ETSI Plugtests Report V1.0.0 (2019-07)

4th ETSI NFV Plugtests
Sophia Antipolis, France
3rd – 7th June 2019

ETSI

Keywords
Testing, Interoperability, NFV, MANO, VNF, VIM

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 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-préfecture de Grasse (06) N° 7803/88

Important notice

Copyright Notification

Contents

Executive summary

The 4th NFV Plugtests was organised by the ETSI Centre for Testing and Interoperability in the context of the NFV Plugtests Programme, and hosted by ETSI in Sophia Antipolis, France from 3rd to 7th June 2019.

The 4th NFV Plugtests offered NFV and MEC solution providers and open source projects an opportunity to meet and assess the level of interoperability of their NFV and Edge solutions, while validating their implementation of NFV and MEC specifications and APIs.

40 organisations and over 200 engineers were involved (on site and/or remotely) in the preparation of this busy event forming an engaged and diverse community implementers of 45 NFV and MEC solutions such as:

- Virtual Network Functions (VNFs), combined in different Network Services (NS)
- Management and Orchestration solutions: NFV Orchestrators, VNFMs
- NFV Platforms: NFV Infrastructure and Virtual Infrastructure Managers
- MEC Platforms and Applications
- Simulators, test and automation tools implementing NFV and/or MEC APIs.

Different participating organisation and Function Under Test were able to interact remotely through the NFV HIVE (Hub for Interoperability and Validation at ETSI) which provides a secure framework to interconnect participants' labs and implementations and is a key element for an efficient Plugtests preparation and successful events.

The main highlights of this 4th NFV Plugtests were:

- Stable results in NFV Interoperability Test Sessions in general, targeting both single and multi-vendor networks services and a wide range of operations from on-boarding and instantiation, manual and automatic scaling from different triggers, NS updates, fault and performance management and termination. As a high point, for the first time in a Plugtests, some Test Sessions included VNFMs and NFVOs from different providers, see 8.1.4. This new type of Session showed very positive results for basic operations such as on-boarding, instantiations, manual scaling and terminations. This testing triggered interesting discussions during the Plugtests and lead to important findings in several NFV Specifications, see 10.1.1.1
- Significant progress in NFV API Conformance Testing coverage and results, building on the success and learnings of the Remote NFV API Plugtests run only few months before the 4th NFV Plugtests. The 1-week event allowed to run more Test Sessions and obtain higher execution and success rates than the 2 months and a half of remote event. As it can be seen in 9.2, the most significant improvements were seen on NFV SOL005 API implementations, possibly thanks to the early availability this time of the associated Test Specifications. The 4th NFV Plugtests allowed to identify and fix 25 new issues on the NFV API Conformance Robot Test Suites, see 10.3.

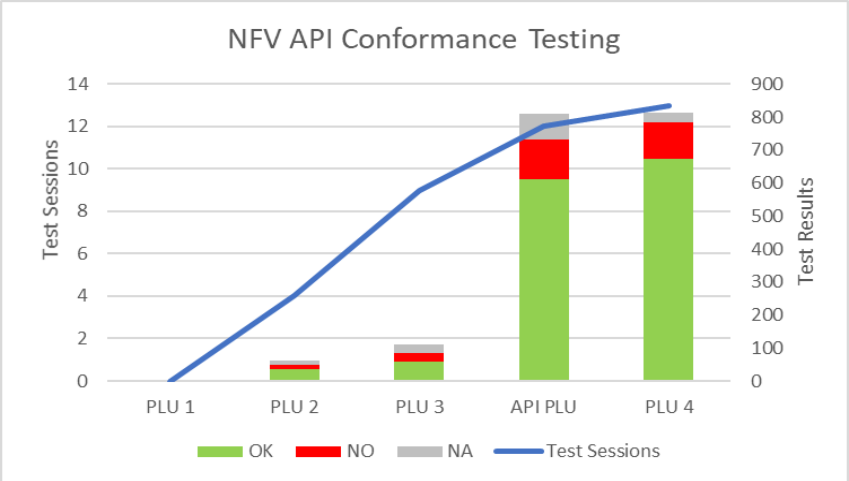

Figure 1. Progress of NFV API Conformance Testing

- First MEC Interoperability Test Sessions. The experimental MEC Track consisted of 3 different types of testing covering routing, application lifecycle and MEC API. A total of 8 participating organisations took part in 6 optional Test Sessions and a lot of interest was shown by the whole participating community. The MEC Track was an excellent exercise to promote and create awareness of MEC APIs, and MEC-in-NFV Specifications, which will probably allow to grow the number of participating organisations in future events.

Once again, the Plugtests allowed to validate and identify a significant number of issues, bugs and inconsistencies in NFV and MEC Specifications, Open APIs and Test Plans, which are compiled in section 10 and will help to increase the quality and consistency of future releases.

The following sections describe in detail the preparation of the 4th NFV Plugtests, the participating implementations, the test plans and testing procedures, the overall results, as well as the lessons learnt, and the feedback collected during the event.

The Test Plans and the present Report are fed back to ETSI NFV and MEC Industry Specification Groups.

1 Introduction

This 4th NFV Plugtests aimed at verifying interoperability and API Conformance across different implementations of the main components of the NFV and MEC Architectural Frameworks, including:

- Virtual Network Functions (VNF), eventually providing additional Element Manager (EM) functionality, combined in different Network Services (NS)
- Management and Orchestration (MANO) solutions, providing integrated NFV Orchestrator (NFVO) and VNF Management (VNFM) functionalities
- Standalone VNFMs and NFVOs
- NFV Platforms, including hardware, providing NFV infrastructure (NFVI) and Virtual Infrastructure Manager (VIM) functionalities
- MEC Platforms and Applications

In addition, Test and support VNFs were used to simulate certain components, validate the correct behaviour of the Network Services (NS) and/or automate part of the Interoperability Test Plan

Remote integration and pre-testing among participants were key for the preparation and a successful Plugtests. For that purpose, the NFV Plugtests HIVE: Hub for Interoperability and Validation at ETSI. a dedicated VPN based network was used to interconnect local and remote implementations in a reliable and secure way.

All the participating implementations, Functions Under Test or Test/support Functions were connected and/or accessible through the HIVE network: most of the NFV platforms and MANO solutions run remotely on participants' labs, a subset of them were deployed locally at ETSI. VNF and NS Packages and images were made available in a local repository hosted at ETSI and uploaded to the different NFV platforms during the pre-testing phase. Some test and support VNFs were deployed locally on local infrastructure at ETSI, also accessible through HIVE. The Plugtests room itself provided direct connectivity to the HIVE network.

All the information required to organise, coordinate and manage the 4th NFV Plugtests was compiled and shared with participants in a dedicated private WIKI. Part of the information presented in this document has been extracted from there: https://wiki.plugtests.net/NFV-PLUGTESTS (login required).

.

2 References

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

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

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

[NFV002] ETSI GS NFV 002: "Network Functions Virtualisation (NFV); Architectural Framework".

[NFV003] ETSI GS NFV 003: "Network Functions Virtualisation (NFV); Terminology for main concepts in NFV".

[IFA005] ETSI GS NFV-IFA 005: "Network Functions Virtualisation (NFV); Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification".

[IFA006] ETSI GS NFV-IFA 006: "Network Functions Virtualisation (NFV); Management and Orchestration; Vi-Vnfm reference point - Interface and Information Model Specification".

[IFA007] ETSI GS NFV-IFA 007: "Network Functions Virtualisation (NFV); Management and Orchestration; Or-Vnfm reference point - Interface and Information Model Specification".

[IFA008] ETSI GS NFV-IFA 008: "Network Functions Virtualisation (NFV); Management and Orchestration; Ve-Vnfm reference point - Interface and Information Model Specification".

[IFA010] ETSI GS NFV-IFA 010: "Network Functions Virtualisation (NFV); Management and Orchestration; Functional requirements specification".

[IFA013] ETSI GS NFV-IFA 013: "Network Functions Virtualisation (NFV); Management and Orchestration; Os-Ma-Nfvo reference point - Interface and Information Model Specification".

[TST002] ETSI GS NFV-TST 002: “Network Functions Virtualisation (NFV); Testing Methodology; Report on NFV Interoperability Testing Methodology”

[TST007] ETSI GR NFV-TST 007: “Network Functions Virtualisation (NFV); Testing; Guidelines on Interoperability Testing for MANO”

[SOL002] ETSI GS NFV-SOL 002 V2.4.1: “Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Ve-Vnfm Reference Point”

[SOL003] ETSI GS NFV-SOL 003 V2.4.1: “Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Or-Vnfm Reference Point”

[SOL005] ETSI GS NFV-SOL 005 V2.4.1: “Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point”

[MEC003] ETSI GS MEC 003 V2.1.1: “Multi-access Edge Computing (MEC); Framework and Reference Architecture”

[MEC011] ETSI GS MEC 011 V1.1.1: “Mobile Edge Computing (MEC); Mobile Edge Platform Application Enablement”

[MEC017] ETSI GR MEC 017 V1.1.1: “Mobile Edge Computing (MEC); Deployment of Mobile Edge Computing in an NFV environment”

[MEC025] ETSI GS MEC-DEC 025 V2.1.1: “Multi-access Edge Computing (MEC); MEC Testing Framework”

[1NFVPLU-TP] 1st ETSI NFV Plugtests Test Plan: https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/1st_ETSI_NFV_Plugtests_Test_Plan_v1.0.0.pdf

[2NFVPLU-TP] 2nd ETSI NFV Plugtests Test Plan: https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/2nd_ETSI_NFV_Plugtests_Test_Plan_v1.0.0.pdf

[3NFVPLU-TP] 3rd ETSI NFV Plugtests Test Plan: https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/3rd_ETSI_NFV_Plugtests_Test_Plan_v1.0.0.pdf

[4NFVPLU-NFVTP] 4th ETSI NFV Plugtests Test Plan for NFV Interoperability:

https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/4th_ETSI_NFV_Plugtests_Test_Plan_NFV_IOP_v1.0.0.pdf

[4NFVPLU-MECTP] 4th ETSI NFV Plugtests Test Plan for MEC Interoperability:

https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/4th_ETSI_NFV_Plugtests_Test_Plan_MEC_IOP_v1.0.0.pdf

[1NFVPLU-R] 1st ETSI NFV Plugtests Report: https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/1st_ETSI_NFV_Plugtests_Report_v1.0.0.pdf

[2NFVPLU-R] 2nd ETSI NFV Plugtests Report: https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/2nd_ETSI_NFV_Plugtests_Report_v1.0.0.pdf

[3NFVPLU-R] 3rd ETSI NFV Plugtests Report: https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/3rd_ETSI_NFV_Plugtests_Report_v1.0.0.pdf

[RNFVPLU-R] ETSI Remote NFV API Plugtests Report: https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/ETSI_Remote_NFV_API_Plugtests_Report_v1.0.0.pdf

[FORGE] ETSI Forge https://forge.etsi.org

[PYTHON] https://www.python.org/

[PIP] https://pip.pypa.io

[JAVA] https://www.java.com/

[MOCK] http://www.mock-server.com/

[TCP-DUMP] https://www.tcpdump.org/

[WHIRESHARK] https://www.wireshark.org/

[RF] https://robotframework.org/

[JSON-SCH-LIB] https://pypi.org/project/robotframework-jsonschemalibrary/

[DEP-LIB] https://pypi.org/project/robotframework-dependencylibrary/

[JSON-LIB] https://github.com/nottyo/robotframework-jsonlibrary

[MOCK-LIB] https://github.com/etsi-cti-admin/robotframework-mockserver

[ROBOT-TS] Robot Test Suite for NFV API Conformance https://forge.etsi.org/gitlab/nfv/api-tests

[ISSUE-TR] Issue Tracker for the Robot Test Suite for NFV API Conformance https://forge.etsi.org/gitlab/nfv/api-tests/issues

[BUGZILLA] ETSI Forge Bugzilla https://forge.etsi.org/bugzilla/

3 Abbreviations

For the purposes of the present document, the terms and definitions given in [NFV003] and [TST002] apply.

4 Technical and Project Management

4.1 Scope

The 4th NFV Plugtests focused on testing 3 main areas

- Multi-vendor NFV Interoperability
- NFV API Conformance
- Multi-vendor MEC and MEC-in-NFV Interoperability

4.1.1 NFV Interoperability

The main goal of the multi-vendor NFV interoperability test sessions was to validate ETSI NFV end-to-end capabilities such as onboarding, instantiation, manual and automatic scaling, updates, fault and performance management and termination.

During these sessions, the Systems Under Test (SUTs) were made of different combinations of the following Functions Under Test (FUTs):

- One or several NFV Platforms, including hardware and providing pre-integrated VIM and NFVI functionality
- One MANO solution, providing pre-integrated NFVO and generic VNFM functionality
- One Network Service, composed of VNFs from one or different providers, eventually including EM and specific VNFM management functionality.

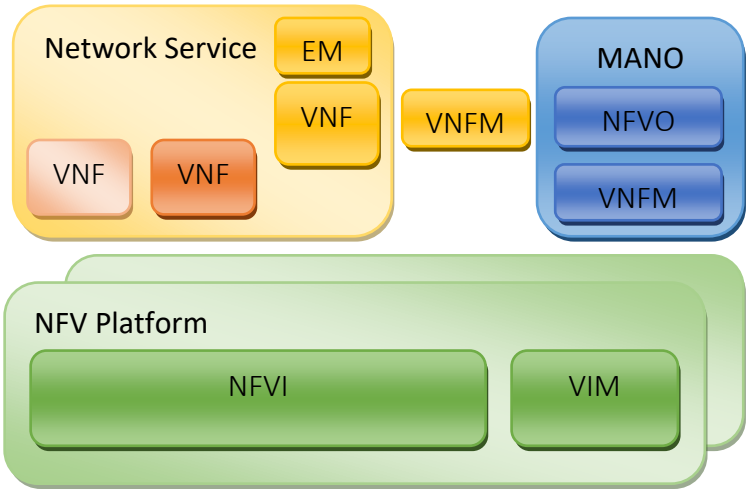

Figure 2. Interoperability System Under Test

4.1.2 NFV API Conformance

The main goal of the NFV API Conformance test sessions was to run individual Test Sessions between participants and experts in the Plugtests team allowing to:

- validate the Robot Test Suites (run by Plugtests experts) for the NFV API Conformance Test Specification: NFV TST010
- asses the level of conformance of participating Functions Under Test of VNFs, VNFMs, and NFVOs (operated by participants) with NFV SOL002, SOL003, and SOL005 APIs and OpenAPIs

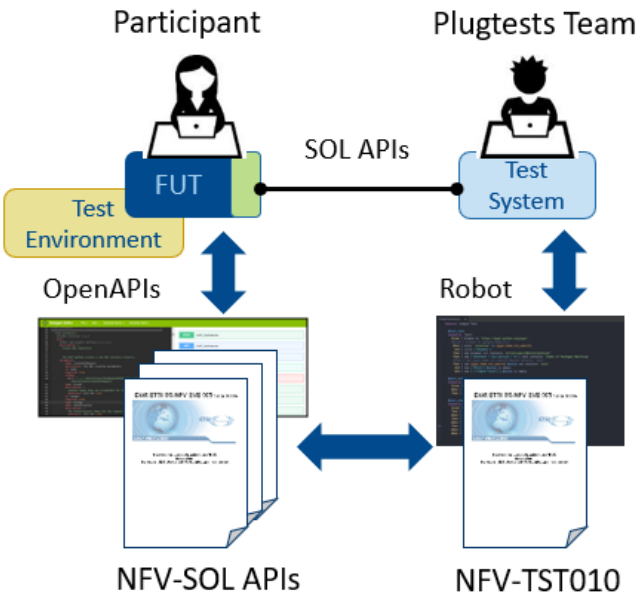

Figure 3. NFV API Conformance testing

4.1.3 MEC Interoperability

A new optional type of test sessions was added to this fourth edition of the NFV Plugtests, as a first experimental "MEC Interoperability track". The MEC Track consisted of 3 types of interoperability tests covering routing, application lifecycle and APIs for Multi-access Edge Computing solutions. The main goal of the MEC Track was to run multi-vendor Test Sessions among participants with the support of experts in the Plugtests Team allowing them to test interoperability between systems and learn more about these emerging standards. The figure below shows in yellow the components involved in the MEC Track.

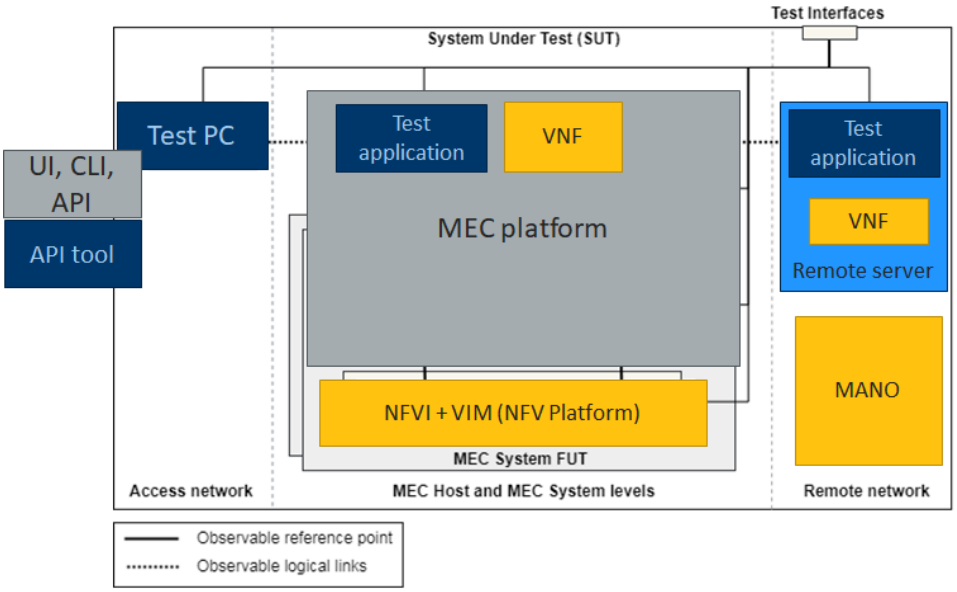

Figure 4. MEC Interoperability Testing

4.2 Timeline

The 4th NFV Plugtests preparation started in parallel with the Remote NFV API Plugtests and run through different phases as described in the figure below.

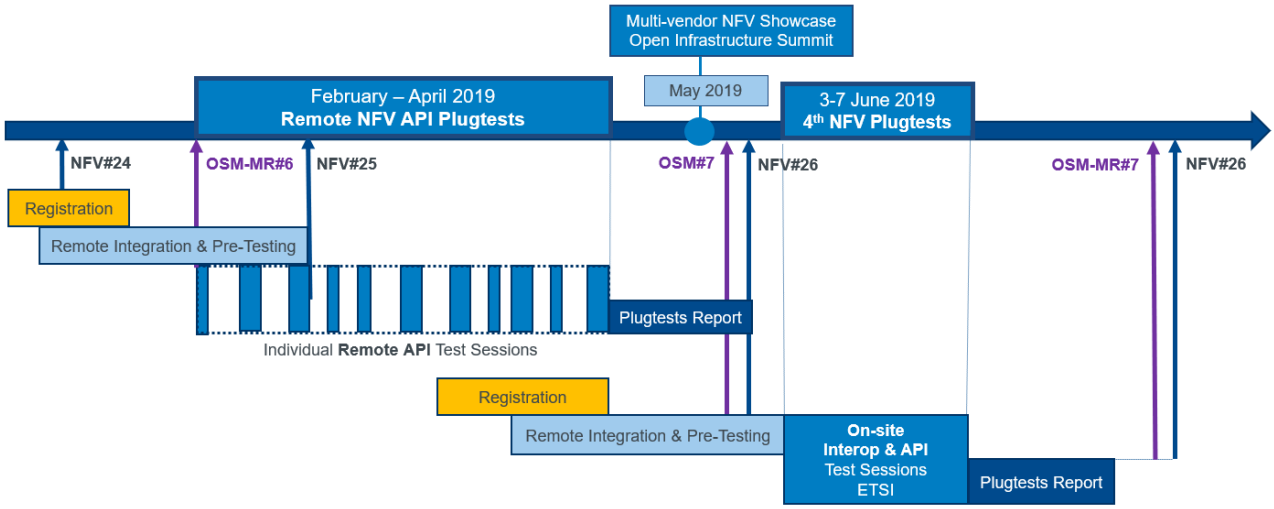

Figure 5. 4th NFV Plugtests timeline

Registration to Plugtests was open from March to mid-April 2019 to any organisation willing to participate with a Function Under Test, or to support the testing. A number of observers (network operators, research, and academia) and supporting Open Source communities participated to the test plan development and review. Overall, over 200 people were involved in the Plugtests preparation either locally or remotely.

The following section describe the different phases of the Plugtests preparation. It is worth noting the preparation phases, weekly conf-calls were run among organisers and participants to discuss and track the remote-integration progress, anticipate and solve technical issues, review the test plan, and prepare for the face to face test sessions.

4.2.1 Remote Integration

During the Remote Integration phase, the following activities were run in parallel:

1) FUT Documentation

Participants documented their FUTs, by filling in or updating a form compiling the Interoperability Features Statement (IFS) and Technical Questions (TQ) concerning their implementations. The final IFS Templates for each type of FUT can be found in the annex of the 4th NFV Plugtests Test Plan [4NFVPLU-TP].

Participants providing VNFs complemented their documentation with diagrams and resource requirements. As for the previous NFV Plugtests, some example VNFs and NSs were made available by the Plugtests team to facilitate the documentation of participating VNFs. These examples are documented in the Annex A of the 1st NFV Plugtests Report [1NFVPLU-R].

Participants providing MANO solutions developed and made available descriptor samples for the VNF and NS examples and supported the VNF providers in the creation of their own VNF and NS Descriptors.

Participants providing NFV Platforms created and documented projects and credentials for each participating MANO solution, and exposed and documented the North Bound Interfaces (NBI) of their VIM.

