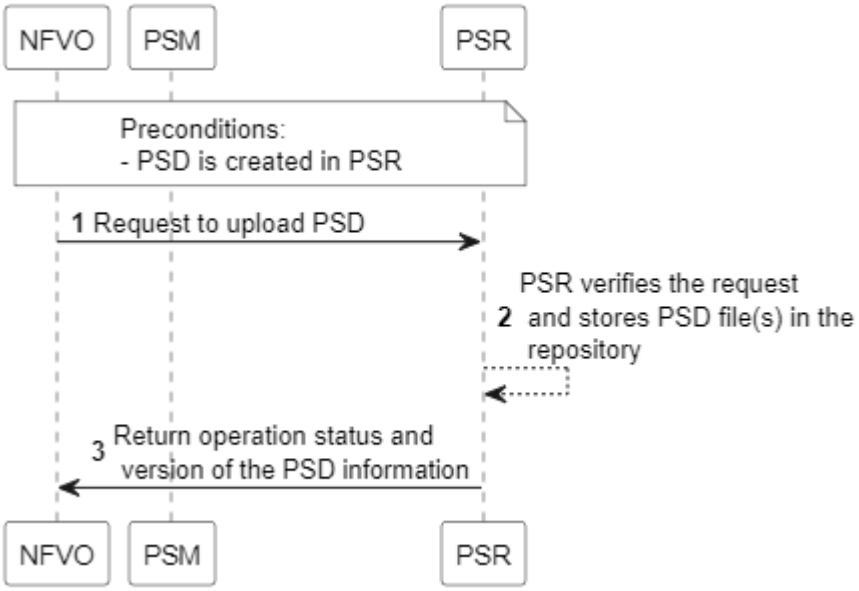


**Figure A.1.4-1: High-level procedure of upload PSD**

The procedure of upload PSD comprises the following steps:

1) The NFVO requests the PSR to upload PSD information. The request includes information about the PSD such as identifier of the PSD information, PSD file to be uploaded, a version tag for the PSD being uploaded and user defined data to be added to the PSD information.

2) The PSR verifies the request and uploads PSD file(s) in the repository. Once the PSD file(s) are uploaded, the state of the specified PSD version in the PSD information is set to "ENABLED".

3) The PSR returns the operation status and version of the PSD information to NFVO.

## A.1.5 Query PSD information of an instantiated PaaS Service

Figure A.1.5-1 illustrates the high-level procedure of query PSD information of an instantiated PaaS Service.

As a pre-condition for this procedure, the PSD related to the PaaS Service has been onboarded and is available to the PSR.

The source of the trigger for the query PSD information could be some internal orchestration and management procedure or an explicit request by an external consumer to NFVO, such as the OSS/BSS.

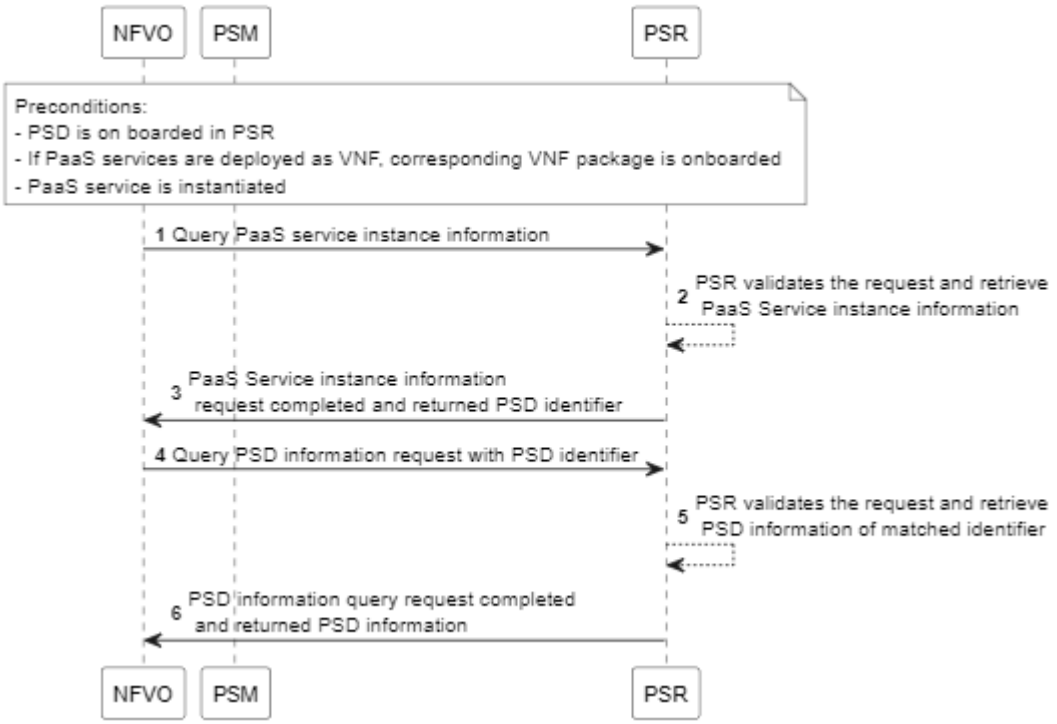


**Figure A.1.5-1: High-level procedure of query PSD information of PaaS service**

The procedure of querying PSD information of a PaaS Service comprises the following steps:

1) The NFVO requests the PSR to query PaaS service instance information. The request includes information about the PaaS Service.

NOTE: The option that NFVO calls the descriptor management interface exposed by PSR is also possible (for example for not instantiated PaaS Services).

2) The PSR validates the request and retrieve PaaS service instance information. The PaaS service instance information includes identifier of the PSD from which the PaaS Service instance is based on, the type of PaaS service, and other relevant information.

3) The PSR returns the identifier of the PSD information of PaaS Service to NFVO.

4) The NFVO requests the PSR to query PSD information. The request includes information about PSD identifier.

5) The PSR validates the request and retrieve PSD information of matched request PSD identifier.

6) The PSR returns data of PSD information to NFVO. PSD information includes identifier of the PSD information, version of the PSD information, files confirming the PSD, operation state of the PSD, user defined data and other relevant information.

## A.1.6 Delete PSD

Figure A.1.6-1 illustrates the high-level procedure of delete PSD information in PSR.

As a pre-condition for this procedure, the PSD related to the delete has been created and is available to the PSR and no PaaS Service is currently instantiated based on the PSD.

The source of the trigger for the delete PSD information could be some management procedure or an explicit request by an external consumer to NFVO, such as the OSS/BSS.

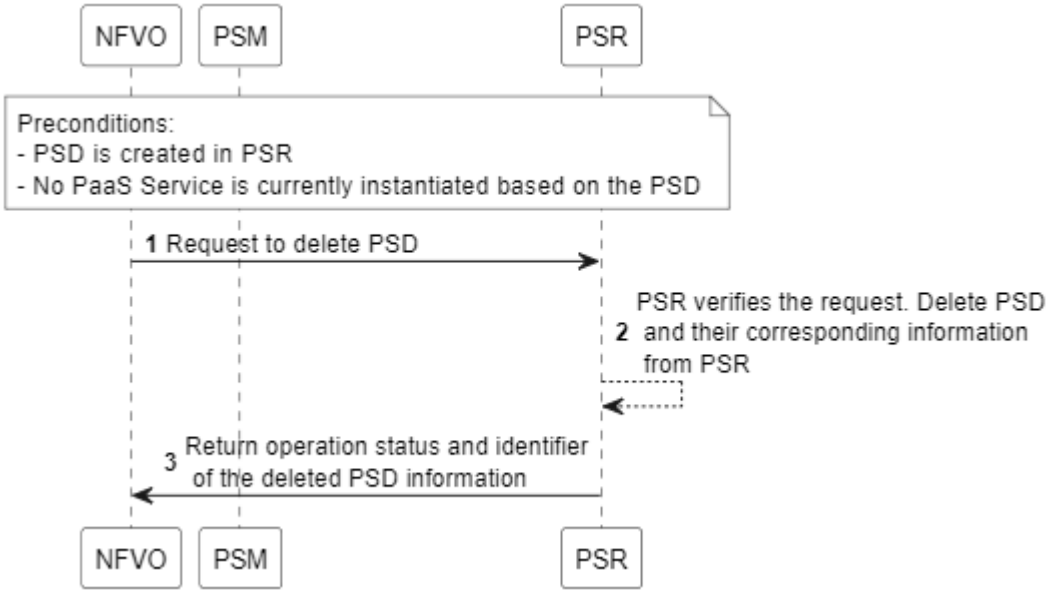


**Figure A.1.6-1: High-level procedure of delete PSD**

The procedure of delete PSD comprises the following steps:

1) The NFVO requests the PSR to delete PSD information. The request includes information about the PSD such as identifier of the PSD information to be deleted.

NOTE: This operation can delete one or more PSD(s), along with their associated information.

2) The PSR verifies the request and deletes the PSD along with their corresponding information from the PSR.

3) The PSR returns the operation status and identifier of the deleted PSD information to NFVO.

## A.1.7 Termination and de-registration of a PaaS Service

Figure A.1.7-1 illustrates the high-level procedure of termination and de-registration of a PaaS Service.

As a pre-conditions for this procedure, the PaaS Service to terminate has been instantiated and is available to the PSR.

In addition, the PaaS service instance is not consumed by any consumer.

The NFVO triggers the procedure when determining that a PaaS Service is to be terminated. The source of the trigger could be some internal orchestration and management procedure or an explicit request by an external consumer, such as the OSS/BSS.

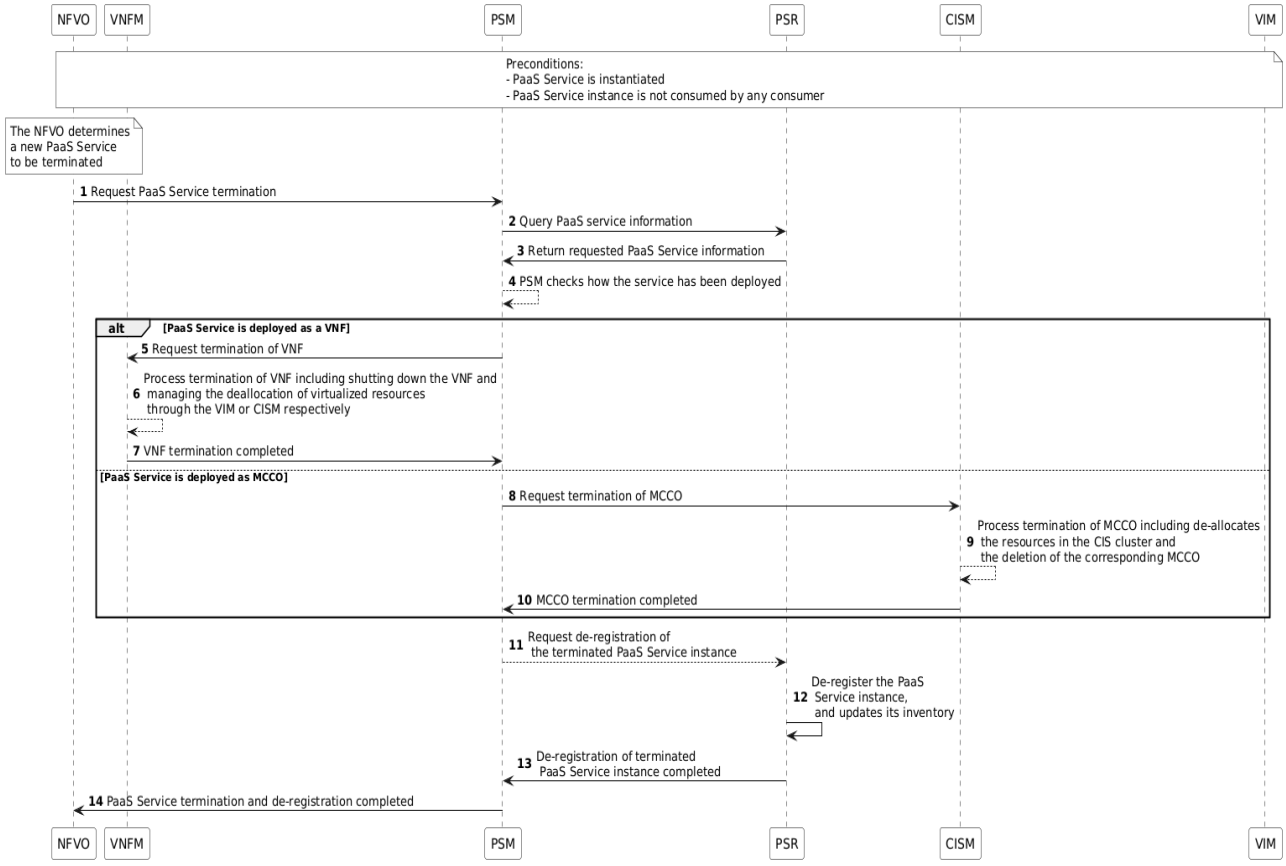


**Figure A.1.7-1: High-level procedure of termination and de-registration of a PaaS Service**

The procedure of termination and de-registration of a PaaS Service comprises the following steps:

1) The NFVO sends a request to the PSM to initiate the termination of the PaaS Service. The request includes identifier of the PaaS Service instance to terminate.

2) The PSM queries the PSR for PaaS service information for the PaaS Service to terminate.

3) The PSR returns to the PSM the PaaS Service information. This can contain artefacts necessary for the termination of the PaaS Service.

4) PSM checks how the PaaS service has been deployed

Depending on the form of deployment of the PaaS Service:

5) In case the PaaS Service is deployed as a VNF, the PSM requests to VNFM to terminate corresponding VNF.