All the information described above was made available in the Plugtests WIKI, so that it could be easily maintained and consumed by participants.

2) Test Plan Development

The Test Plan development was led by ETSI Centre for Testing and Interoperability following the methodology and guidelines defined by ETSI NFV TST WG in TST002 and TST007, building on the learnings and achievements of previous NFV Plugtests.

The Test Plan was developed and consolidated in an iterative way, taking into account input and feedback received from different stakeholders: ETSI NFV TST WG, supporting Open Source Communities and Plugtests participants. See details in Clause 8.

3) Connection to HIVE

The interconnection of different FUTs involved in the testing relies on HIVE: Hub for Interoperability and Validation at ETSI. NFV Plugtests Programme participants can keep and use their connection to HIVE between Plugtests events. New participants that joined the programme after the Remote API Plugtests, were invited and helped to get their implementations available on HIVE.

At the end of this phase, over 45 remote sites were connected to HIVE and each of them was allocated a dedicated network. The interconnection of remote labs allowed to run integration and pre-testing tasks remotely among any combination of participating FUTs and helped to ensure an efficient use of the face to face Plugtests time and a smoother run of the Test Sessions.

A site-to-site connection to HIVE was mandatory for participants providing NFV Platforms and MANO Solutions, and highly recommended for participants providing VNF and VIM software. The latest could also rely on client-to-site connection to HIVE, as long as they had no software (i.e. support function) running locally in their labs and only required access to remote labs for trouble shooting and infrastructure access purposes

Additional details on the remote test infrastructure are provided in Clause 6.

4) Once the above steps were completed, FUTs could start cross-FUT remote integration, see 7.1 for details on the procedures.

4.2.2 Pre-testing

Once remote integration was completed, participants had the opportunity to run remote pre-testing among different combinations of VNF, MANO and NFV Platforms.

The pre-testing test plan was a subset of the tests run during the previous NFV Plugtests, in order to allow participants to concentrate on advanced features and configurations during the on-site phase.

Additional details on the pre-testing plan and procedures are provided in Clause 7.

4.2.3 On-site

From 3 -7 June, participants sent representatives to ETSI to collaboratively run the Test Sessions. The on-site Plugtests week was organised as follows:

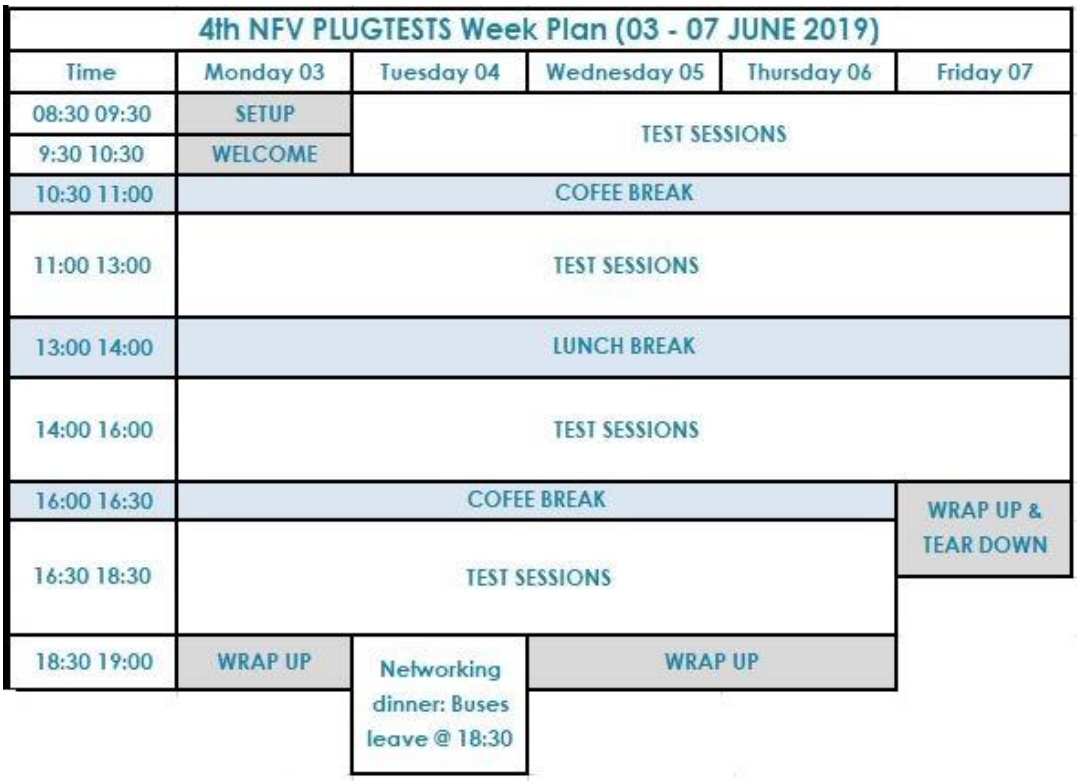

Figure 6. Plugtests on-site week plan

The Test Sessions were run in parallel, so that all the participants were busy all the time. At a given moment in time, there were up to 11 active test sessions: 8 on NFV IOP (one per MANO solution), 2 on NFV API conformance (one per conformance test system) and 1 on MEC IOP.

4.3 Tools

4.3.1 Plugtests Wiki

The NFV Plugtests Wiki was the main entry point for all the information concerning this event, from logistics aspects to testing procedures. Access to this Wiki is restricted to companies participating to the NFV Plugtests Programme.

4.3.2 Test Session Scheduler

The Test Session Scheduler allowed the Plugtests organisers to produce a daily schedule during the on-site phase. This tool has the following objectives:

- maximise the number of test sessions
- balance the amount of test sessions among participants
- take into account supported features of the participating FUTs
- minimise the number of participants not involved in a test session anytime.

The picture below shows a partial view a daily schedule. Each yellow box corresponds to a specific Test Session addressing a Multi-Vendor (or Single-Vendor) Network Service configuration including 1, 2 or more VNFs from different providers, 1 MANO solution and 1 or more VIM&NFVI. For each of these sessions a Test Session Report was reported (see next clause). In addition to the pre-scheduled test sessions, participants were invited to request, run and report results for additional test sessions targeting either pre-testing or advanced configurations addressing more complex combinations of FUTs and features.

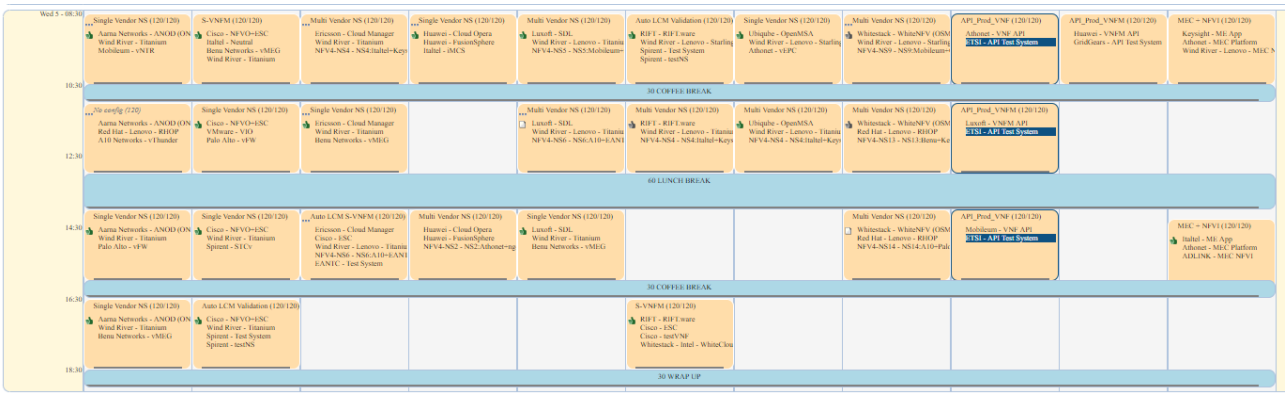

Figure 7. Daily Schedule example

4.3.3 Test Reporting Tool

The Test Reporting Tool guides participants through the Test Plan during the Test Sessions and allows them to create Test Session Reports compiling detailed results for each test case in scope. It allows reporting on pre-scheduled Test Sessions, but also on Test Sessions organised on the fly among participants to prepare, complete or complement the scheduled testing (freestyle sessions).

Only the companies providing the FUTs and Test Functions (when applicable) for each specific Test Session have access to the Test Session Reports (TSR) contents and specific results. All the companies providing the FUTs for a Test Session, i.e. VNF provider(s), MANO provider and NFV Platform provider(s) are required to verify and approve the reported results at the end of the session.

						Benu Networks - vMEG
3996		2019-06-05 11:00	120	CISCO	Single Vendor NS	Cisco - NFVO+ESC VMware - VIO Palo Alto - vFW
3997		2019-06-04 08:30	120	LUXOFT	Multi Vendor NS	Luxoft - SDL Wind River - Lenovo - Titanium NFV4-NS6 - NS6:A10+EANTC
3998		2019-06-05 14:00	120	AARNA (ONAP)	Single Vendor NS	Aarna Networks - ANOD (ONAP) Wind River - Titanium Palo Alto - vFW
4001		2019-06-04 16:30	120	MEC Track	MEC Basic	Athonet - ME App ADLINK - MEC Platform
4003		2019-06-05 14:00	120	LUXOFT	Single Vendor NS	Luxoft - SDL Wind River - Titanium Benu Networks - vMEG
4004		2019-06-05 08:30	120	CISCO	S-VNFM	Cisco - NFVO+ESC Italtel - Neutral Benu Networks - vMEG Wind River - Titanium
4005		2019-06-05 11:00	120	WHITESTACK (OSM)	Multi Vendor NS	Whitestack - WhiteNFV (OSM) Red Hat - Lenovo - RHOP NFV4-NS13 - NS13:Benu+Keysight
4007		2019-06-06 08:30	120	ERICSSON	Single Vendor NS	Ericsson - Cloud Manager VMware - VIO Palo Alto - vFW
4008		Freestyle			MEC Basic	Keysight - ME App A10 Networks - MEC Platform
4009		2019-06-03 14:30	120	MEC Track	MEC + NFVI	Keysight - ME App A10 Networks - MEC Platform Red Hat - Lenovo - MEC NFVI
4011		2019-06-05 08:30	120	RIFT	Auto LCM Validation	RIFT - RIFT.ware Wind River - Titanium Spirent - Test System Spirent - testNS
4012		2019-06-05 08:30	120	RIFT	Auto LCM Validation	RIFT - RIFT.ware Wind River - Lenovo - StarlingX Spirent - Test System Spirent - testNS
4013		2019-06-05 14:15	120	MEC Track	MEC + NFVI	Italtel - ME App Athonet - MEC Platform ADLINK - MEC NFVI
4014		2019-06-04 14:00	120	MEC Track	MEC + NFVI	Keysight - ME App Benu Networks - MEC Platform

Figure 8. Test Reporting Tool (extract of the TSR list)

Another interesting feature of this tool is the ability to generate real-time statistics (aggregated data) of the reported results, per test case, test group, test session or overall results. These stats are available in real time for all participants and organisers and allow tracking the progress of the testing with different levels of granularity, which is extremely useful to analyse the results.

5 Participation

5.1 Functions Under Test

The tables below summarise the different Functions Under Test provided by the Plugtests participants, and the location from where they were provided / supported or connected to the HIVE network:

5.1.1 VNFs

Organisation	Solution	Team Location	Short Description
A10 Networks	vThunder	USA/Germany	Virtual Traffic detection function: ADC, CGN, FW, Security
Athonet	vEPC	Italy	Virtual EPC
Benu Networks	vMEG	USA / India	Virtual multi-service Edge Gateway
EANTC	NFV Test Function	Germany	Traffic generator
Eurecom	OAI-EPC	France	OAI EPC NW Functionality
Fortinet	FortiGate	France	FW, Security
Italtel	iMCS	Italy	IMS Core, Border and Service Layer
Italtel	IVAS	Italy	MEC App
Keysight	IxLoad	USA/Romania	Application traffic simulator
Keysight	IxNetwork	USA/Romania	Application traffic simulator
Keysight	BPS	USA/Romania	Application traffic simulator
Mobileum	vNTR,	India	Virtual NTR, Roaming
Mobileum	vDRA	India	Virtual Diameter Routing Agent
ng4T	NG40 vTester	Germany	Simulator, functional, capacity and load vTester
Palo Alto Networks	vFW	USA/China/EU	Firewall, Security
Spirent	CloudStress	California, USA	Synthetic Workload generator
Spirent	TestCenter Virtual	California, USA	Traffic Generator, Traffic Analyzer, Capture, Control Plane emulator
ZTE	vCube-vFW	China	Firewall, CGN

Table 1. VNFs Under Test

5.1.2 NS

The VNFs under test were combined in multi-vendor network services as follows:

NS Name	VNFs
NS1: A10+Fortinet+Spirent	vThunder (A10) FortiGate (Fortinet) TestCenter Virtual (Spirent)
NS2: Athonet+ng4T	vEPC (Athonet) vTester (ng4T)
NS3: Benu+Italtel	vMEG (Benu) iMCS (Italtel)
NS4: Italtel+Keysight	iMCS (Italtel) IxLoad (Keysight)
NS5: Mobileum+ng4T	vNTR (Mobileum) vTester (ng4T)
NS6: A10+EANTC	vThunder (A10) TestVNF (EANTC)
NS7: Palo Alto+EANTC	vFW (Palo Alto) TestVNF (EANTC)
NS8: ZTE+Spirent	vCube-vFW (ZTE) TestCenter Virtual (Spirent)
NS9: Mobileum+Eurecom+ng4T+Fortinet	vDRA (Mobileum) OAI (Eurecom) vTester (ng4T) FortiGate (Fortinet)
NS10: Athonet+Keysight	vEPC (Athonet) IxLoad (Keysight)
NS11: Palo Alto+ng4T	vFW (Palo Alto) vTester (ng4T)
NS12: Mobileum+ Keysight	vNTR (Mobileum) IxLoad (Keysight)
NS13: Benu + Keysight	vMEG (Benu) IxLoad (Keysight)
NS14: A10+Palo Alto+Keysight	vThunder (A10) vFW (Palo Alto) IxLoad (Keysight)
NS15:A10+Spirent	vThunder (A10) TestCenter Virtual (Spirent)
NS16: Benu+Spirent	vMEG (Benu) TestCenter Virtual (Spirent)

Table 2. NSs Under Test

5.1.3 MANOs

Organisations	Name	Location	Short Description
Aarna Networks	ANOD	NH, USA / India	ONAP distribution (Casablanca)
Cisco	NFVO and ESC	USA/UK	NFVO + Generic VNFM
Ericsson	Orchestrator Cloud Manager	Sweeden/Ireland	NFVO + Generic VNFM,

Huawei	CloudOpera Orchestrator NFV	Xi'an/China	NFVO + Generic VNFM
Luxoft	SDL	Romania	NFVO + Generic VNFM
Nokia	CBAM	Hungary	VNFM
RIFT	RIFT.ware	USA / India	NFVO + Generic VNFM
Ubiqube	OpenMSA	Ireland / France / India	Network and Security Automation Framework
Whitestack	WhiteNFV	USA	OSM distribution (Rel FIVE) - NFVO + Generic VNFM
ZTE	CloudStudio	China	NFVO + Generic VNFM

Table 3. MANOs Under Test

5.1.4 VIM&NFVIs

Organisations	Name	HW	Location	Short Description
Adlink UC3M	Fog05	Adlink POD @Adlink	Paris, France	Distributed VIM for fog environment
Red Hat Lenovo	Red Hat OpenStack Platform	Lenovo POD - SR630/SR650 and OCP servers - G8272/NE2572 and G8052 switches	NC, USA	OpenStack Queens
Red Hat Lenovo	Lenovo Open Cloud	Lenovo POD - SR630/SR650 and OCP servers - G8272/NE2572 and G8052 switches	NC, USA	OpenStack Queens
VMware	VCD	Dell POD @OSM Remote Lab	Canada	vCloud Director 9.7 + NSX-V
VMware	VIO	Dell POD @OSM Remote Lab	Canada	VMware Integrated Openstack (Queens) + NSX-T
Whitestack Intel	WhiteCloud	Intel POD @OSM Remote Lab	USA	OpenStack Queens
Wind River	Titanium Cloud R5 – Core Configuration	Wind River @OSM Remote Lab	Hudson, MA, USA	OpenStack Pike + OF 1.3
Wind River Lenovo	Titanium Cloud R5 – Core Configuration	Lenovo POD - SR630/SR650 and OCP servers - G8272/NE2572 and G8052 switches	NC, USA	OpenStack Pike + OF 1.3
Wind River Lenovo	StarlingX 1.0	Lenovo POD - SR630/SR650 and OCP servers - G8272/NE2572 and G8052 switches	NC, USA	OpenStack Pike + OF 1.3
ETSI	*ETSI VIM*	*ETSI POD @ETSI*	*France*	*Test Environment for API Testing*

Table 4. VIM&NFVIs Under Test

5.2 Test Functions

In addition to the Test VNFs included in the VNF section, the following Test Functions were available during the Plugtests to support or complement the scope of the Test Sessions. Their use was optional.

Organisation	Name	Team Location	Short Description
EANTC	Auto LCM Validation	Berlin, Germany	TD automation tool
Grid Gears	Standards as a Service	Austria	NFV API Conformance (TST010)
Spirent	CloudSure	California, USA	Cloud and NFV Test Case driven validation with automatic NFV Lifecycle Validation
ETSI	Test System	ETSI, France	NFV API Conformance (TST010)

Table 5. Test Functions

5.3 Technical Support

The organisations below provided technical support and expertise to the Plugtests Team and contributed actively to the test plan development and technical arrangements to prepare and run the Plugtests.

Organisation	Role
EdgeGap	Technical Support
Nextworks	Technical Support
XflowResearch	Technical Support

Table 6. Technical Support

5.4 Observers

The following organisations joined the NFV Plugtests as observers and contributed with technical advice and test plan review:

Organisation	Role
DOCOMO	Telecommunications service provider

Table 7. Observers

5.5 Open Source Communities

The Open Source communities listed below were actively involved in the Plugtests preparation and contributed to the Test Plan review. Their solutions were widely present in the Test Sessions, sometimes through multiple distributions:

Community / Project	Role	Details
ETSI OSM - Open Source MANO	MANO	https://osm.etsi.org
ONAP	MANO	https://www.onap.org/
Fog05	VIM&NFVI	https://projects.eclipse.org/proposals/eclipse-fog05
Open Stack	VIM&NFVI	https://www.openstack.org
StarlingX	VIM&NFVI	https://www.starlingx.io/
Open Air Interface	VNF (vEPC)	http://www.openairinterface.org

Table 8. Supporting Open Source communities

6 Test Infrastructure

6.1 HIVE

The remote integration, pre-testing and on-site phases were enabled by the NFV Plugtests Programme's HIVE network

Figure 9. NFV Plugtests HIVE network

The NFV HIVE (Hub for Interoperability and Validation at ETSI) network interconnects securely participants' remote labs and Functions under Test and allows for remote multi-party interoperability testing and validation activities. Over 45 remote locations including several OSM Remote Labs participating to the Plugtests leveraged the HIVE network to make their Functions Under Test available for the test sessions.

During the on-site phase, a local network deployed in the Plugtests room allowed participants to access the remote labs, the local VNF Repository and some support functions running on ETSI servers. One participating NFV Platform (VIM&NFVI) was also deployed locally in the ETSI Lab and connected to HIVE.

The figure below describes how the NFV and OSM HIVE networks are interconnected to support the NFV Plugtests activities.

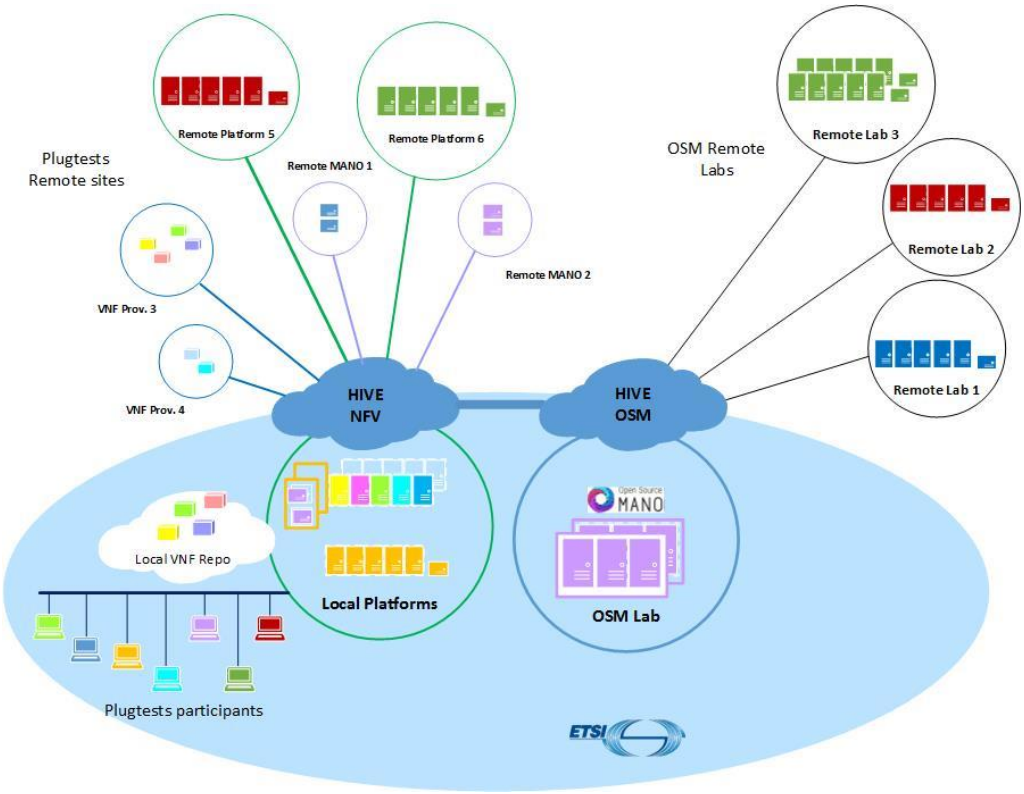

Figure 10. Remote Test Infrastructure

6.2 NFV API Conformance Test System

The NFV API Conformance Test Sessions relied on a Test System acting as API consumer for the NFV APIs exposed by different NFV components over different reference points. The capabilities required for the test system were:

- Sending configurable HTTP(S) requests
- Allowing custom payloads to be exchanged
- Automatically applying headers validation on the response payloads
- Automatically applying schema validation on the response payloads

The test system was deployed as a virtual machine connected to HIVE and able to run the Robot Framework Test Suites developed in the context of [TST010] NFV API Conformance Test Specifications. The Test Suites were developed from the OpenAPI definitions provided with ETSI NFV SOL API Specifications [SOL002], [SOL003] and [SOL005] by ETSI NFV SOL WG. The VM was based on ubuntu 16-04. The following software and python libraries were installed as required to run the test suites:

- Software:
  - python3[PYTHON] and pip [PIP] to run RobotFramework Test Suites
  - Java JRE[JAVA] and MockServer [MOCK] application to handle notifications server
  - Packet capturing software [TCP-DUMP] and [WHIRESHARK] to perform manual network traffic and data packets verification.
- Libraries:
  - Robot Framework[RF], JSONSchemaLibrary[JSON-SCH-LIB] and DependencyLibrary[DEP-LIB] through pip
  - JSONLibrary[JSON-LIB] and MockServerLibrary[MOCK-LIB] through source code (git repo) to apply extensions.