6) The VNFM processes the termination request, which includes shutting down the VNF and managing the deallocation of virtualised resources through the VIM or CISM (Container Infrastructure Service Manager).

7) Once the termination of the VNF completes, the VNFM confirms to the PSM that the VNF termination realizing the PaaS Service is complete.

8) In case the PaaS Service is deployed as a set of one or more MCCO, the PSM requests to the CISM the termination of MCCO.

9) The CISM processes the termination request. The CISM de-allocates the resources in the CIS cluster and the deletion of the corresponding MCCO.

10) The CISM confirms to the PSM the termination of the MCCO.

11) The PSM requests the PSR to de-register the PaaS Service instance.

12) The PSR de-registers the consumable PaaS Service instance and updates its inventory.

13) The PSR confirms to the PSM the PaaS Service de-registration.

14) The PSM confirms to the NFVO the termination and de-registration of the PaaS Service requested.

## A.1.8 Subscription for PaaS Service lifecycle events

Figure A.1.8-1 illustrates the high-level procedure of subscribe, with a filter, to notifications sent by the PSM on lifecycle events of PaaS Services.

The source of the trigger for the subscription could be some management procedure or an explicit request by an external consumer to NFVO, such as the OSS/BSS.

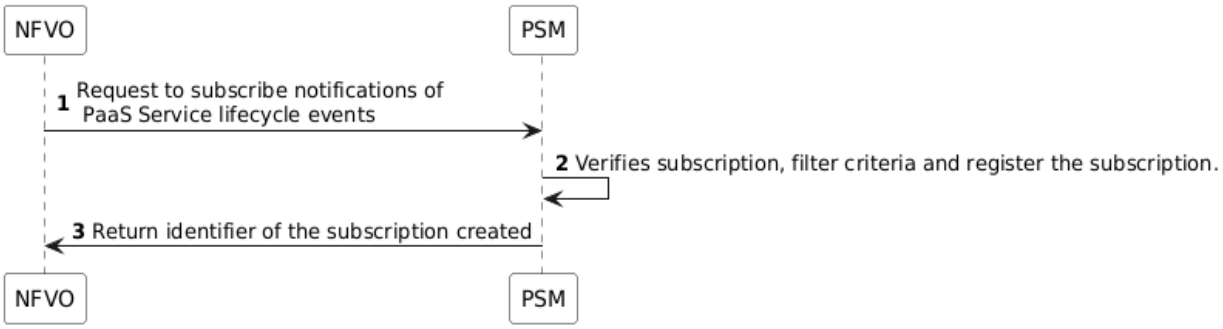


**Figure A.1.8-1: High-level procedure of subscription of a PaaS Service lifecycle events**

The procedure of subscription for PaaS Service lifecycle events comprises the following steps:

1) The NFVO requests the PSM to subscribe notification of PaaS Service lifecycle events. The request includes information about input filter for selecting PaaS Services of interest and the specific types of lifecycle events to be notified about.

2) The PSM verifies the request for subscription, filter criteria and register the subscription.

3) The PSM returns the identifier of the subscription created to NFVO.

## A.1.9 Scale out of a PaaS Service

Figure A.1.9-1 illustrates the high-level procedure of scale out of a PaaS Service. As a pre-conditions for this procedure, the PaaS Service to scale out has been instantiated and is available to the PSR. In addition, the scaling information is available in the descriptor.

The NFVO triggers the procedure when determining that a PaaS Service is to be scaled out. The source of the trigger could be some internal orchestration and management procedure or an explicit request by an external consumer, such as the OSS/BSS.

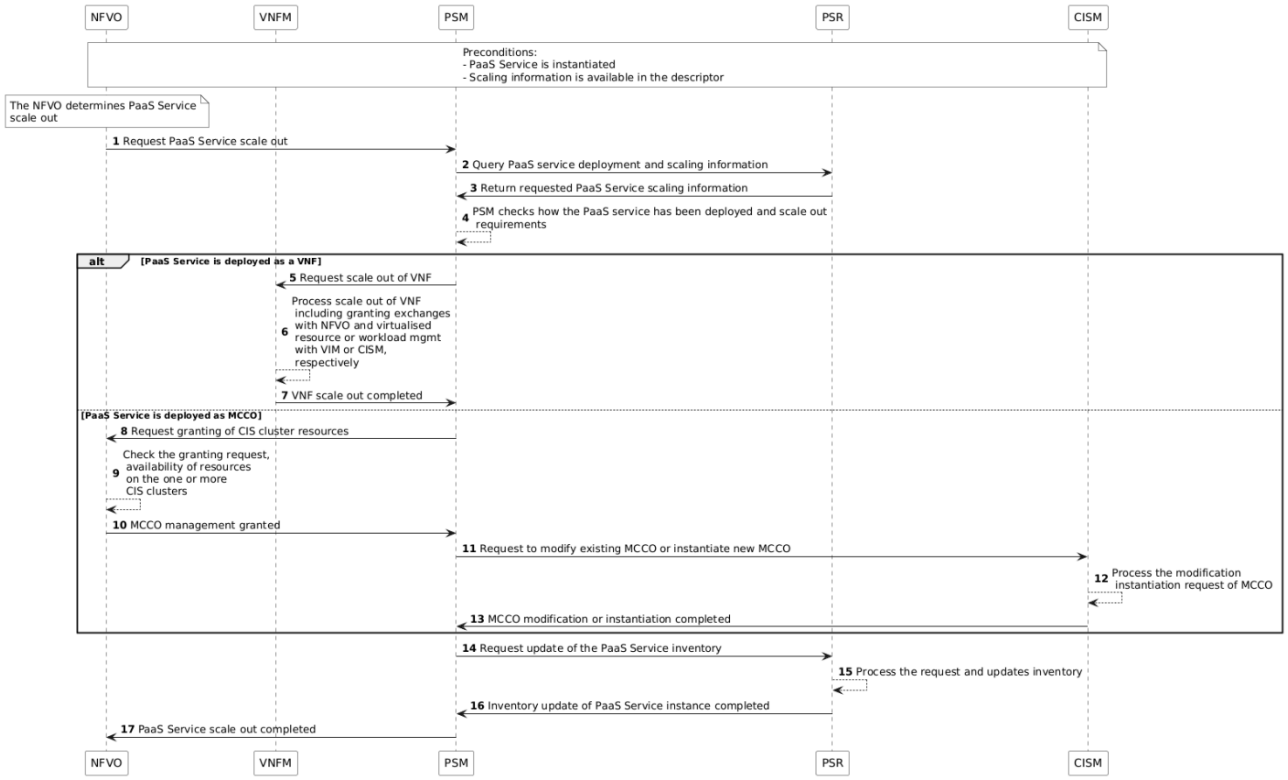


**Figure A.1.9-1: High-level procedure of scale out of a PaaS Service**

The procedure of scale out of a PaaS Service comprises the following steps:

1) The NFVO requests the PSM to scale out PaaS Service. The request includes information about the of PaaS Service.

2) The PSM queries the PSR for deployment information and scaling information of PaaS Service to scale out.

3) The PSR returns to the PSM the PaaS Service deployment and scaling information.

4) PSM checks how the PaaS service has been deployed and scale out requirements.

Depending on the form of deployment of the PaaS Service:

5) In case the PaaS Service is to be deployed as a VNF, the PSM requests the VNFM to scale out the corresponding VNF.

6) The VNFM processes the scale out request. The VNFM interacts with the NFVO for the necessary granting exchanges, and if the granting is successful, then with the VIM or CISM for the deployment of the virtualised resources or workloads, respectively.

7) Once the scale out of the VNF completes, the VNFM confirms to the PSM that the VNF scaling realizing the PaaS Service is complete.

8) In case the PaaS Service is to be deployed as a set of one or more MCCO, first the PSM requests granting of the operation and resource to the NFVO.

9) The NFVO process the granting request, checks the availability of resources on the CIS cluster(s) and determines the placement requirements.

10) The NFVO confirms to the PSM the granting of CIS cluster resources.

11) The PSM requests to the CISM the modification of existing MCCOs or instantiation of new MCCOs.

12) The CISM processes the modification/instantiation request. The CISM allocates the resources in the CIS cluster and the modification/creation of the corresponding MCCO.

13) The CISM confirms to the PSM the modification/instantiation of the MCCO whose workloads realize the PaaS Service.

14) The PSM requests the PSR to update PaaS Service inventory.

15) The PSR update the PaaS Service inventory.

16) The PSR confirms to the PSM the update of the PaaS Service inventory information.

17) The PSM confirms to the NFVO the scale out of the PaaS Service requested.

# Annex B (informative): Relationship of ETSI NFV PaaS Services and other Management Systems

## B.1 Interactions between ETSI NFV PaaS Services, NFV-MANO and the 3GPP Service Based Management Architecture (SBMA)

### B.1.1 Description

The Service Based Management Architecture (SBMA) is defined by 3GPP in ETSI TS 128 533 [i.19]. Like in the case of NFV-MANO, VNF generic OAM functions and other PaaS Services reside outside the SBMA management domain.

Any PaaS Service exposes a well-defined IF-F1 interface, specified in clause 6.3 of the present document, to its consumers., Following the SBMA approach, a service consumer residing in SBMA (i.e. inside a MnF) can interact with a PaaS Service (acting as a service producer). The functionality (fully or partially provided by the PaaS Service) can therefore be consumed by SBMA through diverse MnS (management services) procedures.

NOTE: Whether or not a PaaS Service consumed by the MnF consumer is translated to a new or mapped to a specific MnS (supported, for example, by a MnF) is out of scope of the present document.

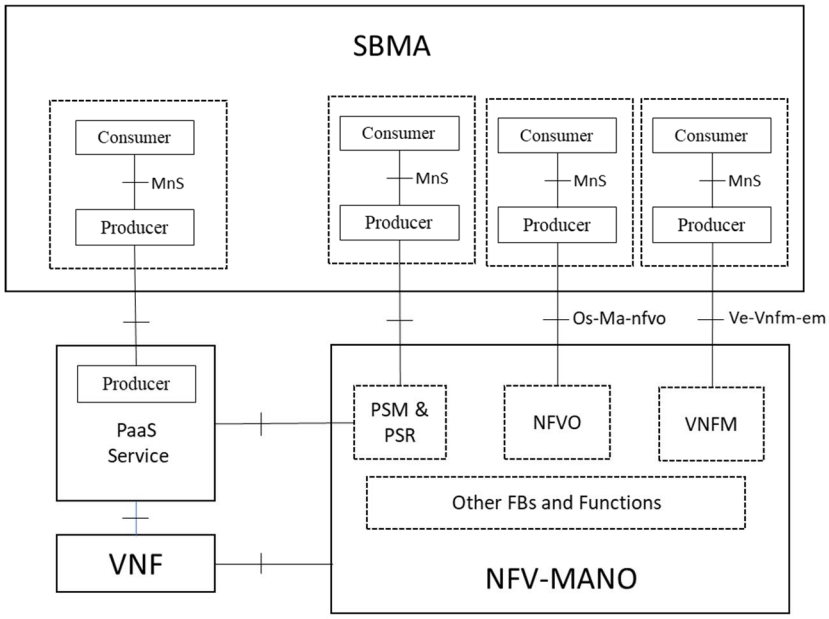


**Figure B.1.1-1 Interactions between ETSI NFV PaaS Services, NFV-MANO and the 3GPP Service Based Management Architecture (SBMA)**

Regarding interactions between NFV-MANO and SBMA, SBMA can interact with NFV-MANO with the following entities:

a) NFVO: through the Os-Ma-nfvo reference point defined in ETSI GS NFV-IFA 013 [4].

b) VNFM: through the Ve-Vnfm-em reference point defined in ETSI GS NFV-IFA 008 [i.5].

c) PSM/PSR: although PaaS Services reside outside the NFV-MANO domain, irrelevant of the deployment form (PaaS Service as a VNF, PaaS Service as a new object class, etc.), PSM/PSR are responsible for managing the PaaS Services. An SBMA service consumer can invoke the PaaS Service management operations exposed by the PSM/PSR interfaces described in clause 10 of the present document.

# Annex C (informative): VNF configuration options

## C.1 Possible interplay between PaaS Services and Methods described in ETSI GR NFV-EVE 022

### C.1.1 Overview

The goal of this annex is to summarize the different options the NFV framework supports for configuring a VNF.

As different VNF aspects can be configured (virtualisation dependent and virtualisation independent) more than one method can be exploited.

In summary for configuring a VNF the following options can be considered:

- VNF configuration using one or combination of three methods (i.e. method #A, method #B and method #C) described in ETSI GR NFV-EVE 022 [i.4];
- VNF configuration using one or more of VNF Generic OAM functions;
- VNF configuration using one or more of VNF Generic OAM functions in combination with the methods described in ETSI GR NFV-EVE 022 [i.4].

### C.1.2 Methods described in ETSI GR NFV-EVE 022

ETSI GR NFV-EVE 022 [i.4], clause 4.3 describes three possible VNF configuration methods enabled by the NFV architectural framework:

**Method#A:** A VNF instance receives configuration data directly from the OSS/BSS or from its EM. In the latter case the data can be originated in the EM or received by the EM from the OSS/BSS. The configuration procedure can follow a push (configuration data pushed to the VNF by the OSS/BSS or EM) or pull model (configuration data retrieved by the VNF instance from the OSS/BSS or EM).

**Method#B:** A VNF instance receives configuration data from the VNFM in charge of managing its lifecycle.

**Method#C:** A VNF instance receives configuration data via the NFV infrastructure. This configuration method is intended to enable Day-0 configuration. The actual procedure depends on the selected VIM or CISM solution.