The execution of the tests was performed via command line leveraging on the utilities provided by the Robot Framework tools to run the test suites as a collection of individual and automated API Conformance test cases.

7 Test Procedures

7.1 Remote Integration Procedures

Remote integration procedures for different FUTs and FUT combinations were documented in the WIKI (see next chapters) and the progress captured in a multi-dimensional tracking matrix which was reviewed regularly during the preparation calls. The tracking matrix was like the one shown below.

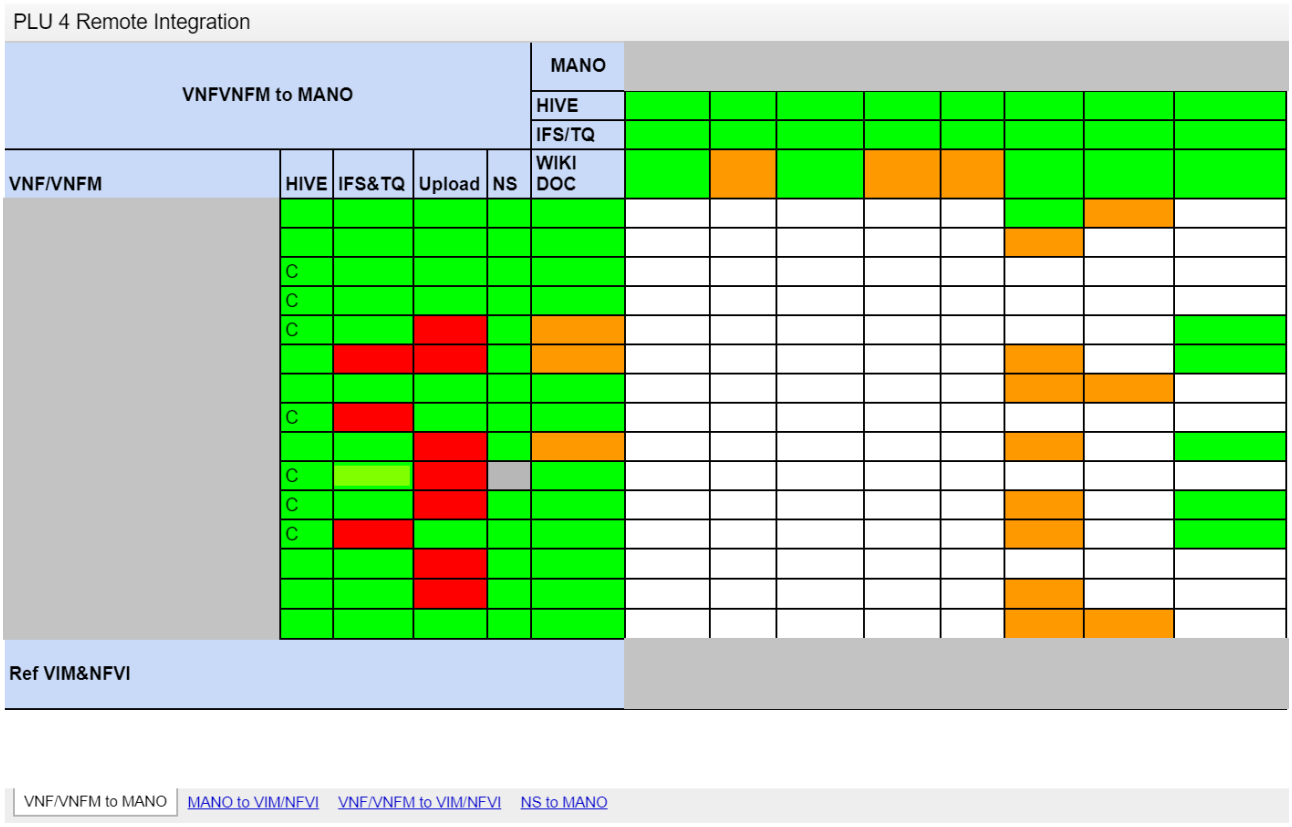

Figure 11. Pre-testing tracking matrix

The progress on each of the dimensions of the remote integration activities (VNF to MANO, MANO to VIM&NFVI VNF to VIM&NFVI and NS to MANO) was tracked following the colour code described below:

Pre-testing Matrix Colour Code	
	Not Started / Not Reported
	Not Applicable
	Pending
	Ongoing
	Completed

Table 9. Remote integration matrix colour code

7.1.1 Per-FUT Procedures

The next sections describe the independent procedures for each type of FUT:

7.1.1.1 VNF

1. **HIVE:** Connect to **HIVE** (site to site or client to site)
2. **IFS&TQ:** Fill in the VNF Interoperability Feature Statements and Technical Questions form
3. **Upload**: Upload VNF Package(s), descriptors, artifacts, and SW image(s) to the central VNF Repository.
4. **NS:** Team up with other VNF providers to build a multi-vendor NS
5. **WIKI DOC**: Provide a schema describing your VNF / NS

7.1.1.2 MANO

1. **HIVE:** Connect to **HIVE**. site-to-site VPN connectivity is required for MANO solutions
2. **IFS&TQ:** Fill in the MANO Interoperability Feature Statements and Technical Questions form
3. **WIKI DOC**: provide VNF and NS Descriptor examples for the Ref VNF&NS examples.

7.1.1.3 VIM&NFVI

1. **HIVE:** Connect to **HIVE** with site-to-site VPN or ship HW to ETSI
2. **VIM Ready**: Install, start and configure VIM
3. **IFS&TQ:** Fill in the VIM&NFVI Interoperability Feature Statements and Technical Questions form
4. **WIKI DOC**: provide additional information required for MANO solutions to integrate: NBI IP@, projects, credentials, VNF Management IP pool, etc...

7.1.2 Cross-FUT Procedures

7.1.2.1 MANO to VIM&NFVI

1. Test connectivity from MANO to VIM&NFVI & from VIM&NFVI to MANO
2. Connect MANO to VIM (get NBI IP, project and credentials on VIM&NFVI wiki page)
3. Verify VIM resources can be accessed from MANO
4. Specify the reference VNF in the pre-testing table
5. Upload Ref VNF to VIM
6. On-board, instantiate and terminate Reference VNF from MANO (* see pre-testing sessions below)

7.1.2.2 VNF to VIM&NFVI

1. Upload VNF image to VIM
2. Verify physical network connectivity in NFVI will allow for VNF/NS deployment
3. Manual creation of VM's and network infrastructure required for this VNF (to prepare MANO On-board automatic execution)
4. Manual execution Instantiate VNF steps (to prepare MANO automatic execution)

5. Manual execution Scale in/out VNF steps (to prepare MANO automatic execution)
6. Manual execution Terminate VNF steps (to prepare MANO automatic execution)

7.1.2.3 VNF to MANO

1. Identify the reference VIM&NFVI(s) that will be used for pre-testing
2. Create VNFD/NSD for MANO, and upload to VNF Repository (VNF folder under NFVPLU4)
3. On-board VNFD/NSD to MANO
4. Upload VNF image(s) to Ref VIM
5. Instantiate VNF/NS from MANO
6. Scale in/out
7. Terminate VNF/NS

7.1.2.4 NS to MANO

1. Identify the reference VIM&NFVI(s) that will be used for pre-testing
2. Create multi-vendor NSDs for MANO, and upload to VNF Repository (NS folder under NFVPLU4)
3. On-board NSD to MANO
4. Upload VNF image(s) to Ref VIM
5. Instantiate NS from MANO
6. Scale in/out
7. Terminate NS

7.2 NFV Pre-testing Procedure

Following remote integration, participants were invited to run pre-testing sessions and formally capture the results in the Test Reporting Tool. The recommendation was to start pre-testing with a Single-VNF Network Service configuration involving one VNF, one MANO solution and one VIM&NFVI and increment complexity gradually.

Running and reporting results for a pre-testing session required all the involved parties (VNF, MANO, VIM&NFVI) to have successfully achieved remote integration (connection to HIVE, descriptors created and uploaded, images uploaded, credentials shared, VIM ready, etc...).

Participants were encouraged to find suitable testing partners in the pre-testing matrix (those that had completed remote integration) and arrange a convenient timeframe for pre-testing, leveraging Slack to facilitate discussions among the different teams.

Once the arrangements made the pre-testing procedure was as follows:

1) Open the Test Reporting Tool, go to the “Reports” TAB

2) Create a “Freestyle” report, select configuration = Pre-testing

3) Select the MANO, VIM&NFVI and VNF involved in the pre-testing session

Once the above completed, a Test Session Report would open, and participants would follow the Interoperability Testing Procedure remotely (see next clause)

7.3 NFV IOP Testing Procedure

During the on-site part of the Plugtests, a daily Test Session Schedule was produced with the Plugtests Scheduler. Test Sessions were organised in several parallel tracks, ensuring that all participants had at least one pre-scheduled Test Session every day. Pre-scheduled Test Sessions addressed the multi-vendor Network Service configuration, involving one MANO solution, one or more VIM&NFVIs and one network serviced with at least 2 VNFs from different providers.

Pre-scheduled test sessions could be completed with additional sessions addressing other test configurations / test groups, such as Multi-site, EPA, Service Function Chaining, or specific VNFM deployments (see 8.1 for details)

Participants could choose to run the above-mentioned additional testing as “freestyle” test sessions, and create a test report on the fly, or to ask the Plugtests team to schedule the additional sessions for them.

During each test session the procedure for interoperability testing was as follows:

1) The MANO, VIM&NFVI(s), and VNFs representative would sit together. Each VNF was expected to have previously run a single-VNF NS test session with the same MANO and VIM(s)

2) One representative of the team opened the Test Session Report and the Test Plan.

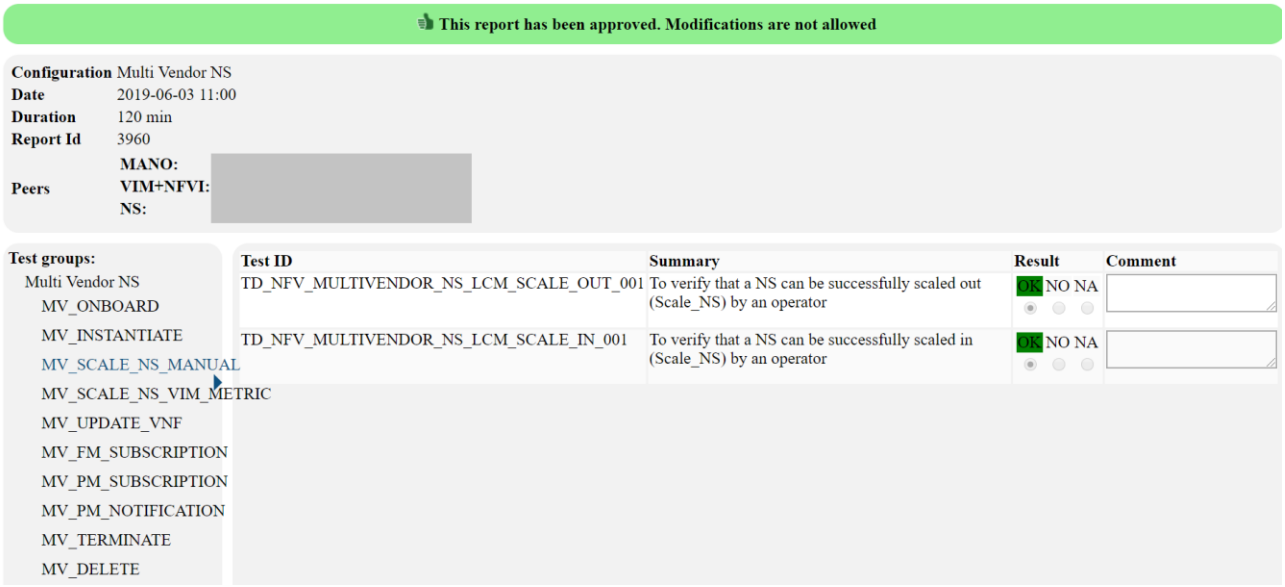

Figure 12. Test Session Report

3) For each Test in each group of the Test Plan:

a. The corresponding Test Description was applied to the target SUT Configuration:

SUT_MULTI_VENDOR_NS

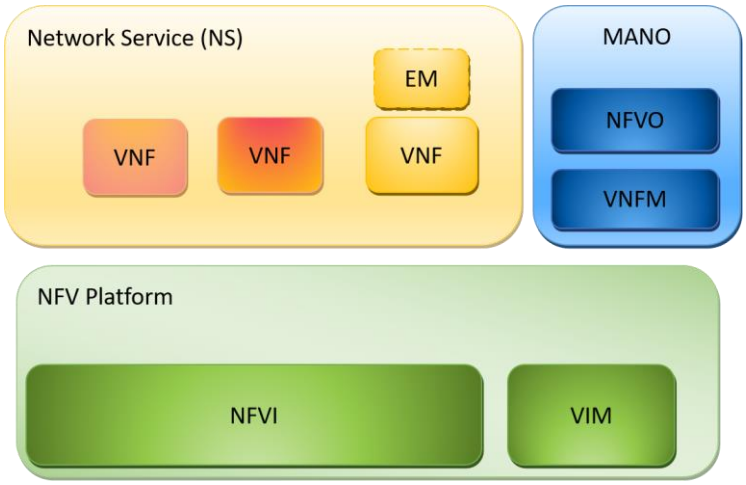

Figure 13. SUT Configuration example.

b. VNFs, MANO and VIM&NFVI providers jointly executed the different steps specified in the Test Description and evaluated interoperability through the different IOP Checks prescribed in it.

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_SCALE_OUT_001			
Test Purpose	To verify that a NS can be successfully scaled out (Scale_NS) by an operator			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_AUTO-LCM-VALIDATION			
References	[IFA013] Clause 7.3.3 [IFA007] Clause 7.2.4 [IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [SOL005] Clause 6.3 [SOL003] Clause 5.4.4			
Applicability	* [IFS_NFV_NFVO_5] NFVO supports NS scaling by adding/removing VNF instances * [IFS_NFV_NS_10] NS can scale out/in by adding/removing VNF instances (Scale_NS)			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger NS scale out (Scale_NS) in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives instantiation request for the additional VNF(s) to be deployed for the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for the additional VNFs in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the instantiation of the additional VNFs	
	4	IOP Check	Verify that the additional resources have been allocated in the VIM according to the descriptors	
	5	IOP Check	Verify that the additional VNF instance(s) are running and reachable from the management network	

	6	IOP Check	Verify that the initial configuration for the additional VNF(s) has been successfully applied	
	7	IOP Check	Verify that the additional VNF instance(s) in the NS are considered INSTANTIATED by the VNFM	
	8	IOP Check	Verify in the NFVO that the NS has been scaled out (i.e. query or display the NS instance resource)	
	9	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

Table 10. Test Description example

c. The Test Result was reported to the Test Session Report, as follows:

i. OK: all IOP Checks were successful

ii. NOK: at least one IOP Check failed. A comment was requested.

iii. NA: the feature was not supported by at least 1 of the involved FUTs. A comment was requested to clarify the missing feature.

4) Once all the tests in the Test Session Report were executed and results recorded, the VNFs, MANO and VIM&NFVI providers reviewed the Report and approved it.

7.4 NFV API Testing Procedure

The NFV API Conformance Test Sessions aimed at validating the conformance of the participants FUTs to the SOL002, SOL003 and SOL005 API specifications, while validating the API Conformance Robot Test Suite. The Test System was run in ETSI premises and connected to HIVE, which provided the connectivity to the participating FUTs, either local or remote. The Test System run the Robot Framework Test Suites developed for the NFV API Conformance Test Specification [TST010] and made available in the ETSI Forge [ROBOT-TS]. It is worth mentioning that the development of the NFV API Conformance Test Specification was still in progress at the time of this Plugtests, and the feedback of the NFV API Plugtests activities was fundamental input for the validation and consolidation of the Robot Framework Test Suites before publication, as detailed in clause 10.

1. The following Test Procedures were defined to make sure the testing was run in an efficient and consistent way across remote test sessions. Four different phases were foreseen: Preparation Phase

2. Runtime Phase

3. Results Processing Phase

4. Results Reporting Phase

The planned duration for each test session was set to a total 2 hours

7.4.1 Preparation Phase

The preparation phase is intended to cover all the preliminary steps to make sure the tests could be executed smoothly. This phase covered also the check of the connectivity and reachability from the ETSI-hosted API Test System to the participating FUT. Even if most of the preparatory actions could be performed online, in the first part of the Test Session, the Implementation Conformance Statements (ICS) and a set of configuration information was required from each participant. This information allowed to prepare and configure in advance the Test System for the given Test Session, including the selection of the conformance tests applicable for each FUT.

For this purpose, the Plugtests Team prepared the ICS and Configuration Parameters tables (see Annex A) and participants were requested to fill them in ahead of the Test Sessions considering the capabilities of their FUTs, in terms of support of applicable SOL specifications interfaces and operations.

The main outcome of the Preparation Phase was therefore the preparation and configuration of the API Test System with the applicable set of Robot Framework tests for the given FUT.

Runtime Phase

The main goal of the runtime phase was to execute the actual Robot Framework conformance tests. The execution of the tests was carried out in subsequent steps, according to the capabilities of the given FUT and the set of applicable Robot Framework tests. Where applicable indeed, the Plugtests team selected an order of execution of the tests with the aim of easing their automation.

As a general guideline, the API conformance tests were run subsequently for each applicable SOL specification resource endpoint, according to the set of tests available in the Test Plan.

The verdict of each test was considered OK (pass) if the FUT was compliant with the expected behaviour described in the applicable NFV SOL specifications in terms of:

1) Response code, headers and body

2) Resource status and/or post-condition

7.4.2 Results Processing Phase

For each run, Robot Framework generates a set of outputs to report on the execution of the tests. These outputs are:

- *report.html* : summary execution info & statistics
- *log.html* : detailed per test case log trace
- *output.xml* : detailed per test case output

In this phase, the Plugtests Team, with the support of the participants, was responsible to check and validate for each test execution these outputs, as a mean to assess the result of each applicable API conformance test.

7.4.3 Results Reporting Phase

All the test results were reported in the ETSI Test Reporting Tool [TRT], for analysis and statistics purposes. For this, the Plugtests Team filled the TRT with results of the test sessions, and the participants were requested to confirm and approve the results.

7.5 MEC IOP Testing Procedure

The testing procedure for the MEC track followed as much as possible the methodologies of NFV Interoperability Testing. Nevertheless, given the novelty of the proposal few aspects have been organized differently.

The schedule of test sessions was prepared during the on-site event, to allow vendors to make sure they would cover the NFV tests before trying the MEC one. Moreover, during the first day of the event, an overview of the MEC track was given to all the Plugtests participants audience and a final briefing with every vendor interested was hosted in a dedicated conference room.

During that meeting, each test was presented to the participants and discussed in detail. Each scenario and components were covered. A schedule for each session was agreed between participants, and those were booked through ETSI Test Reporting Tool [TRT]. MEC Track sessions were run in parallel or interleaved with NFV Interoperability sessions on a best effort case.

Scenarios were split in two (2) groups: one was for a MEC platform to test with a MEC application, and the second was for a similar setup to include an NFVI infrastructure. This was necessary as some MEC applications are network based and support some test cases which were initially made for the MEC platform.

The following is a sample report of a test case between a MEC application, a MEC platform and a NFVI infrastructure. Participants had to run through the 3 different set of tests, and each of them had to agree on the result (thus resulting in a thumbs up/down).

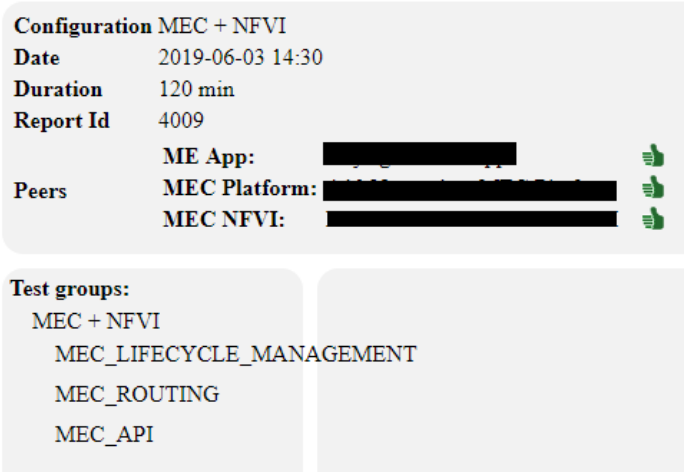

Figure 14. MEC Test Session Report

Each test session was pre-scheduled, and results were reported in ETSI Test Reporting Tool [TRT]. Every participant to each session was required to agree on the result for those to be valid.