### C.1.3 Interplay with VNF Generic OAM Functions and other PaaS Services

**Method#A:** configuration data directly from the OSS/BSS or from its EM.

Based on the VNF configuration method#A described in ETSI GR NFV-EVE 022 [i.4], a possible variation when considering PaaS Services is depicted in figure C.1.3-1.

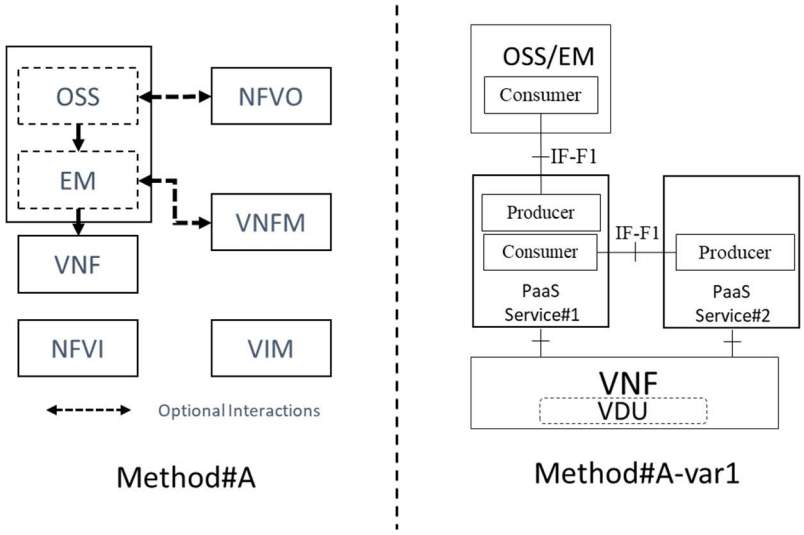


**Figure C.1.3-1: VNF configuration through OSS**

Method#A-var-1: According to this variation of method #A, OSS conveys VNF configuration information (or makes a request for a VNF configuration management action) to one or more PaaS Services. Then these PaaS Services convey configuration information to the VNF or performs the necessary VNF configuration management action. PaaS Services can also interact with each other (for example with the Configuration Server) according to the basic principles of operation described in clause 4.2 of the present document for the purpose of configuring the VNF.

**Method#B:** Configuration data from VNFM.

Two different variations depicted in figure C.1.3-2 can be considered for the case of configuring a VNF through VNFM based on method #B described in ETSI GR NFV-EVE 022 [i.4].

Method#B_var-1: Configuration data are originated from the VNFM or received by the VNFM from the VIM, the EM or the NFVO. Then the configuration data are conveyed to a PaaS Service which performs the necessary VNF configuration management action.

Method#B_var-2: Configuration data are forwarded to one or more PaaS Services from OSS/EMs. Then configuration data are forwarded to VNFM, which is responsible to configure the VNF using existing mechanisms (i.e. based on ETSI GS NFV-IFA 008 [i.5]).

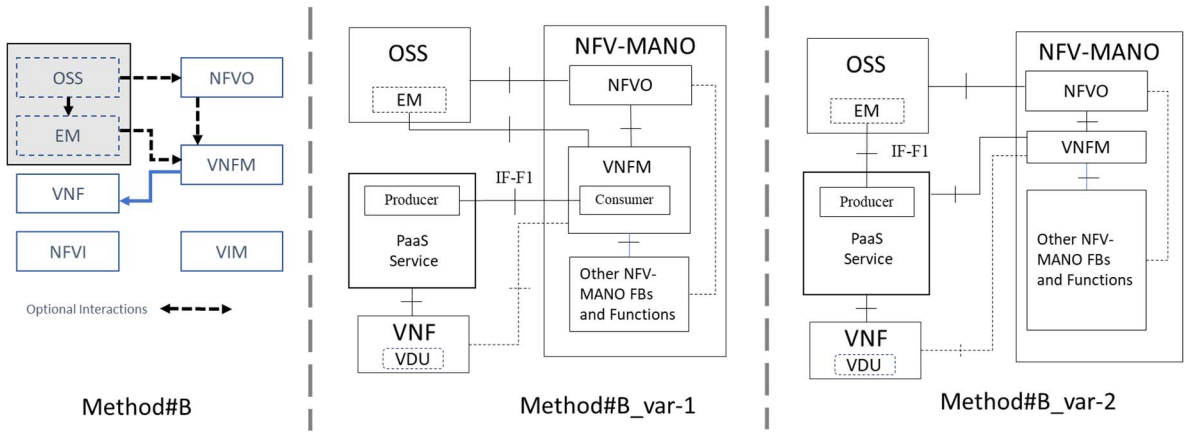


**Figure C.1.3-2: VNF configuration through VNFM**

**Method#C:** Configuration data through other FBs or Functions.

According to method#C, VNF instance receives configuration data via the NFVI as instructed by NFV-MANO entities like VIM or CISM depending on the type of the virtualisation container. This method can facilitate Day-0 VNF configuration.

Two different variations can be considered for the case of configuring a VNF based on this reasoning, depicted in figure C.1.3-3.

Method#C_var-1: Configuration data are originated from any possible source (like the VNFM or received by NFVO after getting the data from OSS). Then the configuration data are conveyed to a PaaS Service from VIM or CISM. The PaaS Service performs the necessary VNF configuration management action.

Method#C_var-2: Configuration data are forwarded to one or more PaaS Services from OSS/EMs. Then configuration data are forwarded to VIM/CISM or even PIM, which are responsible to configure VNF/VDU aspects though interaction with NFVI, to enable Day-0 configuration.

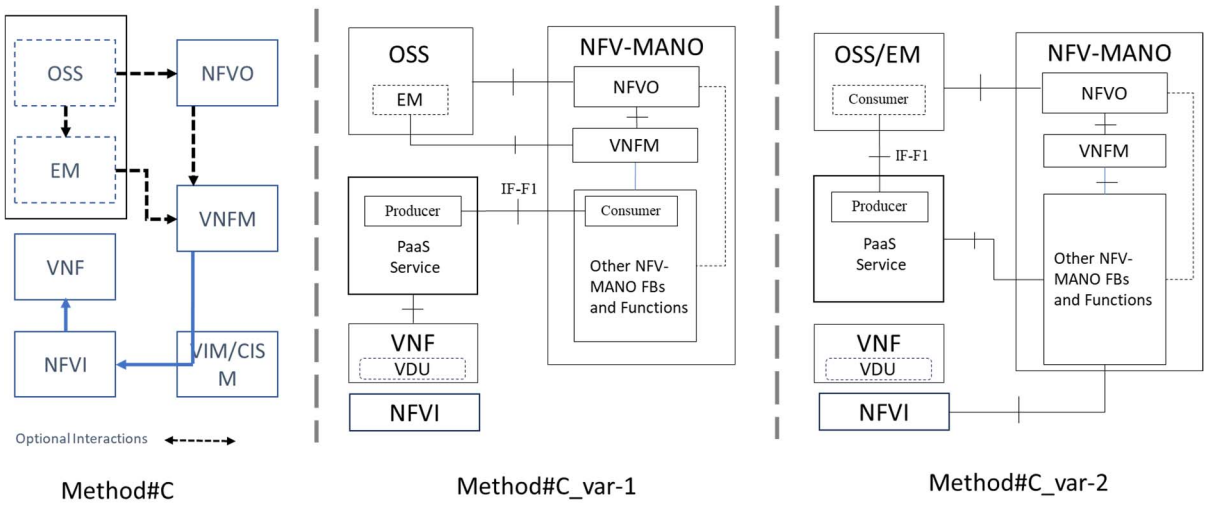


**Figure C.1.3-3: VNF configuration through other FBs or Functions**

# C.2 Summary of VNF Configuration methods

Based on the previous analysis a VNF can be configured using one of the following methods:

| Identifier | Description | Notes |
|---|---|---|
| VNF_Conf_1 | From OSS/EM | Method #A described in ETSI GR NFV-EVE 022 [i.4] |
| VNF_Conf_2 | From PaaS Services as instructed by OSS/EM | Method #A-var-1 described in clause C.1.3 |
| VNF_Conf_3 | From VNFM | Method #B described in ETSI GR NFV-EVE 022 [i.4] |
| VNF_Conf_4 | From PaaS Services as instructed by VNFM | Method #B-var-1 described in clause C.1.3 |
| VNF_Conf_5 | From VNFM as instructed by a PaaS Service | Method #B-var-2 described in clause C.1.3 |
| VNF_Conf_6 | From NFVI | Method #C described in ETSI GR NFV-EVE 022 [i.4] |
| VNF_Conf_7 | From PaaS Services as instructed by VIM/CISM/PIM | Method #C-var-1 described in clause C.1.3 |
| VNF_Conf_8 | From NFVI/VIM/CISM as instructed by a PaaS Service | Method #C-var-2 described in clause C.1.3 |

In all cases PaaS Services like the Configuration Server can be exploited. A pros and cons analysis of the methods above is out of scope of the present document.

# Annex D (informative): Change history

| Date | Version | Information about changes |
|---|---|---|
| June 2022 | V0.0.1 | Skeleton and ToC |
| July 2022 | V0.0.2 | • NFVIFA(22)000419r1_FEAT24_IFA049_Overview of the VNF generic OAM functions framework<br>• NFVIFA(22)000422r3_FEAT24 IFA049 Functional requirements for VNF generic OAM function Traffic enforcer function<br>• NFVIFA(22)000435_FEAT24 IFA049 Functional requirements for VNF generic OAM function Network configuration manager<br>• NFVIFA(22)000445r2_FEAT24_IFA049_Functional_requirements_for_VNF_generic_OAM_Metric_analyser<br>• NFVIFA(22)000443r3_FEAT24_IFA049_Functional_requirements_for_VNF_generic_OAM_Log_analyser<br>• NFVIFA(22)000441r3_FEAT24_IFA049_Functional_requirements_for_VNF_generic_OAM_Log_aggregator<br>• NFVIFA(22)000444r3_FEAT24_IFA049_Functional_requirements_for_VNF_generic_OAM_Metric_aggregator<br>• NFVIFA(22)000475r2_FEAT24 IFA049 Functional requirements for VNF generic OAM function Upgrade VNF<br>• NFVIFA(22)000493r1_FEAT24 IFA049 requirements for the generic OAM framework,<br>• NFVIFA(22)000518_FEAT24_IFA049_generic_OAM_framework_considerations,<br>• NFVIFA(22)000540r2_ FEAT24 IFA049 Functional requirements for VNF generic OAM function Traffic enforcer function |
| September 2022 | V0.0.3 | • NFVIFA(22)000541r2_ FEAT24 IFA049 Interface requirements for VNF generic OAM function Traffic enforcer function<br>• NFVIFA(22)000542r2_FEAT24 IFA049 Interface requirements for VNF generic OAM function VNF metrics aggregator function<br>• NFVIFA(22)000562_ FEAT24 IFA049 Interface requirements for VNF generic OAM function VNF metrics analyser<br>• NFVIFA(22)000567r1_ FEAT24 IFA049 Interface requirements for VNF generic OAM function Network configuration manager<br>• NFVIFA(22)000591r2_ FEAT24 IFA049 Interface requirements for VNF generic OAM function Upgrade VNF<br>• NFVIFA(22)000593r1_ FEAT24 IFA049 Interface requirements for VNF generic OAM function VNF log analyser<br>• NFVIFA(22)000594_ FEAT24 IFA049 Interface requirements for VNF generic OAM function VNF Log aggregator function |

| Date | Version | Information about changes |
|---|---|---|
| December 2022 | V0.0.4 | • NFVIFA(22)000755r1 FEAT24 IFA049 VNF generic OAM functions framework<br>• NFVIFA(22)000756r2 FEAT24 IFA049 Description for interface and service level interactions,<br>• FVIFA(22)000674r5_FEAT24_IFA049_Interface_requirement_VNF_generic_OAM_Metric_A,<br>• FVIFA(22)000673r5FEAT24_IFA049_Interace_requirement_VNF_generic_OAM_Metric_Analyser_Clause6.y,<br>• FVIFA(22)000713r2_FEAT24_IFA049_Interace_requirement_VNF_generic_OAM_Log_Analy,<br>• FVIFA(22)000714r3_FEAT24_IFA049_Interface_requirement_VNF_generic_OAM_Log_Aggr,<br>• NFVIFA(22)000812 FEAT24 IFA049 Remove policy management requirements on Traffic management function and Traffic management function interface<br>• NFVIFA(22)000813r4 FEAT24 IFA049 Operation requirements on interface VNF generic OAM function Traffic Management Interface<br>• NFVIFA(22)000814r3 FEAT24 IFA049 Operation requirements on interface VNF generic OAM function Network Configuration Management Interface<br>• NFVIFA(22)000815r4 FEAT24 IFA049 Operation requirements on interface VNF generic OAM function VNF Upgrade Management Interface<br>• NFVIFA(22)000859r1 FEAT24 IFA049 Functional requirements for the Time Function<br>• NFVIFA(22)000860r2 FEAT24 IFA049 Functional requirements for the VNF configuration manager function<br>• NFVIFA(22)000861r1 FEAT24 IFA049 Interface requirements for the Time Function<br>• NFVIFA(22)000862r1 FEAT24 IFA049 Interface requirements for the VNF Configuration Manager Function<br>• NFVIFA(22)000864r3 FEAT24 IFA049 Operation requirements on interface VNF generic OAM function Metrics Analysis, Metrics Aggregator, Log Analyser and Log Aggregator Exposure Interface<br>• NFVIFA(22)000872r1 FEAT24 IFA049 VNF service interactions requirements<br>• NFVIFA(22)000909r1_FEAT24_IFA049_Information_elements_related_to_Network_Config |
| February 2023 | V0.0.5 | • NFVIFA(22)000910r3_FEAT24_IFA049_Information_elementsrelated_to_VNF_Upgrade_Man<br>• NFVIFA(22)000889r4_FEAT24_IFA049_Information_element_exchange_Metrics_aggregato<br>• NFVIFA(22)000890r3 FEAT24 IFA049 Information elements exchange for VNF generic OAM function Metrics Analysis<br>• NFVIFA(22)000981r2 FEAT24 IFA049 Information elements exchange for VNF generic OAM function Metrics Analysis |
| March 2023 | V0.0.6 | • NFVIFA(22)000072r1 FEAT24 IFA049 Information elements exchange for VNF generic OAM function Log Analysis<br>• NFVIFA(22)000073r1 FEAT24 IFA049 Information elements exchange for VNF generic OAM function Log Aggregator<br>• NFVIFA(22)000938r3 FEAT24 IFA049 VNF generic functions descriptors<br>• NFVIFA(23)000107 FEAT24 IFA049 Editorial cleanup clause 4.2<br>• NFVIFA(23)000123 FEAT24 IFA049 Interface operations for the VNF generic OAM function VNF Configuration Manager<br>• NFVIFA(23)000124r2 FEAT24 IFA049 Interface operations for the VNF generic OAM Time function<br>• NFVIFA(23)000138r1 FEAT24 IFA049 Information elements related to the Editor notes in clauses 5, 6 and 7 |

| Date | Version | Information about changes |
|---|---|---|
| April 2023 | V0.0.7 | • NFVIFA(23)000139r1_FEAT24_IFA049_remove_G-AM_analysis_related_clauses_<br>• NFVIFA(23)000175r1_FEAT24_IFA049_VNF_generic_functions_descriptors-clasue_9_3<br>• NFVIFA(23)000179_FEAT24_IFA049_VNF_generic_OAM_Editor_Notes_Clause6_3_2_2_2<br>• NFVIFA(23)000180_FEAT24_IFA049__address_EN_for_clause_6_2_3_and_clause_6_3_3_<br>• NFVIFA(23)000182_FEAT24_IFA049_Remove_Annex_A__Example_realizations<br>• NFVIFA(23)000188_FEAT24_IFA049_VNF_generic_functions_ENs_handling-part_1_2<br>• NFVIFA(23)000195r2_FEAT24_IFA049_VNF_generic_OAM_Editor_Notes_Clause6_2_4<br>• NFVIFA(23)000238_FEAT24_IFA049_VNF_generic_OAM_Editor_Notes_Clause6_3_3_2_2 |
| May 2023 | V0.1.0 | • NFVIFA(23)000268r1 FEAT24 IFA049 1 scope and 3 Definition<br>• NFVIFA(23)000296r1 FEAT24 IFA049 VNF generic functions handling ENs (clauses 6.3.8 and 9.1)<br>• NFVIFA(23)000305r1 FEAT24 IFA049 handling ENs related to MDAF<br>• NFVIFA(23)000307 FEAT24 IFA049 Information elements related to the Editor notes in clause 6.2.6 |
| June 2023 | V0.3.0 | • NFVIFA(23)000331r1 FEAT24 IFA049 handling ENs related to notifications management<br>• NFVIFA(23)000475r1 FEAT24 IFA049-final review-editorial corrections<br>• NFVIFA(23)000476r1 FEAT24 IFA049-final review-technical corrections<br>• NFVIFA(23)000486r1 FEAT24 IFA049 final review comments |

| Date | Version | Information about changes |
|---|---|---|
| December 2023 | V5.0.1 | • NFVIFA(23)000667r1_FEAT21 IFA049 Scope and skeleton extension for release 5 version to include PaaS Services management<br>• NFVIFA(23)000760r1_FEAT24 IFA049ed511 Update structure for release 5<br>• NFVIFA(23)000732r3_FEAT24_IFA049_Functional_requirements_for_Policy_agent<br>• NFVIFA(23)000751_FEAT21 IFA049ed511 Clause 4 Updates for PaaS<br>• NFVIFA(23)000756_FEAT21 IFA049ed511 Clause 8.1 Updated service interactions for PaaS<br>• NFVIFA(23)000759r1_FEAT21 IFA049ed511 Clause 10 PSM and PSR interface requirements<br>• NFVIFA(23)000761r2_FEAT24 IFA049ed511 functional requirements for the VNF testing manager<br>• NFVIFA(23)000762r1_FEAT24 IFA049ed511 interface requirements for the VNF testing manager<br>• NFVIFA(23)000763r1_FEAT24 IFA049ed511 interface operations and IEs for the VNF testing manager<br>• NFVIFA(23)000764_FEAT24 IFA049ed511 supporting automation with the Policy agent<br>• NFVIFA(23)000765_FEAT24 IFA049ed511 application aspects<br>• NFVIFA(23)000766_FEAT24 IFA049ed511 Framework updates<br>• NFVIFA(23)000777_FEAT30 IFA049ed511 Multiple clauses Adding Configuration Server PaaS Service<br>• NFVIFA23)000778_FEAT30 IFA049ed511 Clause 6.3 Configuration Server interfaces<br>• NFVIFA(23)000780_FEAT30 IFA049ed511 Clause 5 Requirements to use the Configuration Server by other functions<br>• NFVIFA(23)000789r1_FEAT24_IFA049_Functional_requirements_for_Notification_manager<br>• NFVIFA(23)000809r1_FEAT24_IFA049_Interface_Requirement_For_Notification_manager<br>• NFVIFA(23)000810r1_FEAT24_IFA049_Interface_Requirement_For_Policy_Agent |
| February 2024 | V5.0.2 | • NFVIFA(24)000035_FEAT30 IFA049ed511 Clause 7 Information elements Configuration Server<br>• NFVIFA(24)000042r2_FEAT21 IFA049ed511 Clause 10.3 PaaS Services LCM interface operations<br>• NFVIFA(24)000043r1_FEAT21 IFA049ed511 Clause 10.3 PSD management interface operations<br>• NFVIFA(24)000044r1_FEAT21 IFA049ed511 Clause 10.3 PaaS Services Registration management interface operations<br>• NFVIFA(24)000088_FEAT24 IFA049 relationship between MDAF and VNF generic OAM functions<br>• NFVIFA(24)000092r1_FEAT21 IFA049ed511 Clause 9.4 PSD general requirements<br>• NFVIFA(24)000093r1_FEAT24 IFA049 Interactions between the Notification manager and generic OAM functions<br>• NFVIFA(24)000094_FEAT21 IFA049ed511 Clause 9.5 Adding PSD modelling specification<br>• NFVIFA(24)000095r1_FEAT21 IFA049ed511 Clause 9.1 and 9.2 PaaS generalization of descriptors<br>• NFVIFA(24)000104r1_FEAT24_IFA049_Notification_manager_Interface_Operation_requirement |