8 Test Plans Overview

8.1 NFV Interoperability

This 4th NFV Plugtests Test Plan was developed by the NFV Plugtests team lead by ETSI Centre for Testing and Interoperability and following the interoperability testing methodology defined by ETSI NFV in [TST002] and building on the test plans of previous NFV Plugtests [1NFVPLU-TP], [2NFVPLU-TP] and [3NFVPLU-TP]

The following clauses summarise the different configurations and interoperability test cases in scope for this Plugtests, and how they were grouped to optimise test session scheduling, duration and results collection and analysis. The four main interoperability test groups identified for the 4th NFV Plugtests were: Single Vendor, Multi-Vendor NS, Specific VNFM, and Automatic LCM validation.

8.1.1 Single Vendor NS

The Single Vendor NS group was based in SUT_SINGLE-VENDOR_NS configuration on the Test Plan [4NFVPLU-TP]. It involves one MANO solution, one VIM&NFVI and one Network Service (NS), including just one VNF in the NS.

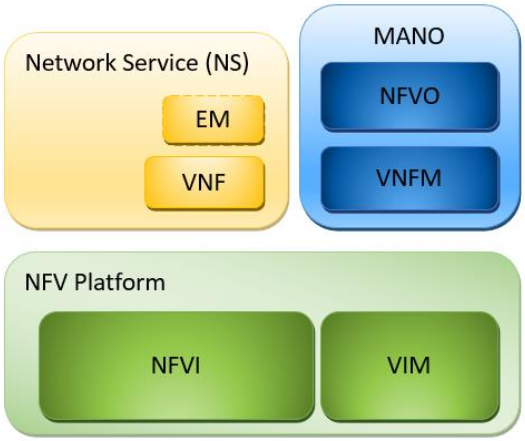

Figure 15: Single-Vendor NS SUT Configurations

For this configuration, the table below lists the groups and TDs that apply:

Group	Test IDs	# TDs
SV_ONBOARD	TD_NFV_ONBOARD_XXX	**2**
SV_INSTANTIATE	TD_NFV_NS_LCM_INSTANTIATE_XXX	**1**
SV_SCALE_NS_MANUAL SV_SCALE_NS_VNF_IND SV_SCALE_NS_VIM_METRI	TD_NFV_NS_LCM_SCALE_OUT/IN_xxx	**8**
SV_SCALE_VNF_MANUAL SV_SCALE_VNF_VNF_IND SV_SCALE_VNF_VIM_METRIC SV_SCALE_VNF_EM	TD_NFV_NS_LCM_SCALE_OUT/IN_VNF_xxx	**10**
SV_SCALE_NS_TO_LEVEL_MANUAL SV_SCALE_NS_TO_LEVEL_VNF_IND SV_SCALE_NS_TO_LEVEL_VIM_METRIC	TD_NFV_NS_LCM_SCALE_TO_LEVEL_xxx	**3**
SV_SCALE_VNF_TO_LEVEL_MANUAL	TD_NFV_NS_LCM_SCALE_TO_LEVEL_VNF_xxx	**4**

Group	Test IDs	# TDs
SV_SCALE_VNF_TO_LEVEL_VNF_IND SV_SCALE_VNF_TO_LEVEL_VIM_METRIC SV_SCALE_VNF_TO_LEVEL_EM		
SV_UPDATE_VNF	TD_NFV_NS_LCM_UPDATE_XXX	**2**
SV_FM_SUBSCRIPTION SV_FM_ALARMS	TD_NFV_FM_NS_XXX	**5**
SV_PM_JOB SV_PM_SUBSCRIPTION SV_PM_NOTIFICATION	TD_NFV_PM_ NS_XXX	**10**
SV_VNF_IND	TD_NFV_VNF_INDICATOR_XXX	**3**
SV_TERMINATE	TD_NFV_NS_LCM_TERMINATE_XXX	**1**
SV_DELETE	TD_NFV_TEARDOWN_DELETE_XXX	**2**

Table 11. Single Vendor NS Test Groups

The complete list of Test Descriptions can be found in [4NFVPLU-TP] Clause 6.1.

8.1.2 Multi Vendor NS

The Multi Vendor NS group was based in SUT_MULTI-VENDOR_NS FUT configurations on the Test Plan [4NFVPLU-TP]. It involves one MANO solution, one VIM&NFVI and one Network Service (NS), including several VNFs from different providers

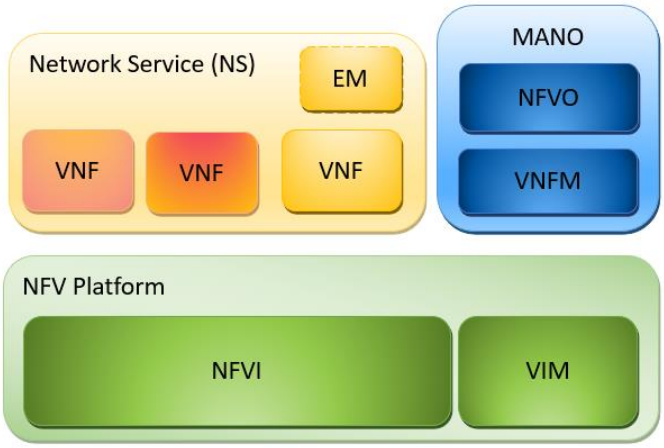

Figure 26: Multi Vendor NS SUT Configurations

For each of these 2 configurations, the table below lists the sub-groups and TDs that apply:

Group	Test IDs	# TDs
MV_ONBOARD	TD_NFV_ONBOARD_XXX	**2**
MV_INSTANTIATE	TD_NFV_NS_LCM_INSTANTIATE_XXX	**1**
MV_SCALE_NS_MANUAL MV_SCALE_NS_VNF_IND MV_SCALE_NS_VIM_METRI	TD_NFV_NS_LCM_SCALE_OUT/IN_xxx	**8**
MV_SCALE_VNF_MANUAL MV_SCALE_VNF_VNF_IND	TD_NFV_NS_LCM_SCALE_OUT/IN_VNF_xxx	**10**

Group	Test IDs	# TDs
MV_SCALE_VNF_VIM_METRIC MV_SCALE_VNF_EM		
MV_SCALE_NS_TO_LEVEL_MANUAL MV_SCALE_NS_TO_LEVEL_VNF_IND MV_SCALE_NS_TO_LEVEL_VIM_METRIC	TD_NFV_NS_LCM_SCALE_TO_LEVEL_xxx	**3**
MV_SCALE_VNF_TO_LEVEL_MANUAL MV_SCALE_VNF_TO_LEVEL_VNF_IND MV_SCALE_VNF_TO_LEVEL_VIM_METRIC MV_SCALE_VNF_TO_LEVEL_EM	TD_NFV_NS_LCM_SCALE_TO_LEVEL_VNF_xxx	**4**
MV_UPDATE_VNF	TD_NFV_NS_LCM_UPDATE_XXX	**2**
MV_FM_SUBSCRIPTION MV_FM_ALARMS	TD_NFV_FM_NS_XXX	**5**
MV_PM_JOB MV_PM_SUBSCRIPTION MV_PM_NOTIFICATION	TD_NFV_PM_ NS_XXX	**10**
MV_VNF_IND	TD_NFV_VNF_INDICATOR_XXX	**3**
MV_TERMINATE	TD_NFV_NS_LCM_TERMINATE_XXX	**1**
MV_DELETE	TD_NFV_TEARDOWN_DELETE_XXX	**2**

Table 22. Multi Vendor NS Test Groups

The complete list of Test Descriptions can be found in [4NFVPLU-TP] Clause 6.1.

8.1.4 Specific VNFM

The Specific VNFM group leverages the SUT_S-VNFM configuration as described in the Test Plan [4NFVPLU-TP].

This configuration involved one MANO solution, one VIM&NFVI and one VNF providing its own VNF Manager. The Specific VNFM and the MANO solutions were requested to both support the same mode (direct and/or in-direct_ for resource management

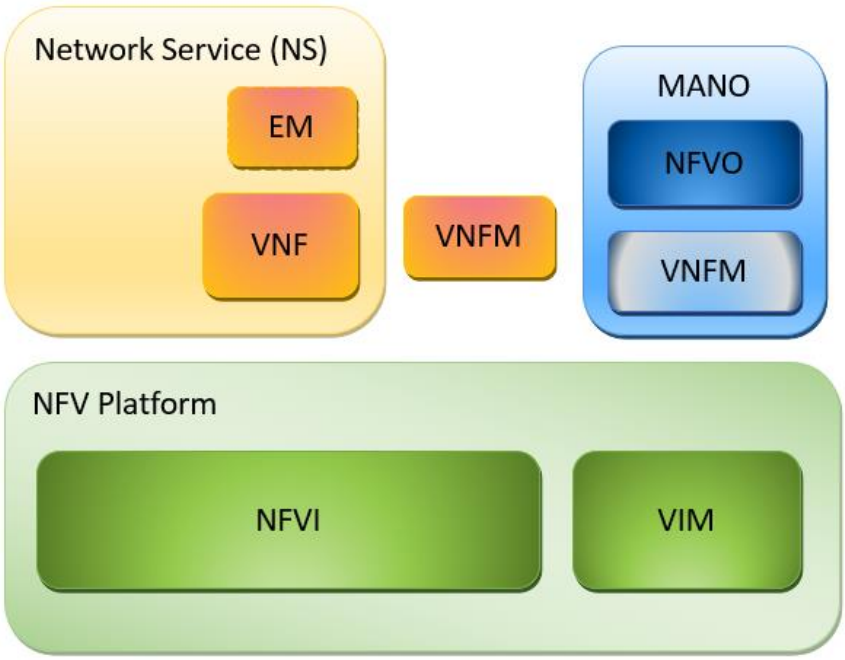

Figure 17: Specific VNFM SUT Configuration

For this configuration, the table below lists the groups and TDs in scope:

Group	Test IDs	# TDs
S-VNFM_ONBOARD	TD_NFV_ONBOARD_XXX	**2**
S-VNFM_INSTANTIATE	TD_NFV_NS_LCM_INSTANTIATE_XXX	**1**
S-VNFM_SCALE_NS_MANUAL S-VNFM_SCALE_NS_VNF_IND S-VNFM_SCALE_NS_VIM_METRIC	TD_NFV_NS_LCM_SCALE_OUT/IN_xxx	**8**
S-VNFM_SCALE_VNF_MANUAL S-VNFM_SCALE_VNF_VNF_IND S-VNFM_SCALE_VNF_VIM_METRIC S-VNFM_SCALE_VNF_EM	TD_NFV_NS_LCM_SCALE_OUT/IN_VNF_xxx	**10**
S-VNFM_SCALE_NS_TO_LEVEL_MANUAL S-VNFM_SCALE_NS_TO_LEVEL_VNF_IND S-VNFM_SCALE_NS_TO_LEVEL_VIM_METRIC	TD_NFV_NS_LCM_SCALE_TO_LEVEL_xxx	**3**
S-VNFM_SCALE_VNF_TO_LEVEL_MANUAL S-VNFM_SCALE_VNF_TO_LEVEL_VNF_IND S-VNFM_SCALE_VNF_TO_LEVEL_VIM_METRIC S-VNFM_SCALE_VNF_TO_LEVEL_EM	TD_NFV_NS_LCM_SCALE_TO_LEVEL_VNF_xxx	**4**
S-VNFM_UPDATE_VNF	TD_NFV_NS_LCM_UPDATE_XXX	**2**
S-VNFM_FM_SUBSCRIPTION S-VNFM_FM_ALARMS	TD_NFV_FM_NS_XXX	**5**
S-VNFM_PM_JOB S-VNFM_PM_SUBSCRIPTION S-VNFM_PM_NOTIFICATION	TD_NFV_PM_ NS_XXX	**10**
S-VNFM_TERMINATE	TD_NFV_NS_LCM_TERMINATE_XXX	**1**
S-VNFM_DELETE	TD_NFV_TEARDOWN_DELETE_XXX	**2**

Table 33. Specific VNFM Test Groups

The complete list of Test Descriptions can be found in [4NFVPLU-TP] Clause 6.1.

8.1.5 Auto LCM Validation

The Auto LCM Validation group leverages the SUT_AUTO-LCM-VALIDATION configuration as described in the Test Plan [4NFVPLU-TP].

It involves one MANO solution, one VIM&NFVI and at least two VNFs. The Test System oversees automating the triggers and IOP checks as described in the Test Plan.

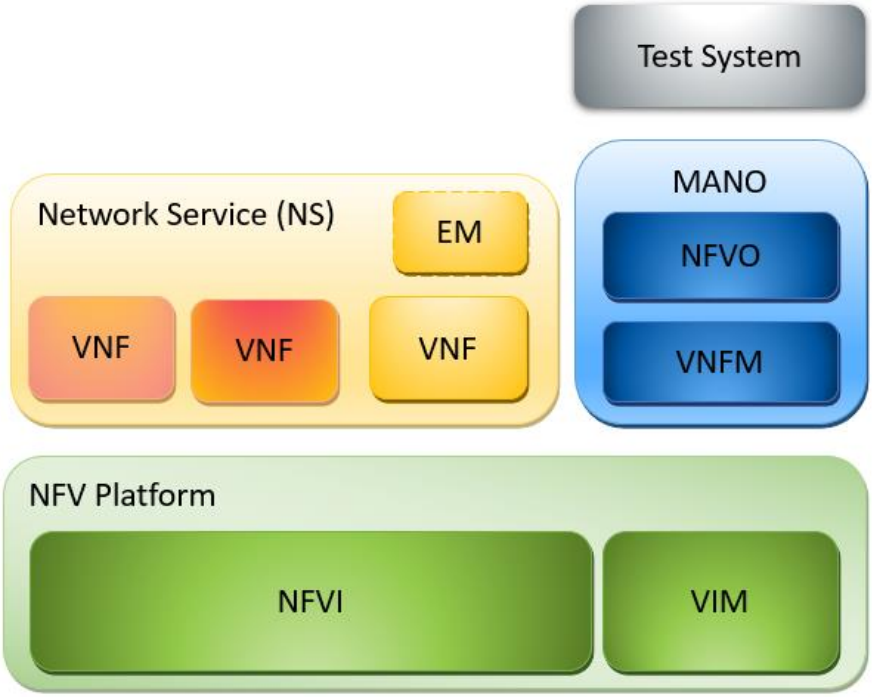

Figure 18: Auto LCM Validation SUT Configuration

For this configuration, the table below lists the groups and TDs in scope:

Group	Test IDs	# TDs
AUTOLCM_ONBOARD	TD_NFV_ONBOARD_XXX	**2**
AUTOLCM_INSTANTIATE	TD_NFV_NS_LCM_INSTANTIATE_XXX	**1**
AUTOLCM_SCALE_NS_MANUAL AUTOLCM_SCALE_NS_VNF_IND AUTOLCM_SCALE_NS_VIM_METRIC	TD_NFV_NS_LCM_SCALE_OUT/IN_xxx	**8**
AUTOLCM_SCALE_VNF_MANUAL AUTOLCM_SCALE_VNF_VNF_IND AUTOLCM_SCALE_VNF_VIM_METRIC	TD_NFV_NS_LCM_SCALE_OUT/IN_VNF_xxx	**10**
AUTOLCM_UPDATE_VNF	TD_NFV_NS_LCM_UPDATE_XXX	**3**
AUTOLCM_FM_SUBSCRIPTION AUTOLCM_FM_ALARMS	TD_NFV_FM_NS_XXX	**4**
AUTOLCM_TERMINATE	TD_NFV_NS_LCM_TERMINATE_XXX	**1**
AUTOLCM_DELETE	TD_NFV_TEARDOWN_DELETE_XXX	**2**

Table 44. Auto LCM Validation Test Groups

The complete list of Test Descriptions can be found in [4NFVPLU-TP] Clause 6.1

8.2 NFV API Conformance

This NFV API Conformance test plan was based on the Robot Framework Test Cases developed for [TST010] NFV API Conformance Test Specification, addressing FUT API Conformance to [SOL002], [SOL003] and [SOL005] specifications.

The test system acted as consumer for the NFV API produced by the FUTs, thus focusing only on testing the server-side of the NFV API under Test.

The following clauses summarise the test cases in scope for this NFV API Plugtests, and how they were grouped to optimise the remote test session executions.

The complete Test Specification can be found in [TST010] and the associated Robot Test Cases are available in the ETSI Forge [ROBOT-TS]

8.2.1 VNF/EM

The test configuration as described in the figure below was defined to test the interfaces exposed by a VNF/EM towards the VNFM, such as VNF Configuration API and VNF Indicator API. The test system acts as the VNFM (SOL002 API Client)

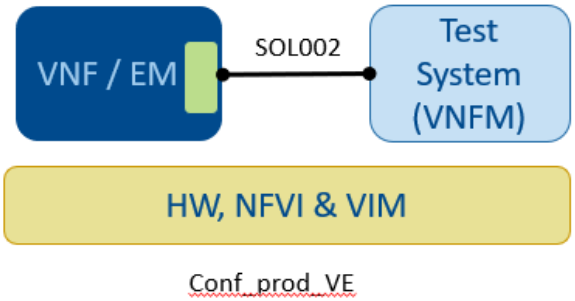

Figure 19: VNF/EM API Test Configuration

8.2.2.1 SOL002

API	Test IDs	# Test Cases
VNF Configuration API	**TD_API_VNF_SOL002_VNF_CONF_1 .. 6**	6
VNF Indicator API	**TD_API_VNF_SOL002_VNF_IND_1 .. 36**	36

Table 55. VNF/EM SOL002 Test Groups

The complete list of Test Cases can be found in [TST010] Clause 6.3

8.2.2 VNFM

The test configuration as described below was defined to test the interfaces exposed by a VNFM towards VNF/EM (SOL002) or towards the NFVO (SOL003). The test system is acting either as VNF/EM (SOL002 API client) or as NFVO (SOL003 API client)

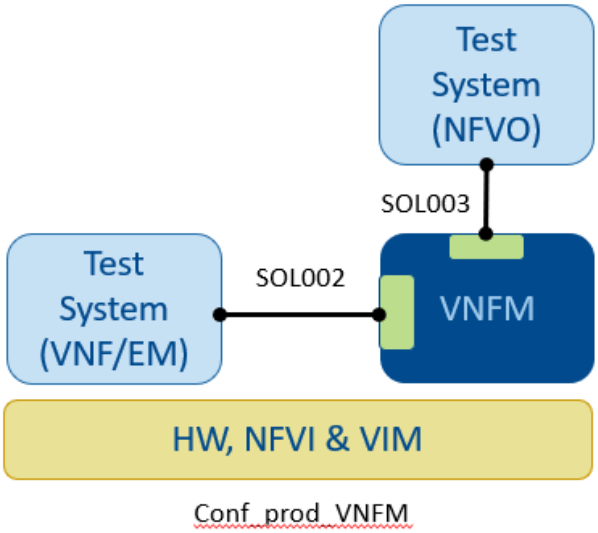

Figure 20: VNFM API Test Configuration

8.2.2.1 SOL002

API	Test IDs	# Test Cases
VNF Fault Management API	TD_API_SOL002_VNF_FM_1..39	39
VNF Life Cycle Management API	TD_API_SOL002_VNF_LCM_1..139	139
VNF Performance Management API	TD_API_SOL002_VNF_PM_1..64	64

Table 66. VNFM SOL002 Test Groups

The complete list of Test Cases can be found in [TST010] Clause 6.3

8.2.2.2 SOL003

API	Test IDs	# Test Cases
VNF Fault Management API	TD_API_SOL003_VNF_FM_1..33	33
VNF Indicator API	TD_API_SOL003_VNF_IND_1..42	42
VNF Performance Management API	TD_API_SOL003_VNF_PM_1..64	64
VNF Life Cycle Management API	TD_API_SOL003_VNF_LCM_1..139	139

Table 77. VNFM SOL003 Test Groups

The complete list of Test Cases can be found in [TST010] Clause 7.3

8.2.3 NFVO

The test configuration as described below was defined to test the interfaces exposed by NFVOs towards the VNFM (SOL003) or OSS/BSS (SOL005). The test system is acting as the VNFM (SOL003 Client) or OSS/BSS (SOL005 client)

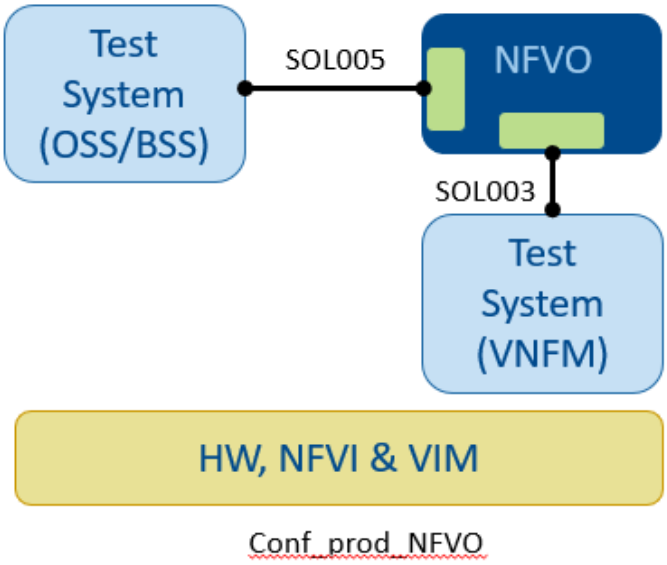

Figure 21: NFVO API Test Configuration

8.2.3.1 SOL003

API	Test IDs	# Test Cases
VNF Package Management API	TD_API_SOL003_VNF_PCKGM_1..75	75
VNF Lifecycle Operation Granting API	TD_API_SOL003_VNF_LCM_GRANT_1..14	14
VNF Virtual Resource Quota Available Notification	TD_API_SOL003_VR_QUOTA_NOTIF_1..19	19

Table 88. NFVO SOL003 Test Groups

The complete list of Test Cases can be found in [TST010] Clause 7.3

8.2.3.2 SOL005

API	Test IDs	# Test Cases
NSD Management API	TD_API_NFVO_SOL005_NSD_MGMT_1..91	91
NS Performance Management API	TD_API_NFVO_SOL005_NS_PM_1..64	64
VNF Package Management API	TD_API_NFVO_SOL005_VNF_PCKGM_1..84	84
NS Fault Management API	TD_API_NFVO_SOL005_NS_FM_1..36	36

Table 99. NFVO SOL005 Test Groups

The complete list of Test Cases can be found in [TST010] Clause 5.3

8.3 MEC Interoperability

8.3.1 MEC Test Configurations

Two (2) different configurations were created for the MEC Track. For both configurations, the list of tests was the same:

The “MEC Basic” configuration which included 2 type of products: a MEC platform and a MEC application.

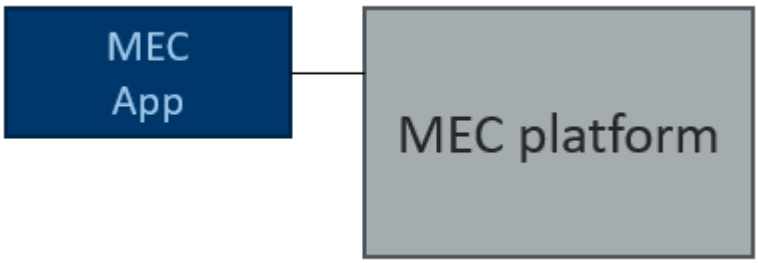

Figure 22: Test configuration “MEC Basic”

The “MEC+NFVI” configuration which included a NFVI platform, a MEC platform and a MEC application.

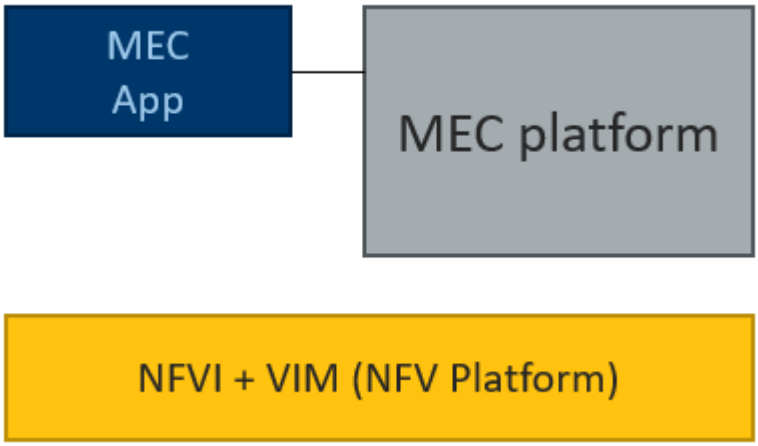

Figure 23: Test configuration “MEC + NFVI”

8.3.2 MEC Test Descriptions

8.3.2.1 Onboarding applications

Below is the list of Test Descriptions related to application’s onboarding within a MEC Platform:

Test ID	Description
TD_MEC_PROV_IMG	Onboard application image to the MEC platform
TD_MEC_INSTANCE_START	MEC Platform can start a new instance of the application to be executed in MEC location
TD_MEC_INSTANCE_STOP	Provide status of running application instance

TD_MEC_INSTANCE_STATUS	Stop the running application to be removed from the MEC platform

Table 20. MEC Application On-boarding TDs

8.3.2.2 MEC Routing capabilities

Below is the list of Test Descriptions related to routing traffic within an edge computing environment through the MEC Platform:

Test ID	Description
TD_MEC_ROUTE_TO_LOCAL_GENERIC	Steer traffic from test device locally to edge application
TD_MEC_ROUTE_TO_LOCAL_WEB	Steer web traffic from test device locally to edge application
TD_MEC_ROUTE_TO_EXTERNAL	Steer traffic from test device externally to external application
TD_MEC_ROUTE_CONTROL	Control VNF accessibility from MEC platform allowing/preventing access to edge application

Table 21. MEC Routing Capabilities TDs

8.3.2.3 Exposure & usage of ETSI MEC API

Below is the list of Test Descriptions related to ETSI MEC API:

Test ID	Description
TD_MEC_API_DISCOVERY	Discover available service through API request to MEC platform
TD_MEC_API_USAGE	Use a specific service through MEC platform API

Table 22 MEC APIs TDs

9 Results

9.1 NFV Interoperability

9.1.1 Overall Results

During the Plugtests, a total of 65 NFV interoperability Test Sessions were run with different combinations of the Functions Under Test (FUTs) in scope: VNFs, MANOs and VIM&NFVI.

The following sections provide an overview of the reported results: overall, per test group, pert tests case. To facilitate the analysis, results are presented as follows:

Result	Meaning
OK	Test Case run. Interoperability (or API test) successfully achieved.
NO	Test Case run. Interoperability (or API test) not achieved.
NA	Not Applicable: Feature not supported by one or more Functions Under Test
Run	Total number of Test Cases Run = OK + NO
Total	Total number of Test Cases = OK + NO + NA = Run + Not Run

Table 23: Results Interpretation

Note that the tests cases for which no result was reported (i.e. when the test session run out of time) are not considered in the Total Results

The table below provides the overall results (aggregated data) for all the test cases run during the NFV Interoperability Test Sessions, from all participating companies:

Overall Results	Number of Test Sessions	Interoperability (TCs Run)		TCs Not Run	TCs Totals	
		OK	NO	NA	Run	Total
	65	493	90	557	583	1.140

Table 24: NFV IOP Overall Results

During each Test Session, depending on the targeted configuration and features to be tested, a different number of test cases were offered to the involved participants.

The interoperability test plans included 255 test cases, organised in different groups as described in clause 8.1. Through the 65 Test Sessions run, a total of 1140 Test Results were reported. This figure includes both the executed and non-executed test cases. Overall, a total of 583 individual test cases were executed and results (OK or NO as per table above) reported for them.

	Interoperability		Not Executed	Totals		Totals			
	OK	NO	N/A	Run	Results	% Run	% OK	% NO	% N/A
Single Vendor NS	225	57	248	282	530	53,21%	79,79%	20,21%	46,79%
Multi Vendor NS	197	29	208	226	434	52,07%	87,17%	12,83%	47,93%
Specific VNFM	38	2	66	40	106	37,74%	95,00%	5,00%	62,26%
Auto LCM Validation	33	2	35	35	70	50,00%	94,29%	5,71%	50,00%
TOTAL	493	90	557	583	1.140	51,14%	84,56%	15,44%	48,86%

Table 25. NFV IOP Overall results per Group

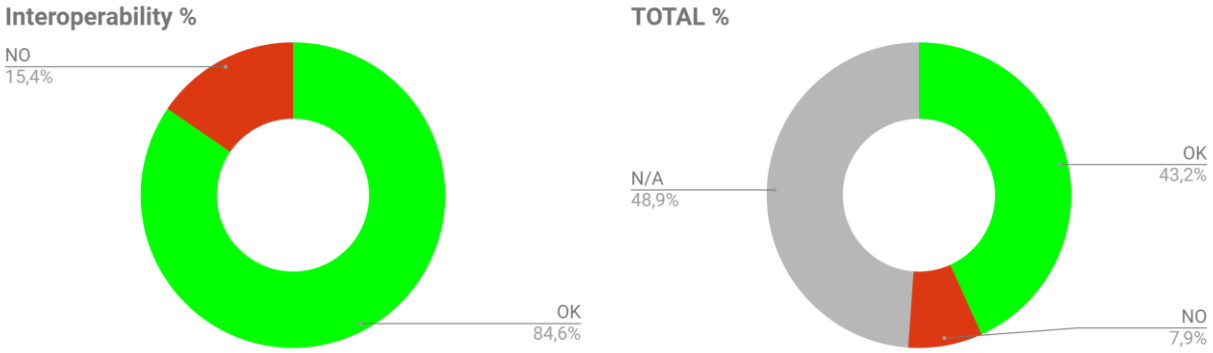

Figure 24. NFV IOP Overall results (%)

The next clauses present more detailed results per test group and test cases and will allow to identify the areas and features with higher execution and interoperability rates.

9.1.2 Results per Group

9.1.2.1 Single Vendor NS

This configuration was offered for pre-testing purposes and to new Plugtests participants, in order to test Single VNF Networks Services before focusing on more complex multi-vendor NS.

	Interoperability		Not Executed	Totals		Totals			
	OK	NO	NA	Run	Results	% Run	% OK	% NO	% NA
SV_ONBOARD	54	1	2	55	57	96,49%	98,18%	1,82%	3,51%
SV_INSTANTIATE	25	2	1	27	28	96,43%	92,59%	7,41%	3,57%
SV_SCALE_NS_MANUAL	24	10	16	34	50	68,00%	70,59%	29,41%	32,00%
SV_SCALE_NS_VNF_IND	4	16	52	20	72	27,78%	20,00%	80,00%	72,22%
SV_SCALE_NS_VIM_METRIC	0	0	10	0	10	0,00%	0,00%	0,00%	100,00%
SV_SCALE_VNF_MANUAL	6	2	2	8	10	80,00%	75,00%	25,00%	20,00%
SV_SCALE_VNF_VNF_IND	2	0	10	2	12	16,67%	100,00%	0,00%	83,33%
SV_SCALE_VNF_VIM_METRIC	0	0	6	0	6	0,00%	0,00%	0,00%	100,00%
SV_SCALE_NS_TO_LEVEL_MANUAL	4	4	8	8	16	50,00%	50,00%	50,00%	50,00%
SV_SCALE_VNF_EM	0	0	6	0	6	0,00%	0,00%	0,00%	100,00%
SV_SCALE_NS_TO_LEVEL_VNF_IND	0	0	7	0	7	0,00%	0,00%	0,00%	100,00%
SV_SCALE_NS_TO_LEVEL_VIM_METRIC	0	0	4	0	4	0,00%	0,00%	0,00%	100,00%
SV_SCALE_VNF_TO_LEVEL_MANUAL	1	0	1	1	2	50,00%	100,00%	0,00%	50,00%
SV_SCALE_VNF_TO_LEVEL_VNF_IND	0	0	2	0	2	0,00%	0,00%	0,00%	100,00%
SV_SCALE_VNF_TO_LEVEL_VIM_METRIC	0	0	3	0	3	0,00%	0,00%	0,00%	100,00%
SV_SCALE_VNF_TO_LEVEL_EM	0	3	13	3	16	18,75%	0,00%	100,00%	81,25%
SV_UPDATE_VNF	18	10	16	28	44	63,64%	64,29%	35,71%	36,36%

SV_FM_SUBSCRIPTION	4	0	14	4	18	22,22%	100,00%	0,00%	77,78%
SV_FM_ALARMS	0	0	18	0	18	0,00%	0,00%	0,00%	100,00%
SV_PM_SUBSCRIPTION	3	0	18	3	21	14,29%	100,00%	0,00%	85,71%
SV_PM_JOB	0	0	15	0	15	0,00%	0,00%	0,00%	100,00%
SV_PM_NOTIFICATION	2	0	12	2	14	14,29%	100,00%	0,00%	85,71%
SV_VNF_IND	0	6	9	6	15	40,00%	0,00%	100,00%	60,00%
SV_TERMINATE	27	1	1	28	29	96,55%	96,43%	3,57%	3,45%
SV_DELETE	51	2	2	53	55	96,36%	96,23%	3,77%	3,64%
TOTAL	225	57	248	282	530	53,21%	79,79%	20,21%	46,79%

Table 26. Results per Single Vendor NS Sub-Group

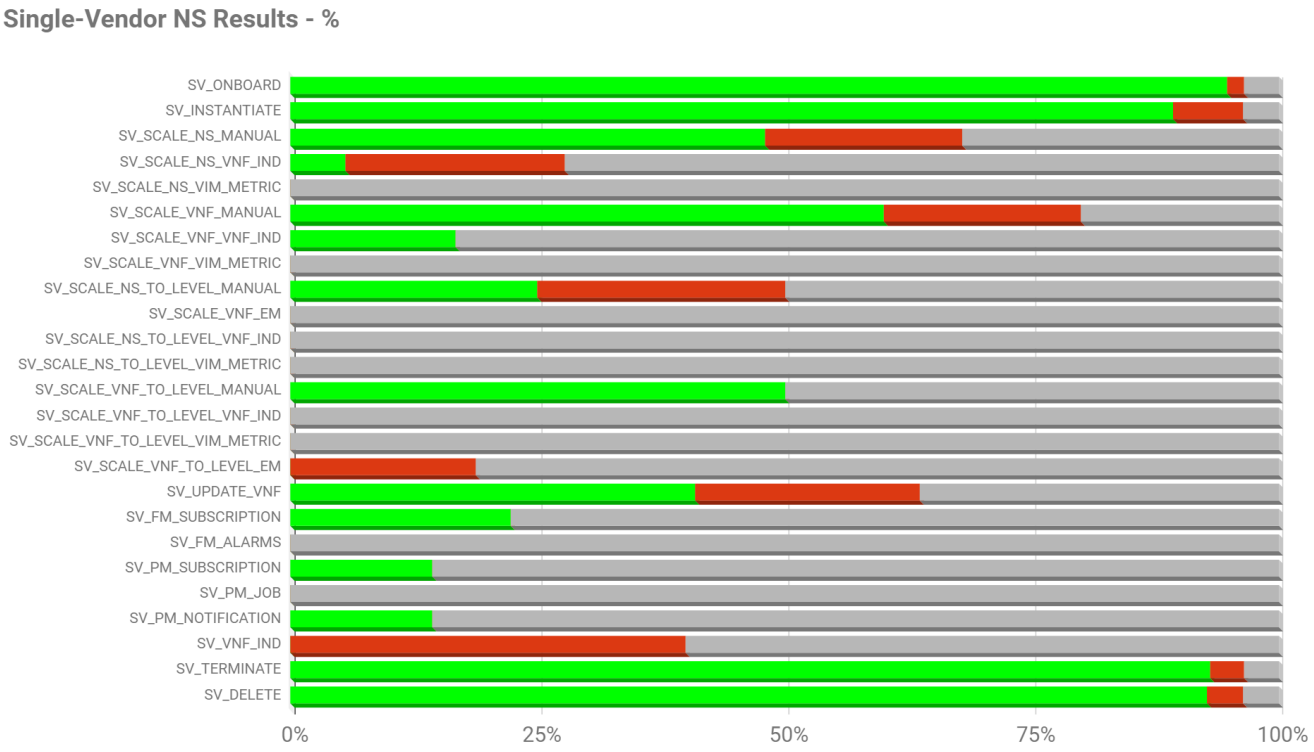

Figure 25. Results per Single Vendor NS Sub-Group - %

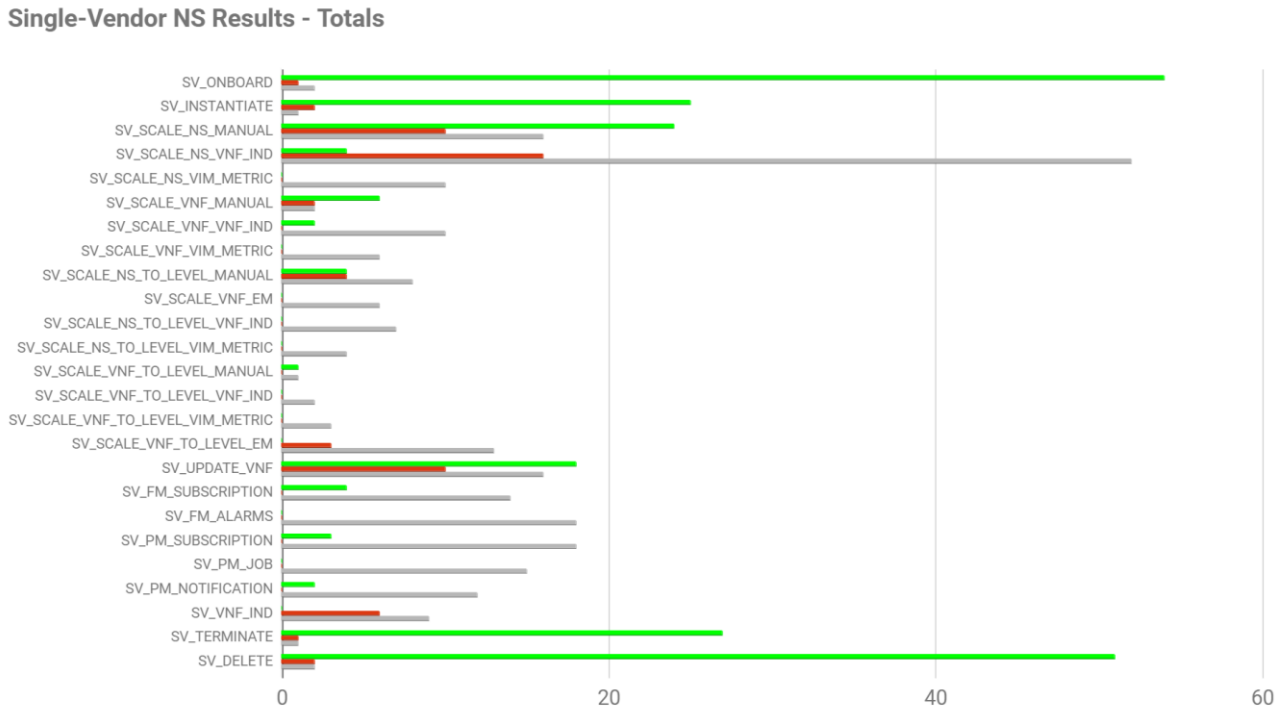

Figure 26. Results per Single Vendor NS Sub Group - Totals

9.1.2.2 Multi Vendor NS

The table and figure below provide an overview of the results per Multi-Vendor NS group. Overall, 24 Multi-Vendor NS test sessions were run.

	Interoperability		Not Executed	Totals		Totals			
	OK	NO	NA	Run	Results	% Run	% OK	% NO	% NA
MV_ONBOARD	46	0	0	46	46	100,00%	100,00%	0,00%	0,00%
MV_INSTANTIATE	22	0	0	22	22	100,00%	100,00%	0,00%	0,00%
MV_SCALE_NS_MANUAL	14	2	8	16	24	66,67%	87,50%	12,50%	33,33%
MV_SCALE_NS_VNF_IND	2	4	26	6	32	18,75%	33,33%	66,67%	81,25%
MV_SCALE_NS_VIM_METRIC	4	0	8	4	12	33,33%	100,00%	0,00%	66,67%
MV_SCALE_VNF_MANUAL	5	2	8	7	15	46,67%	71,43%	28,57%	53,33%
MV_SCALE_VNF_VNF_IND	4	0	8	4	12	33,33%	100,00%	0,00%	66,67%
MV_SCALE_VNF_VIM_METRIC	0	0	6	0	6	0,00%	0,00%	0,00%	100,00%
MV_SCALE_VNF_EM	0	0	6	0	6	0,00%	0,00%	0,00%	100,00%
MV_SCALE_NS_TO_LEVEL_MANUAL	1	0	0	1	1	100,00%	100,00%	0,00%	0,00%
MV_SCALE_NS_TO_LEVEL_VNF_IND	0	0	1	0	1	0,00%	0,00%	0,00%	100,00%
MV_SCALE_NS_TO_LEVEL_VIM_METRIC	0	0	1	0	1	0,00%	0,00%	0,00%	100,00%
MV_SCALE_VNF_TO_LEVEL_MANUAL	1	0	0	1	1	100,00%	100,00%	0,00%	0,00%
MV_SCALE_VNF_TO_LEVEL_VNF_IND	1	0	1	1	2	50,00%	100,00%	0,00%	50,00%
MV_SCALE_VNF_TO_LEVEL_VIM_METRIC	0	0	2	0	2	0,00%	0,00%	0,00%	100,00%

MV_SCALE_VNF_TO_LEVEL_EM	0	0	2	0	2	0,00%	0,00%	0,00%	100,00%
MV_UPDATE_VNF	14	4	8	18	26	69,23%	77,78%	22,22%	30,77%
MV_FM_SUBSCRIPTION	4	4	18	8	26	30,77%	50,00%	50,00%	69,23%
MV_FM_ALARMS	0	6	27	6	33	18,18%	0,00%	100,00%	81,82%
MV_PM_JOB	7	6	27	13	40	32,50%	53,85%	46,15%	67,50%
MV_PM_SUBSCRIPTION	3	0	21	3	24	12,50%	100,00%	0,00%	87,50%
MV_PM_NOTIFICATION	3	0	13	3	16	18,75%	100,00%	0,00%	81,25%
MV_VNF_IND	0	1	17	1	18	5,56%	0,00%	100,00%	94,44%
MV_TERMINATE	22	0	0	22	22	100,00%	100,00%	0,00%	0,00%
MV_DELETE	44	0	0	44	44	100,00%	100,00%	0,00%	0,00%
TOTAL	197	29	208	226	434	52,07%	87,17%	12,83%	47,93%