| Date | Version | Information about changes |
|---|---|---|
| March 2024 | V5.0.3 | • NFVIFA(24)000090_FEAT24_IFA049_operations_and_IEs_for_the_VNF_testing_manager<br>• NFVIFA(24)000091_FEAT24_IFA049_operations_and_IEs_for_the_policy_agent<br>• NFVIFA(24)000129_FEAT21_IFA049ed511_Clause_9_2_Address_EN_PaaS_descriptors<br>• NFVIFA(24)000134_IFA049ed511_Editorial_cleanup<br>• NFVIFA(24)000135_FEAT21_IFA049ed511_Clause_1_Update_scope_with_PaaS<br>• NFVIFA(24)000136_FEAT21_IFA049ed511_Address_EN_PSD_flavour<br>• NFVIFA(24)000139_FEAT21_IFA049ed511_Clause_10_3_Address_EN_Create_PaaS_Service<br>• NFVIFA(24)000148r1_FEAT24_IFA049_Network_configuration_manager_with_service_mesh<br>• NFVIFA(24)000168r1_FEAT24_IFA049ed511_interactions_between_the_Policy_agent and other entities<br>• NFVIFA(24)000169_FEAT24_IFA049ed511__Policy_agent_interactions with the policy-managed objects<br>• NFVIFA(24)000171_FEAT24_IFA049ed511_Policy_agent_notifications<br>• NFVIFA(24)000173r1_FEAT21_IFA049ed511_PaaS_Service_resources_information_notifications<br>• NFVIFA(24)000174_FEAT21_IFA049ed511_Service_interaction_requirements_on_SB-V<br>• NFVIFA(24)000175_FEAT21_IFA049ed511_specification_of_notifications_and_subscriptions<br>• NFVIFA(24)000176_FEAT24_IFA049ed511_Editorial_cleanup_removal_of_unnecessary_ENs<br>• NFVIFA(24)000197_FEAT21_Editorial_cleanup_update_clause_numbering_for_policy_agent |
| May 2024 | V5.0.4 | • NFVIFA(24)000170_FEAT24_IFA049ed511_Address_EN_on_policy_enforcement_handling<br>• NFVIFA(24)000172_FEAT24_IFA049ed511_Address_EN_on_notifications_specification<br>• NFVIFA(24)000261_IFA049ed511_Multiple_clauses_Editorial_changes<br>• NFVIFA(24)000274_FEAT24_IFA049ed511_description_of_the_table_for_input_parame<br>• NFVIFA(24)000286_IFA049ed511_Multiple_clauses_small_technical_alignments<br>• NFVIFA(24)000288_IFA049ed511_Making_some_functions_generic_PaaS_Services |
| June 2024 | V5.1.2 | Initial ed521 draft version created from published version 5.1.1 |
| August 2024 | V5.1.3 | • NFVIFA(24)000391_IFA049ed521_Clauses_ordering_issue<br>• NFVIFA(24)000392r1_IFA049ed521_Policy_execution_functional_requirements<br>• NFVIFA(24)000395r1_IFA049ed521_requirements_for_network_monitoring<br>• NFVIFA(24)000405_FEAT24_IFA049_PaaSService_Instantiate_use_case<br>• NFVIFA(24)000406r4_FEAT24_IFA049_PaaSService_Query_PSD_Information<br>• NFVIFA(24)000411_IFA049ed521_Updating_the_description_for_EMs<br>• NFVIFA(24)000414_IFA049ed521_Updating_the_scope_of_the_VNF_configuration_mana<br>• NFVIFA(24)000421_IFA049ed521_functional_requirements_for_notifications<br>• NFVIFA(24)000424r1_FEAT24_IFA049_PaaSService_create_PSD<br>• NFVIFA(24)000425r1_FEAT24_IFA049_PaaSService_upload_PSD<br>• NFVIFA(24)000434r2_FEAT24_IFA049_PaaSService_delete_PSD<br>• NFVIFA(24)000437r1_IFA049ed521_overlapping_framework_design |
| September 2024 | V5.1.4 | • NFVIFA(24)000474r1_IFA049ed521_PaaS_Services_Interactions_with_VNFs<br>• NFVIFA(24)000475r3_IFA049ed521_interactions_with_SBMA<br>• NFVIFA(24)000476r1_IFA049ed521_relationship_with_FEAT30_for_config<br>• NFVIFA(24)000477_IFA049ed521_testing_management_operations<br>• NFVIFA(24)000478r1_IFA049ed521 PaaS Services policy enforcement<br>• NFVIFA(24)000491_IFA049ed521 Adding a list of VNF generic OAM functions in clause 4<br>• NFVIFA(24)000514r2_IFA049ed521_PaaS_Service Termination use case<br>• NFVIFA(24)000516r2_FEAT24_IFA049_PaaSService subscription for notification<br>• Rapporteur actions: updated the pointer to C.1.3 instead of C.1.2 in cause C.1.4. Updated clause number C.1.4 to C2 |

| Date | Version | Information about changes |
|---|---|---|
| October 2024 | V5.1.5 | • NFVIFA(24)000573r1_FEAT24_IFA049_PaaSService_Use_Case_sequence_diagram_update<br>• NFVIFA(24)000602r1_FEAT24_IFA049_PaaSService_Scale_out_use_case<br>• NFVIFA(24)000613r1_FEAT24_IFA049__PaaSService_Instantiate_use_case_update |
| October 2024 | V5.1.6 | • Rapporteur action fixed wrong autonumbering in clause A.1.9 in the annex |
| March 2025 | V5.2.2 | • Initial ed531 draft version created from published version 5.2.1 |
| March 2025 | V5.2.3 | • NFVIFA(25)000057r1_FEAT24_IFA049_Update_For_Log_analysis_input_parameter<br>• NFVIFA(25)000059_FEAT24_IFA049ed531_Management_of_PaaS_Services |
| May 2025 | V5.2.4 | • NFVIFA(25)000104_ IFA049ed531 Updates related to certificate management of PaaS service VNFs<br>• NFVIFA(25)000136_IFA049ed531_Review_contribution_regarding_IFA026_referencing |
| October 2025 | V5.3.2 | Baseline for drop 541 |
| November 2025 | V5.3.3 | • NFVIFA(25)000272_FEAT24_IFA049ed541_metrics_aggregator<br>• NFVIFA(25)000263r2_FEAT24 IFA049ed541VNF Testing Manager updates<br>• NFVIFA(25)000316r1_FEAT24 IFA049ed541VNF Testing Manager domain areas.<br>• Rapporteur action added "Operator B" in figure 4.2.1.8-1 in right hand side OSS (fixed copy error) |

# History

| Version | Date | Status |
|---|---|---|
| V5.1.1 | June 2024 | Publication |
| V5.2.1 | December 2024 | Publication |
| V5.3.1 | September 2025 | Publication |
| V5.4.1 | January 2026 | Publication |
| | | |