Table 27. Results per Multi Vendor NS Sub-Group

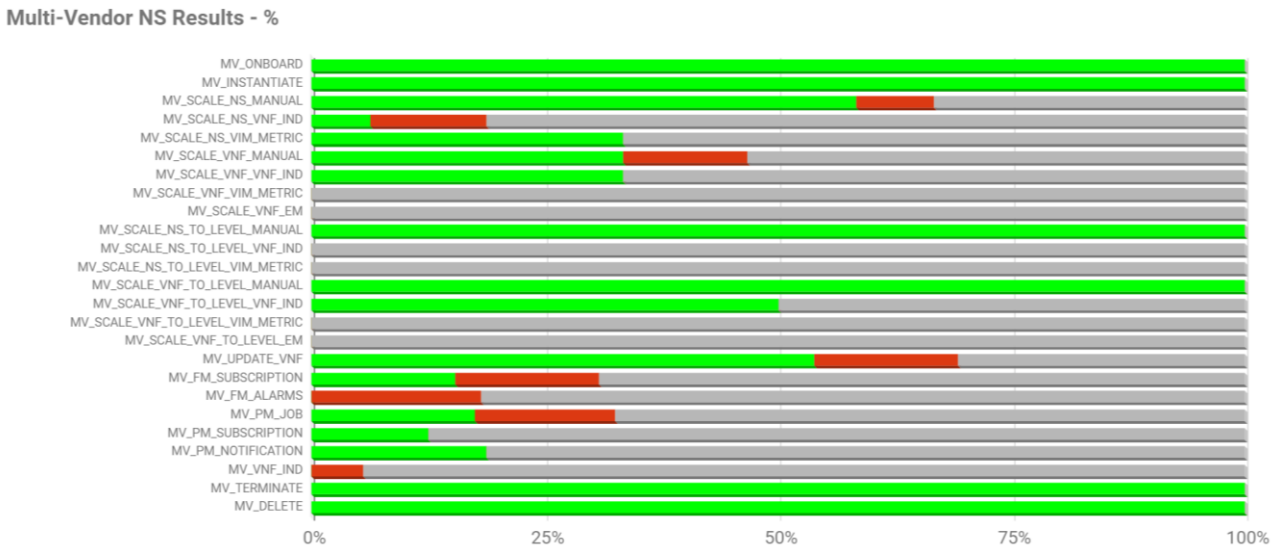

Figure 27. Results per Multi Vendor NS Sub-Group - %

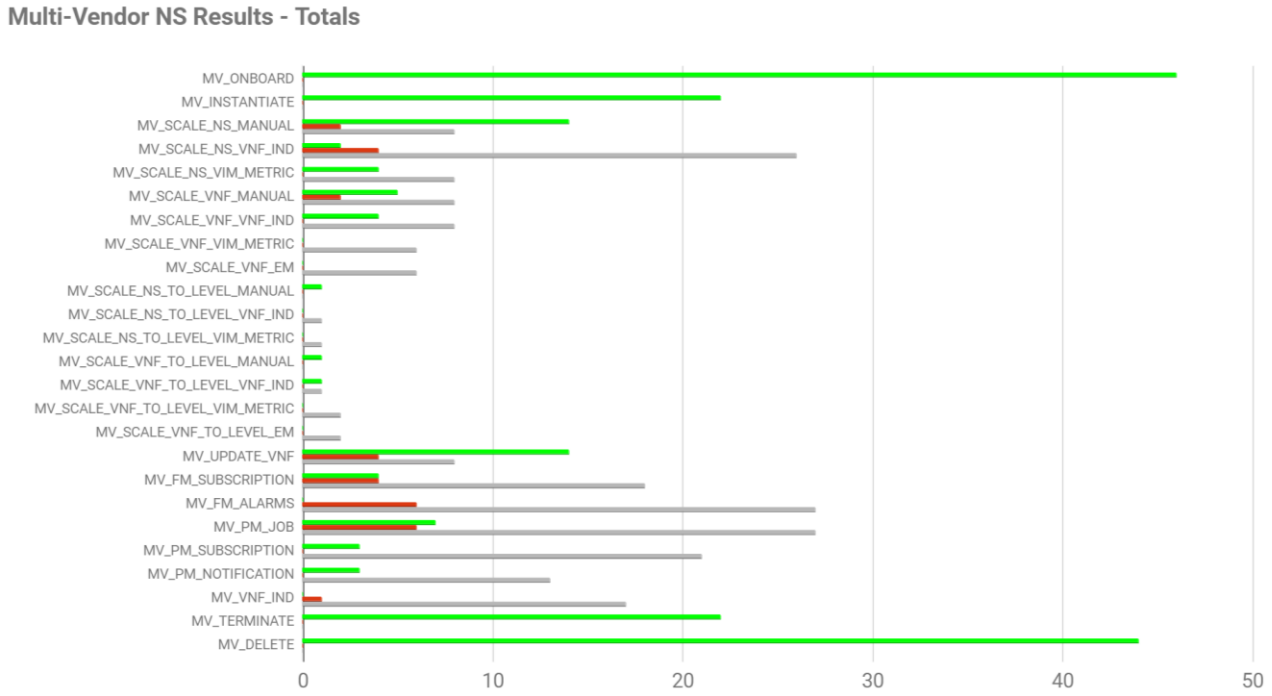

Figure 28. Results per Multi Vendor NS Sub-Group - Totals

9.1.2.3 Specific VNFM

This configuration was offered to test VNFMs and NFVOs from different providers. While this configuration has been offered since the 2nd NFV Plugtests, this was the first event where it was tested. Overall 7 Test Sessions involving different combinations of FUTs were run, and results reported as follows:

	Interoperability		Not Executed	Totals		Totals			
	OK	NO	NA	Run	Results	% Run	% OK	% NO	% NA
S-VNFM_ONBOARD	11	0	1	11	12	91,67%	100,00%	0,00%	8,33%
S-VNFM_INSTANTIATE	6	1	0	7	7	100,00%	85,71%	14,29%	0,00%
S-VNFM_SCALE_NS_MANUAL	0	0	12	0	12	0,00%	0,00%	0,00%	100,00%
S-VNFM_SCALE_NS_VNF_IND	0	0	20	0	20	0,00%	0,00%	0,00%	100,00%
S-VNFM_SCALE_NS_VIM_METRIC	0	0	4	0	4	0,00%	0,00%	0,00%	100,00%
S-VNFM_SCALE_VNF_MANUAL	6	0	0	6	6	100,00%	100,00%	0,00%	0,00%
S-VNFM_SCALE_VNF_VNF_IND	0	0	4	0	4	0,00%	0,00%	0,00%	100,00%
S-VNFM_SCALE_VNF_VIM_METRIC	0	0	2	0	2	0,00%	0,00%	0,00%	100,00%
S-VNFM_SCALE_VNF_EM	0	0	2	0	2	0,00%	0,00%	0,00%	100,00%
S-VNFM_SCALE_NS_TO_LEVEL_MANUAL	0	0	0	0	0	0,00%	0,00%	0,00%	0,00%
S-VNFM_SCALE_NS_TO_LEVEL_VNF_IND	0	0	1	0	1	0,00%	0,00%	0,00%	100,00%
S-VNFM_SCALE_NS_TO_LEVEL_VIM_METRIC	0	0	2	0	2	0,00%	0,00%	0,00%	100,00%
S-VNFM_SCALE_VNF_TO_LEVEL_MANUAL	0	0	0	0	0	0,00%	0,00%	0,00%	0,00%

S-VNFM_SCALE_VNF_TO_LEVEL_VNF_IND	0	0	1	0	1	0,00%	0,00%	0,00%	100,00%
S-VNFM_SCALE_VNF_TO_LEVEL_VIM_METRIC	0	0	1	0	1	0,00%	0,00%	0,00%	100,00%
S-VNFM_SCALE_VNF_TO_LEVEL_EM	0	0	0	0	0	0,00%	0,00%	0,00%	0,00%
S-VNFM_UPDATE_VNF	0	0	12	0	12	0,00%	0,00%	0,00%	100,00%
S-VNFM_FM_SUBSCRIPTION	0	1	3	1	4	25,00%	0,00%	100,00%	75,00%
S-VNFM_FM_ALARMS	0	0	0	0	0	0,00%	0,00%	0,00%	0,00%
S-VNFM_PM_JOB	0	0	0	0	0	0,00%	0,00%	0,00%	0,00%
S-VNFM_PM_SUBSCRIPTION	0	0	0	0	0	0,00%	0,00%	0,00%	0,00%
S-VNFM_PM_NOTIFICATION	0	0	0	0	0	0,00%	0,00%	0,00%	0,00%
S-VNFM_TERMINATE	5	0	0	5	5	100,00%	100,00%	0,00%	0,00%
S-VNFM_DELETE	10	0	1	10	11	90,91%	100,00%	0,00%	9,09%
TOTAL	38	2	66	40	106	37,74%	95,00%	5,00%	62,26%

Table 28. Results per Specific VNFM Sub-Group

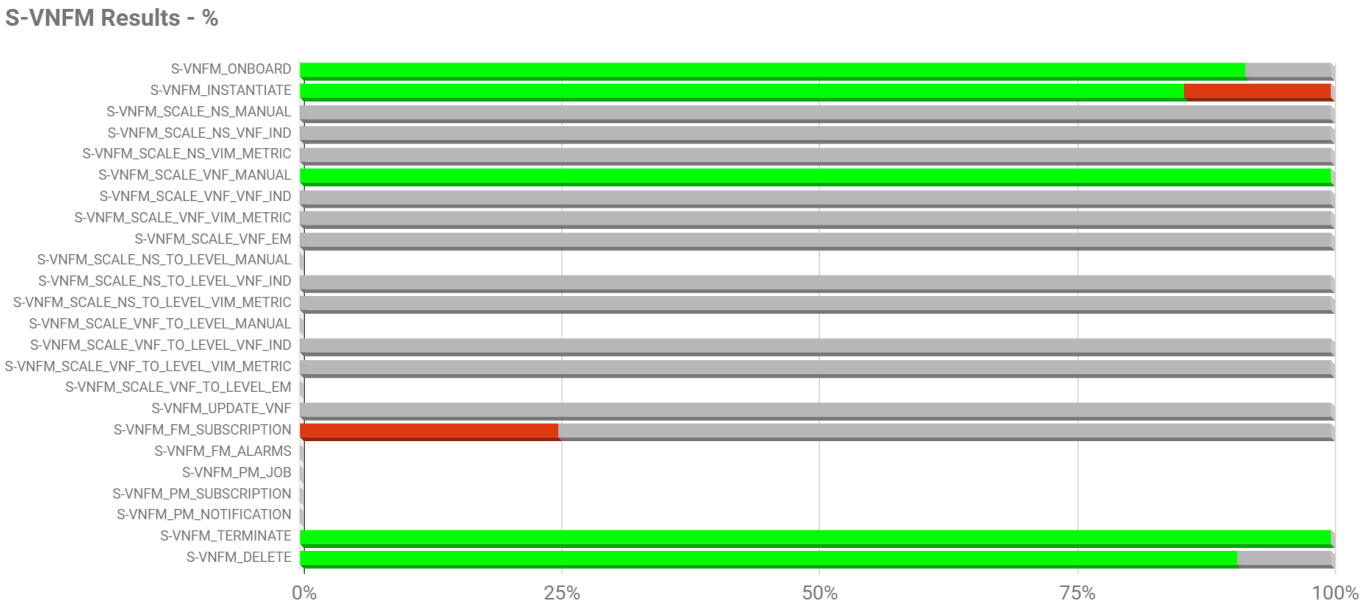

Figure 29. Results per Specific VNFM Sub-Group - %

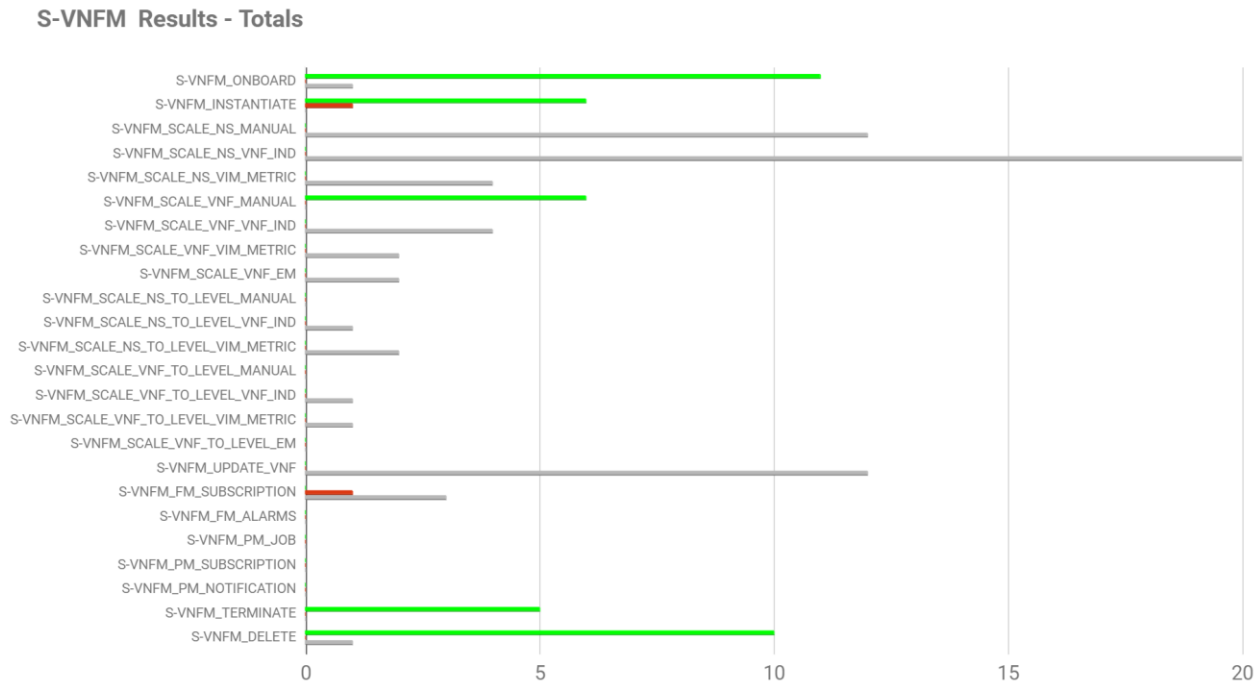

Figure 30. Results per S-VNFM Sub-Group - Totals

9.1.2.4 Auto LCM Validation

The table and figure below provide an overview of the results per Auto LCM Validation group. Overall, 24 Multi-Vendor NS test sessions were run.

	Interoperability		Not Executed	Totals		Totals			
	OK	NO	NA	Run	Results	% Run	% OK	% NO	% NA
AUTOLCM_ONBOARD	7	0	3	7	10	70,00%	100,00%	0,00%	30,00%
AUTOLCM_INSTANTIATE	5	0	0	5	5	100,00%	100,00%	0,00%	0,00%
AUTOLCM_SCALE_NS_MANUAL	4	2	2	6	8	75,00%	66,67%	33,33%	25,00%
AUTOLCM_SCALE_NS_VNF_IND	0	0	12	0	12	0,00%	0,00%	0,00%	100,00%
AUTOLCM_SCALE_NS_VIM_METRIC	2	0	2	2	4	50,00%	100,00%	0,00%	50,00%
AUTOLCM_SCALE_VNF_MANUAL	2	0	0	2	2	100,00%	100,00%	0,00%	0,00%
AUTOLCM_SCALE_VNF_VNF_IND	0	0	0	0	0	0,00%	0,00%	0,00%	0,00%
AUTOLCM_SCALE_VNF_VIM_METRIC	0	0	0	0	0	0,00%	0,00%	0,00%	0,00%
AUTOLCM_UPDATE_VNF	0	0	4	0	4	0,00%	0,00%	0,00%	100,00%
AUTOLCM_FM_SUBSCRIPTION	0	0	4	0	4	0,00%	0,00%	0,00%	100,00%
AUTOLCM_FM_ALARMS	0	0	6	0	6	0,00%	0,00%	0,00%	100,00%
AUTOLCM_TERMINATE	5	0	0	5	5	100,00%	100,00%	0,00%	0,00%
AUTOLCM_DELETE	8	0	2	8	10	80,00%	100,00%	0,00%	20,00%
TOTAL	33	2	35	35	70	50,00%	94,29%	5,71%	50,00%

Table 29. Results per Auto LCM Validation Sub-Groups

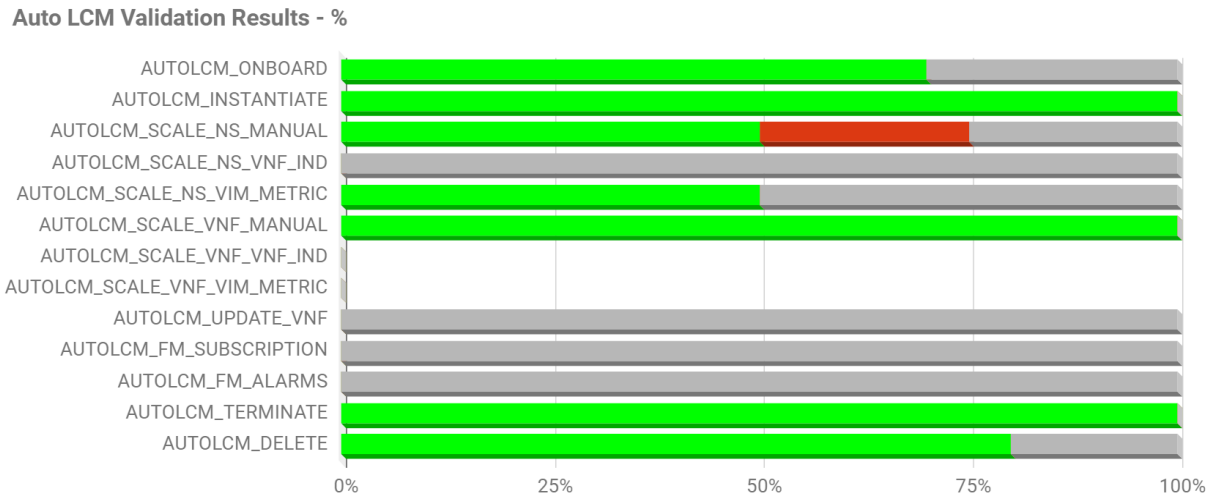

Figure 31. Results per Auto LCM Validation Sub-Group - %

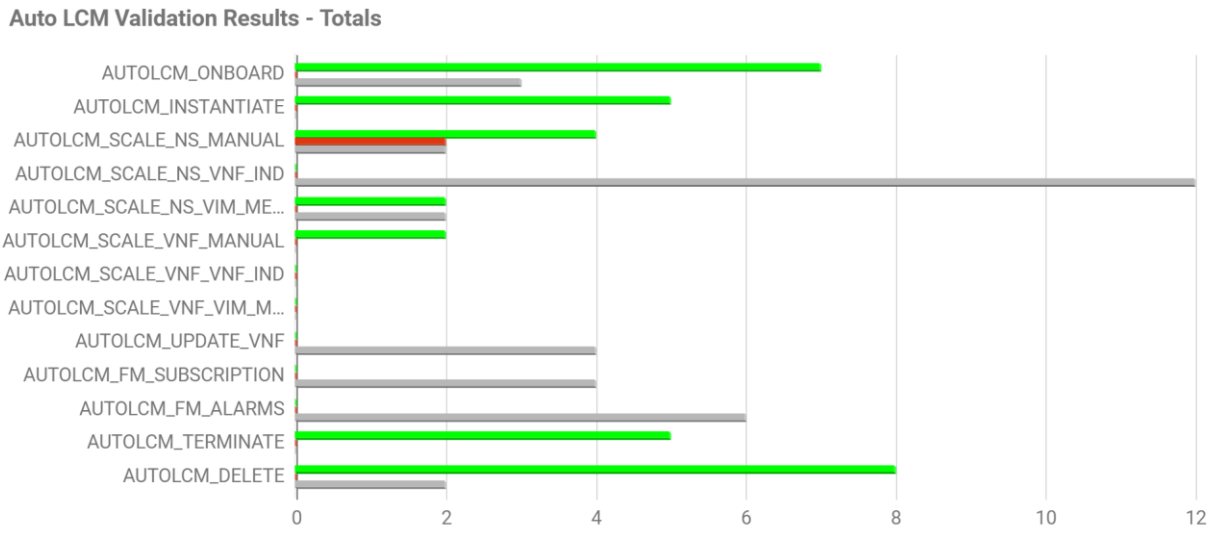

Figure 32. Results per Automatic LCM Validation Sub-Group – Totals

9.1.3 Results per Test Case

The full list of interoperability results grouped per Test Case is provided in Annex A.

9.2 NFV API Conformance Results

During the 4th NFV Plugtests event, a total of 13 NFV API Conformance Test Sessions were run. Seven of them were run 2 or more times to incorporate patches in the Test System or the Function Under Test. The API conformance tests were executed for three different Functions Under Test (FUTs): VNFs, VNFMs and NFVOs. A total of 13 FUTs participated to the NFV API Conformance sessions, that is:

- 2 VNFs
- 3 VNFMs

- 8 NFVOs

The table below provides the overall results (aggregated data) for all the NFV API Conformance tests run during the 4th NFV Plugtests, from all participating organisations.

Overall Results	Number of (unique) Test Sessions	API Conformance (TCs Run)		TCs Not Run	TCs Totals	
		OK	NO	NA	Run	Total
	13	673	110	29	783	812

Table 30: NFV API Conformance overall results

During each remote Test Session, depending on the involved FUT and the features to be tested, a different number of test cases were offered to the involved participant.

Overall, the test plan included 1061 NFV API Conformance test cases, organised in different groups as described in clause 8.2. The test plan was the same as the one used for the Remote API Plugtests run a few months before.

With respect to the Remote API Plugtests, new participants took part at the testing sessions, while some used this opportunity to re-run part of the tests already executed to cross-check new features and fixed issues. Through the 13 Test Sessions run, a total of 812 Test Results were reported. This figure includes both the executed and non-executed test cases. Overall, a total of 783 individual NFV API Conformance test cases were run and results reported for them.

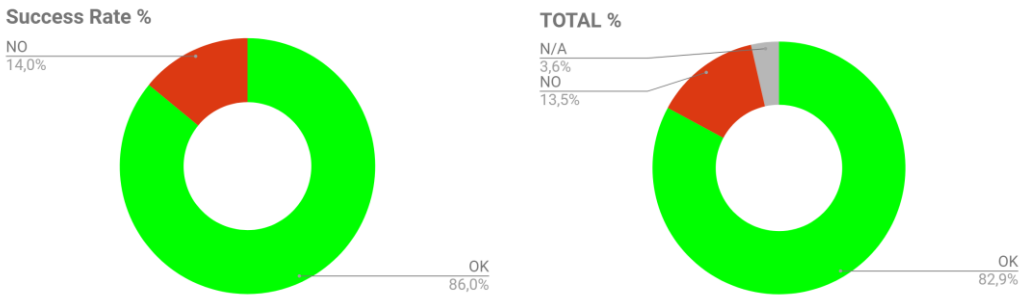

Figure 33. NFVAPI Conformance Overall results (%)

Comparing with previous events, the number of Test Sessions and results reported has significantly increased since previous face to face events were experimental API testing was run. Moreover, even if all the NFV API Conformance Test Sessions were run during the 1-week event, the number of Test Sessions, results and success rates are also higher than the ones reported during the Remote NFV API Plugtests that spanned over 2 months and a half (see [RNFVLU-R]. This progression can be read as a sign of industry adoption of NFV APIs and proves the value of regular (continuous) API Testing.

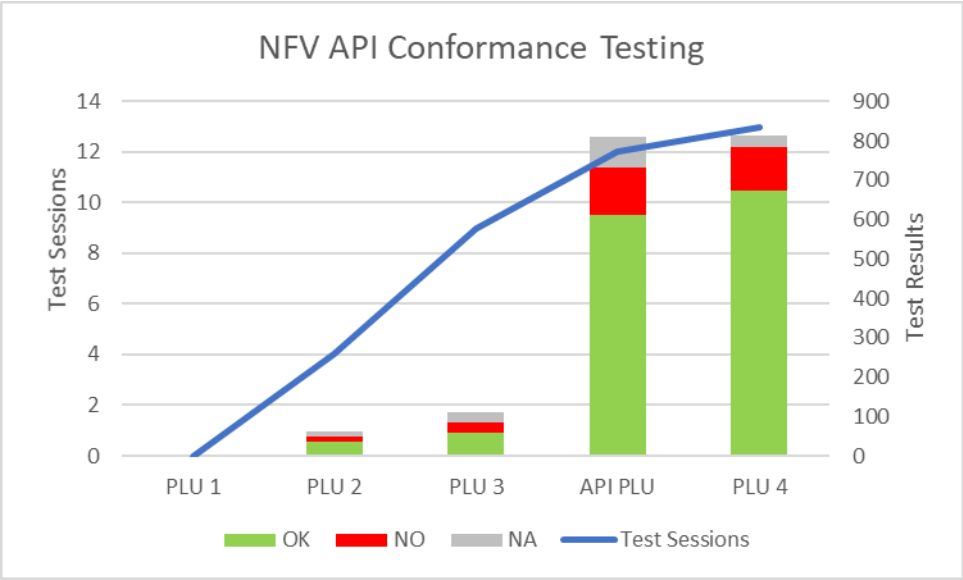

Figure 44. Progress of NFVAPI Conformance Testing

The next clauses present more detailed results per SOL Specification, per FUT type, per SUT configuration and per test group and will allow to identify the areas and APIs with higher execution and conformance rates.

9.2.1 Results per SOL Specification

The tables and figures below provide an overview of the results for the API conformance per SOL specifications, i.e. SOL002, SOL003 and SOL003. Overall the SOL005 and SOL003 APIs have been those with the higher number of Test Cases run and SOL002 had the higher execution and success rate. With respect to the Remote API Plugtests, a highest number of SOL005 API tests have been run, and the overall results have improved.

	API Conformance		Not Executed	Totals		Totals			
	OK	NO	N/A	Run	Results	% Run	% OK	% NO	% N/A
SOL002	67	9	0	76	76	100,00%	88,16%	11,84%	0,00%
SOL003	341	50	21	391	412	94,90%	87,21%	12,79%	5,10%
SOL005	265	51	8	316	324	97,53%	83,86%	16,14%	2,47%
TOTAL	673	110	29	783	812	96,43%	85,95%	14,05%	3,57%

Table 31: Test Results summary per-SOL Specification

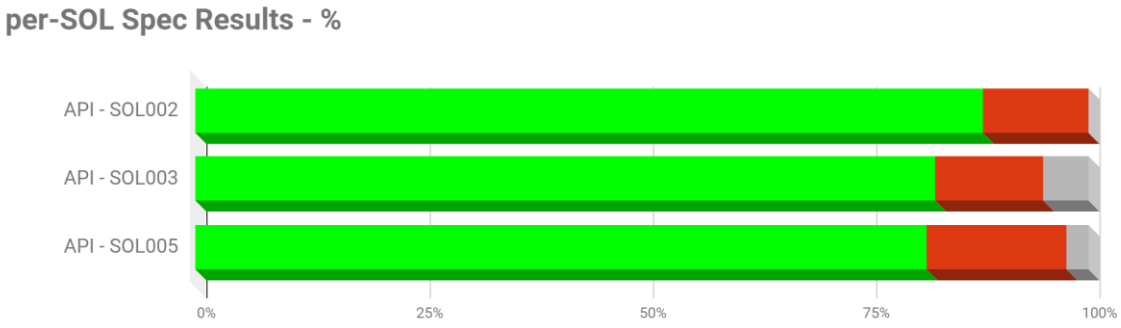

Figure 35. Test results per-SOL Specification - %

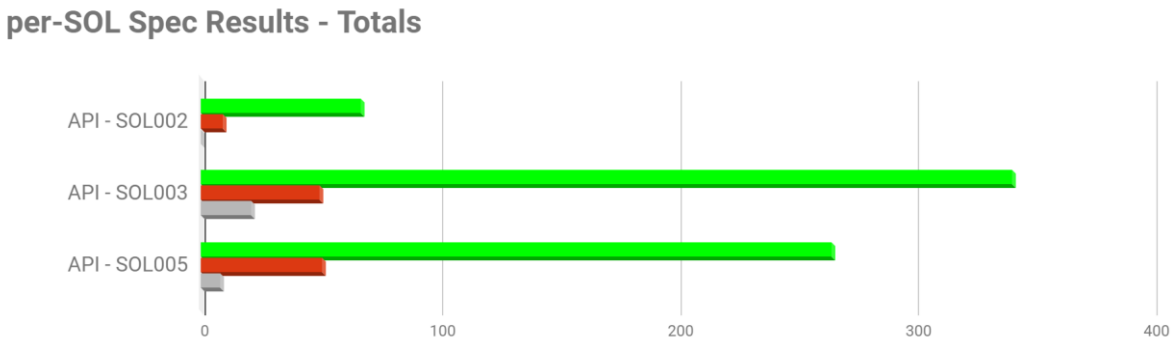

Figure 36. Test Results per-SOL specification -Totals

9.2.2 Results per FUT Type

The tables and figures below summarize the results for the API conformance tests per type of FUT involved in the API conformance test sessions, i.e. VNF, VNFM and NFVO. Overall, VNFM related tests produced excellent results in terms of success rate, while for VNFs a 100% execution rate was achieved. NFVO FUTs were those clearly most tested, and in terms of re related tests also achieved good results.

Overall, in terms of per FUT type split, the results can be considered as similar with respect to those of the Remote API Plugtests.

	API Conformance		Not Executed	Totals	Totals				
	OK	NO	N/A	Run	Results	% Run	% OK	% NO	% N/A
VNF	67	9	0	76	76	100,00%	88,16%	11,84%	0,00%
VNFM	177	11	5	188	193	97,41%	94,15%	5,85%	2,59%
NFVO	429	90	24	519	543	95,58%	82,66%	17,34%	4,42%
TOTAL	673	110	29	783	812	96,43%	85,95%	14,05%	3,57%

Table 32: Test Results summary per-FUT type

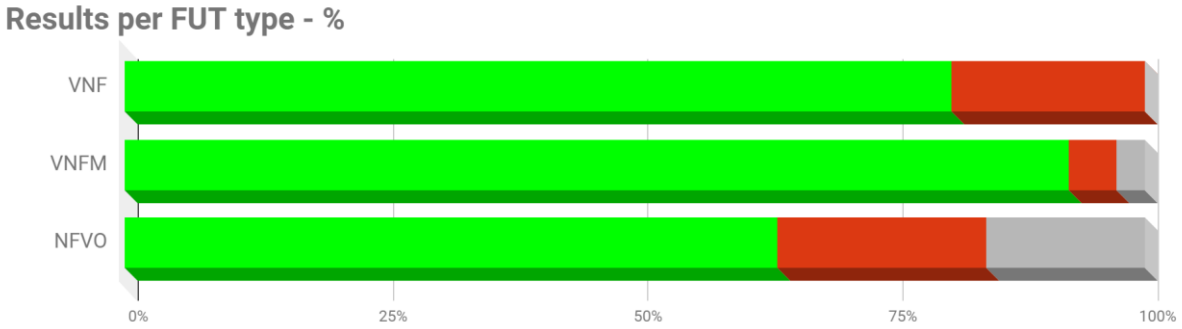

Figure 37. Test Results per- FUT type - %

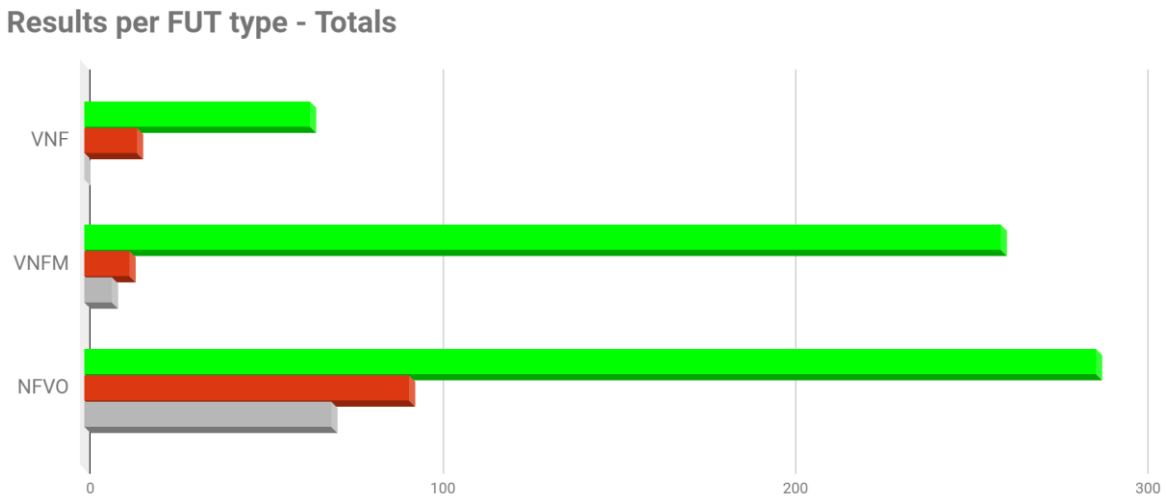

Figure 38. Test Results per-FUT type - Totals

9.2.3 Results per API

The following clauses provide tables and figures which summarize the results for the NFV API conformance tests for the different APIs in each test configuration.

9.2.3.1 VNF - SOL002

	API Conformance		Not Executed	Totals		Totals			
	OK	NO	N/A	Run	Results	% Run	% OK	% NO	% N/A
VNF Configuration API	12	0	0	12	12	100,00%	100,00%	0,00%	0,00%
VNF Indicator API	55	9	0	64	64	100,00%	86,61%	13,39%	0,00%
TOTAL	67	9	0	76	76	100,00%	88,16%	11,84%	0,00%

Table 33: VNF SOL002 test results summary

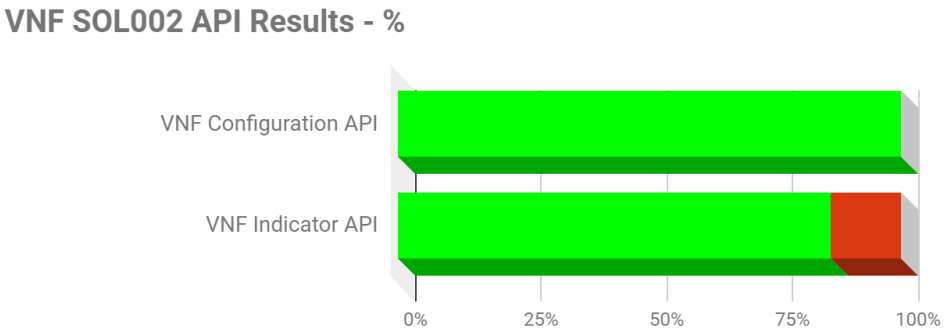

Figure 39. VNF SOL002 test results - %

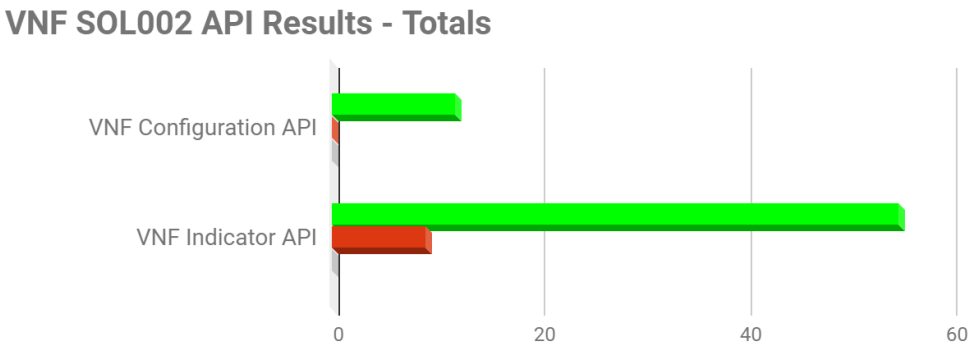

Figure 40. VNF SOL002 test results - Totals

9.2.3.2 VNFM - SOL002

No results were reported for VNFM SOL002 test cases.

9.2.3.3 VNFM - SOL003

	API Conformance		Not Executed	Totals		Totals			
	OK	NO	N/A	Run	Results	% Run	% OK	% NO	% N/A
VNF Lifecycle Management API	155	9	4	164	168	97,62%	94,51%	5,49%	2,38%
VNF Performance Management API	15	2	1	17	18	94,44%	88,24%	11,76%	5,56%
VNF Fault Management API	7	0	0	7	7	100,00%	100,00%	0,00%	0,00%
VNF Indicator API	0	0	0	0	0	0,00%	0,00%	0,00%	0,00%
TOTAL	177	11	5	188	193	97,41%	94,15%	5,85%	2,59%

Table 34: SOL003 VNFM test results summary

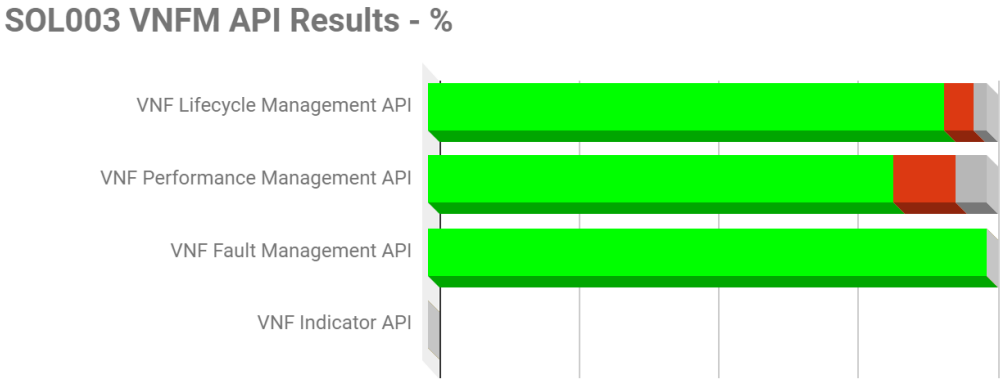

Figure 41. SOL003 VNFM test results - %

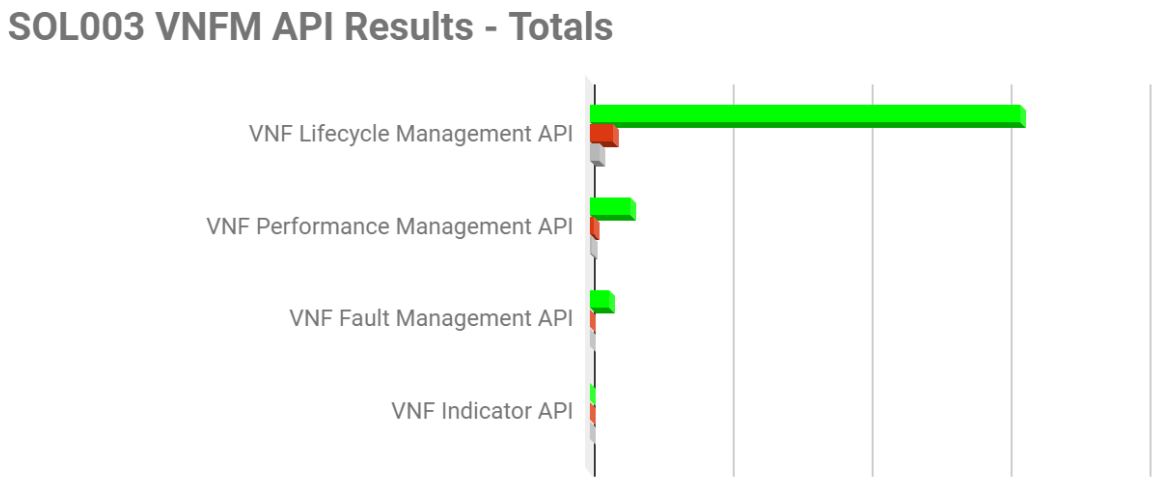

Figure 42. SOL003 VNFM test results - Totals

9.2.3.4 NFVO - SOL003

	API Conformance		Not Executed	Totals		Totals			
	OK	NO	N/A	Run	Results	% Run	% OK	% NO	% N/A
VNF Package Management API	141	34	11	175	186	94,09%	80,57%	19,43%	5,91%
VNF Lifecycle Operation Granting API	23	5	5	28	33	84,85%	82,14%	17,86%	15,15%
Virtualised Resource Quota Notification API	0	0	0	0	0	0,00%	0,00%	0,00%	0,00%
TOTAL	164	39	16	203	219	92,69%	80,79%	19,21%	7,31%

Table 35: NFV SOL003 test results summary

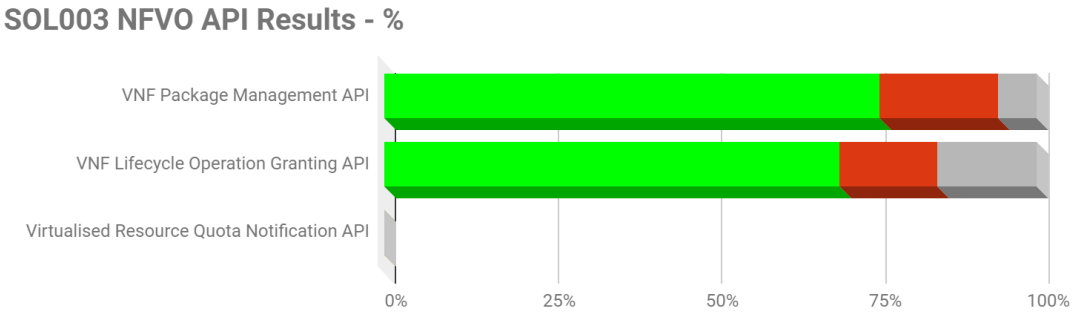

Figure 43. NFVO SOL003 test results - %

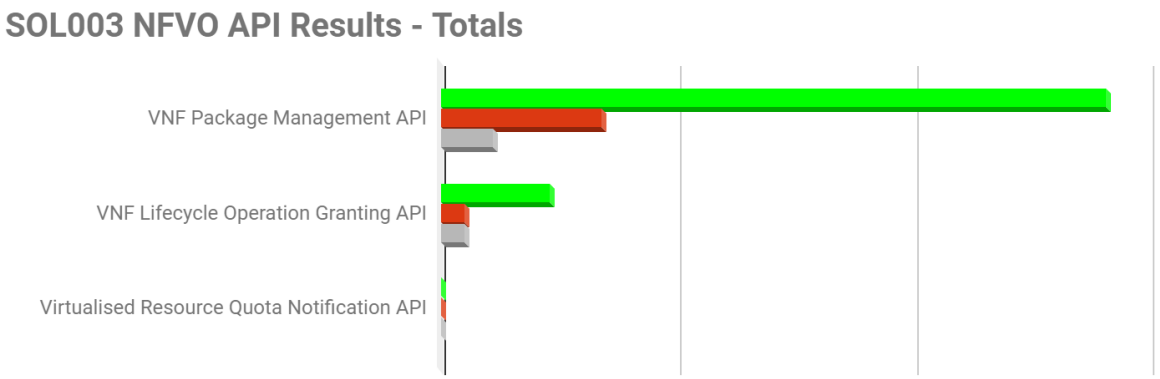

Figure 44. NFV SOL003 test results - Totals

9.2.3.5 NFVO - SOL005

	API Conformance		Not Executed	Totals		Totals			
	OK	NO	N/A	Run	Results	% Run	% OK	% NO	% N/A
NSD Management API	115	30	3	145	148	97,97%	79,31%	20,69%	2,03%
VNF Package Management API	101	18	3	119	122	97,54%	84,87%	15,13%	2,46%
NS Performance Management API	0	0	0	0	0	0,00%	0,00%	0,00%	0,00%
NS Fault Management API	0	0	0	0	0	0,00%	0,00%	0,00%	0,00%
TOTAL	265	51	8	316	324	97,53%	83,86%	16,14%	2,47%

Table 36: NFVO SOL005 test results summary

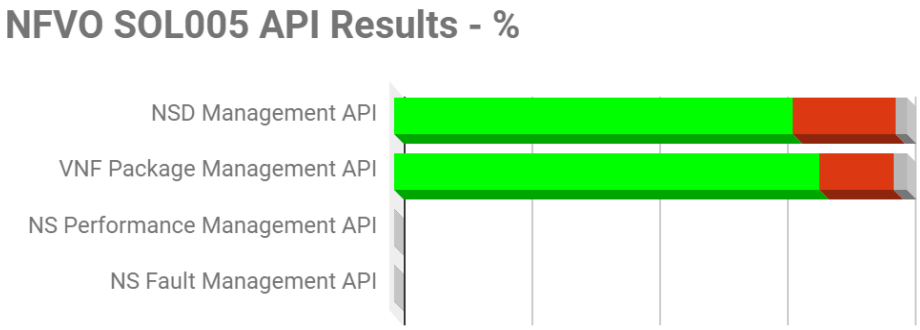

Figure 45. NFVO SOL005 test results %

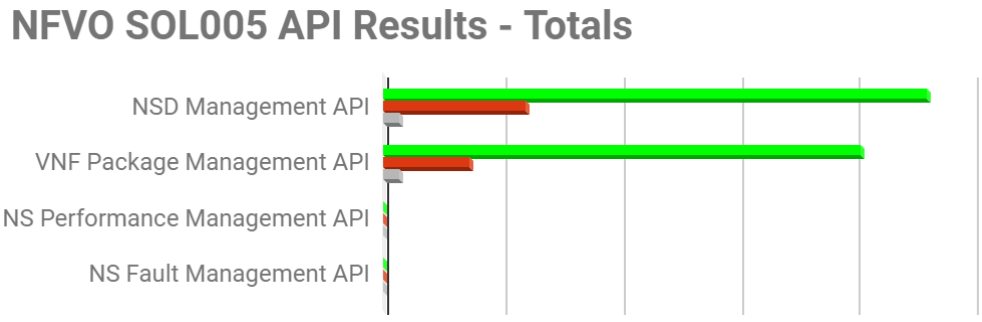

Figure 46. NFVO SOL005 test results - Totals

9.2.4 Results per Test Case

The full list of API Conformance results grouped per Test Case is provided in Annex B.

9.3 MEC Interoperability

9.3.1 Overall Results

As part of the MEC Track a total of 6 Test Sessions were run, combining 8 different Functions Under Test. The MEC interoperability tests were executed for 2 different configurations, one including a MEC App and a MEC Platform (running on its own NFVI) and a second one with a MEC App and a MEC Platform running on an NFVI from a different provider. The subsequent sections report the aggregated testing results.

Overall Results	Number of Test Sessions	Interoperability (TCs Run)		TCs Not Run	TCs Totals	
		OK	NO	NA	Run	Total
	6	44	4	10	48	58

Table 37: MEC IOP Overall Results

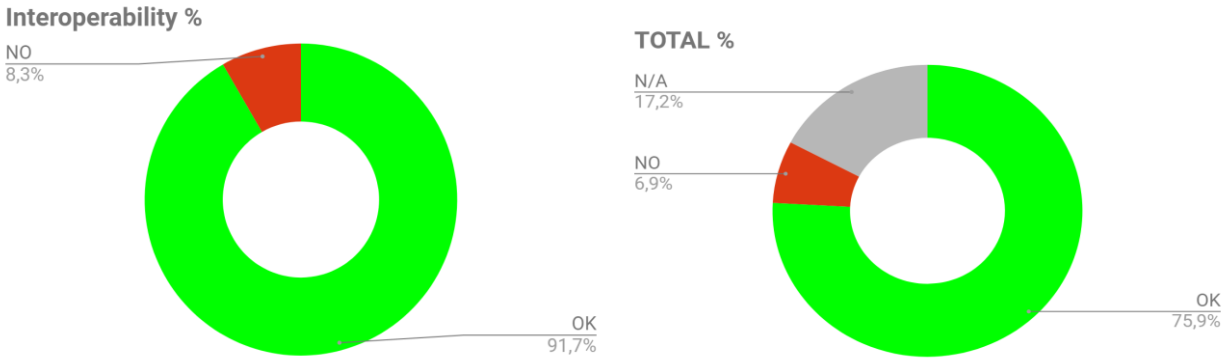

Figure 47. MEC IOP Overall results (%)

The table below provides an overview of the results for each group in the MEC Track.

	Interoperability		Not Executed	Totals		Totals (%)			
	OK	NO	N/A	Run	Results	% Run	% OK	% NO	% N/A
MEC_LIFECYCLE_MANAGEMENT (with NFVI)	12	0	0	12	12	100,00%	100,00%	0,00%	0,00%
MEC_ROUTING (with NFVI)	7	0	3	7	10	70,00%	100,00%	0,00%	30,00%
MEC_API (with NFVI)	4	2	0	6	6	100,00%	66,67%	33,33%	0,00%
MEC_LIFECYCLE_MANAGEMENT	12	0	0	12	12	100,00%	100,00%	0,00%	0,00%
MEC_ROUTING	8	0	4	8	12	66,67%	100,00%	0,00%	33,33%
MEC_API	1	2	3	3	6	50,00%	33,33%	66,67%	50,00%

Table 38: MEC IOP Overall results per group

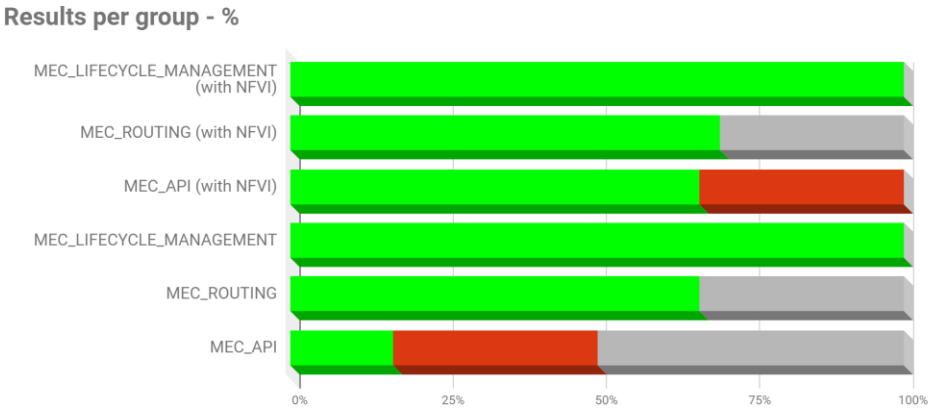

Figure 49. MEC IOP results per group - %

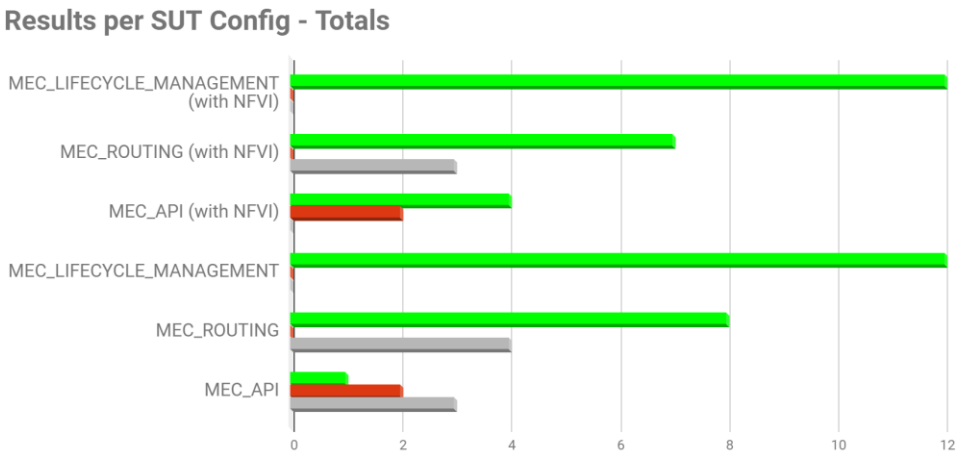

Figure 50. MEC IOP results per group - Totals

9.3.2 Results per Test Case

The Table below provides an overview of the results for each test descriptions in the MEC Track.

	Interoperability		Not Executed		Totals	
	OK	NO	NA	OT	Run	Results
TD_MEC_PROV_IMG	6	0	0	0	6	6
TD_MEC_INSTANCE_START	6	0	0	0	6	6
TD_MEC_INSTANCE_STOP	6	0	0	0	6	6
TD_MEC_ROUTE_TO_LOCAL_GENERIC	4	0	2	0	4	6
TD_MEC_ROUTE_TO_LOCAL_WEB	4	0	2	0	4	6
TD_MEC_ROUTE_TO_EXTERNAL	4	0	2	0	4	6
TD_MEC_ROUTE_CONTROL	3	0	1	0	3	4
TD_MEC_API_DISCOVERY	2	2	2	0	4	6
TD_MEC_API_USAGE	3	2	1	0	5	6
TD_MEC_INSTANCE_STATUS	6	0	0	0	6	6

Table 42: MEC IOP results per TD

10 Plugtests Outcome

During the 4th NFV Plugtests several items were identified as potential issues or inconsistencies and discussed among participants, organisers and some experts from NFV and MEC ISGs. This chapter compiles the highlights of these discussions, the list of reported bugs and some recommendations on NFV and MEC specifications, OpenAPIs, and the test plans. This feedback is submitted to NFV and MEC ISGs for further discussion and eventual resolution in the appropriate working groups.

10.1 Feedback on Base Specifications

10.1.1 NFV Specifications

10.1.1.1 SOL003 - Clarifications on InstantiationVnfRequest

Some doubts were raised on the expected contents of the InstantiationVnfRequest for 2 Use Cases defined in NFV IFA011:

- **Use Case #1: internal CP exposed as external CP of a VNF**

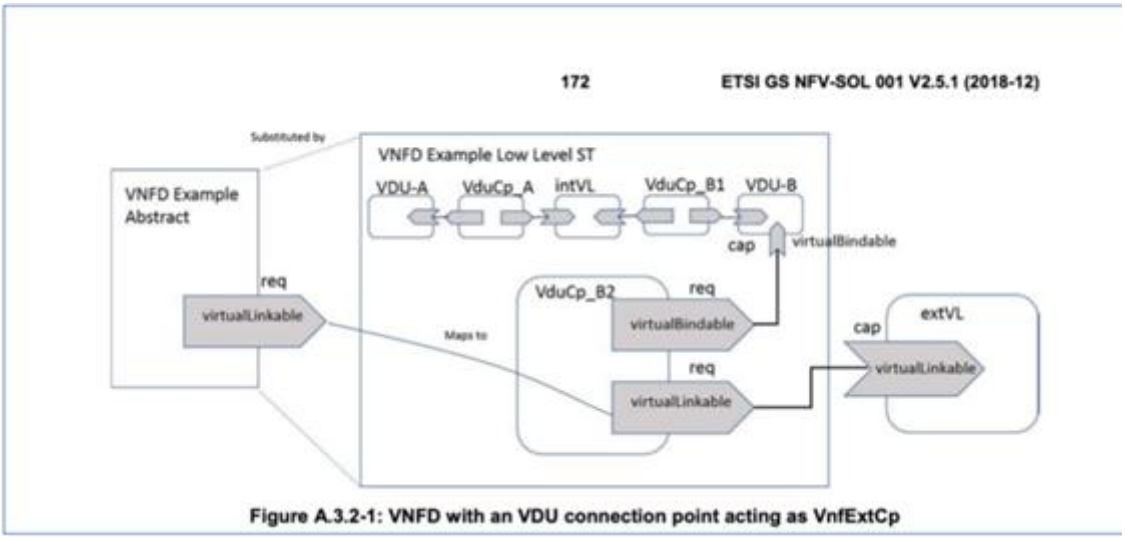

Figure 51.NFV SOL003 Extract (1)

- SOL001 states that in this case the VNFD will not contain explicit definition of a VnfExtCpd node template, but the mapping between the internal CP node template and the external VL is specified in the VNFD service template (though substitution_mappings).
- SOL006 requires the definition of an explicit external CPD associated to the internal one.
- SOL003, in the body of the InstantiateVnfRequest requires the "cpdId" for each external CP. In case of SOL001 it will be the ID of the internal CP node template, in case of SOL006 it will refers to the ID of the external CPD.

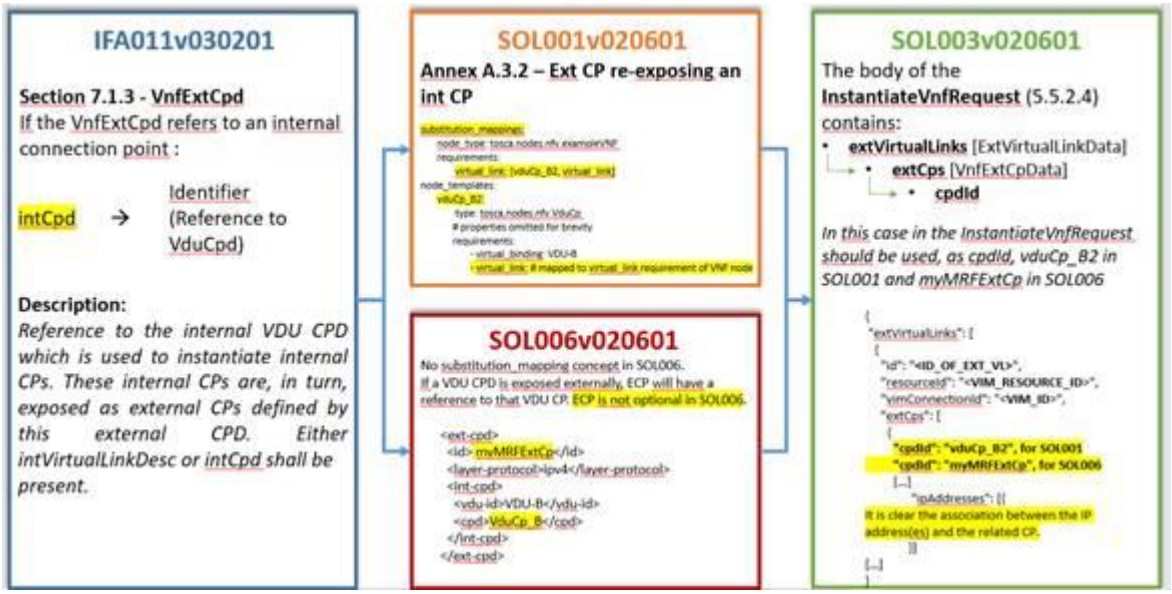

Figure 52. InstatiateRequest through ETSI NFV Specifications (1)

For this Use Case #1 a question was raised on whether it was correct that cpdId was different for SOL001 and SOL006 description of the same VNF. Discussion outlined that whether they are identical or different depends on the naming schemes used by the VNF providers. With the examples in above they are different because these Ids convey information about the type of CPD. However, it could be decided that CPDs that are externally visible are named in a neutral way. For example, the CPD Id for sending/receiving SIP traffic from/to a PDN-GW could be named SIP_Cpd.

In that case both myMRFExtCp (SOL006 VNFD) and vduCp_B2 (SOL001 VNFD) would be renamed SIP_Cpd. The identifier VduCp_B would just be used in a SOL006 VNFD.

- **Use Case #2: external CP of a VNF connected to an internal VL**

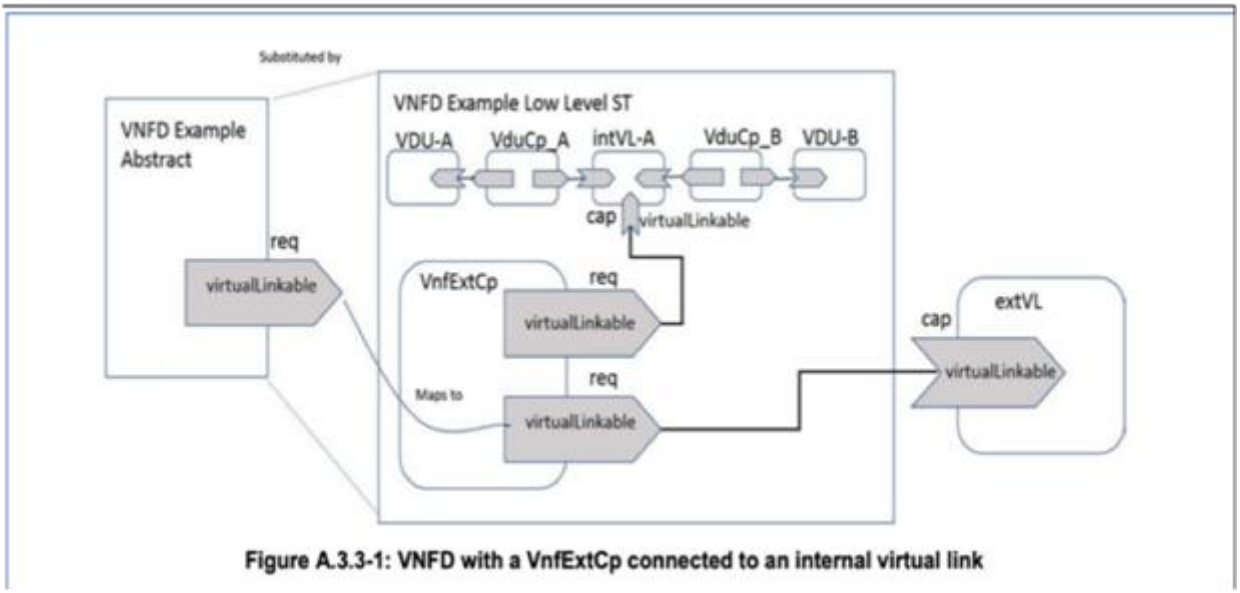

Figure 53. NFV SOL003 Extract (2)

- Here SOL001 and SOL006 require the explicit definition of an external CP node template or an external CPD (resp) from which to instantiate CPs connected to the internal VL.
- The "cpdId" in the body of the InstantiateVnfRequest (defined in SOL003) will be the ID of the defined external CP in both descriptors data model.

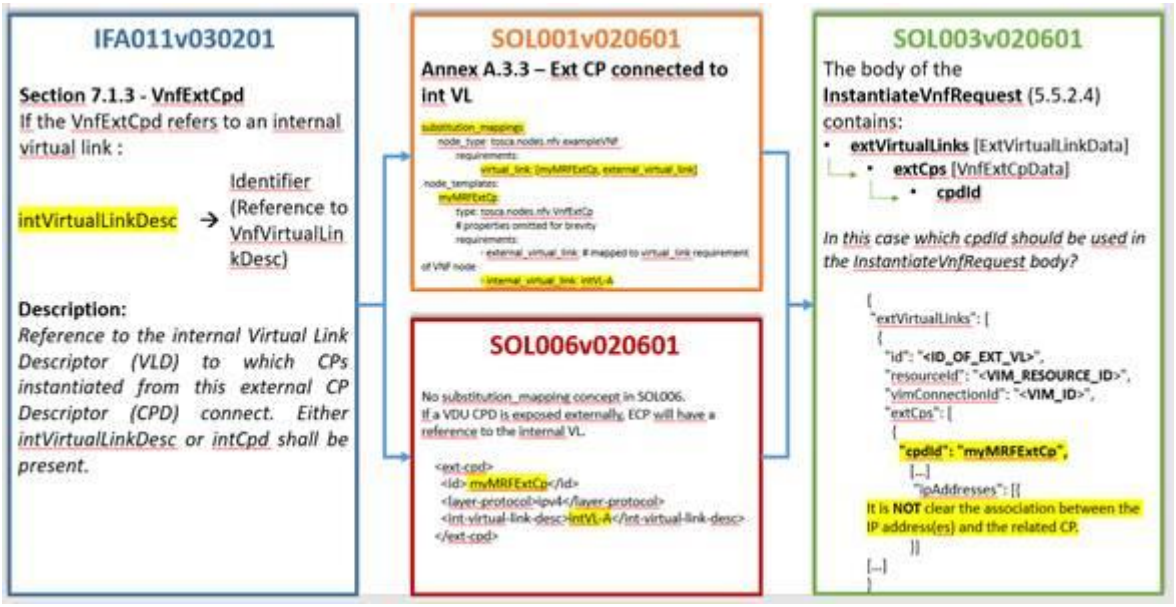

Figure 54. InstatiateRequest through ETSI NFV Specifications (2)

For this second use case, several questions were raised:

1. How should the internal VL (intVL-A) and the external network (extVL) connect? How can be mapped the eventual IP address(es) associated to the external CP to the internal ones?

   For this point, it was agreed that a router would be needed. There are ongoing discussions in NFV SOL WG on wich component of the NFV architecture would be expected to provide the router, and how it should be requested.

   Concerning the mapping, there is some related ongoing work in NFV SOL WG:

   - For the Virtual IP addresses, NFV SOL WG is introducing a specific Cp and a mechanism for the mapping, by linking this Cp with the cps that share these addresses.
   - For floating addresses what is being proposed is that if the VduCps connected to the VnfVirtualLink have requested floating_ip and there is a VnfExtCp connected to this VnfVirtualLink, the floating IP addresses will be allocated to the VnfExtCp. Thus, the information about them will come in SOL003 as part of the VnfExtCpData of the VnfExtCp. Then it would be up to the VNFM to map each floating address to each VduCp instance. In principle it is not foreseen a need for further mapping, i.e. if the VduCps are neither using floating nor VIP addresses, no mapping should be needed.

   A concrete example with Openstack was discussed, where the internal network (int VL deployed by the VNFM) and the external one can be connected through an Openstack router (unless they are the same network). In this case, it may be assumed that the floating IP behaviour is activated to enable connectivity among internal and external networks. That being said, it might not be required that floating IP is activated for all internal CPs in the internal VL, as some of these intCps may just be there to allow the internal communication inside the VNF, so the default private address assigned by OpenStack would be enough for them.

2. Floating IP activated or VIP use-case:

   a. As we do not expose internal connection points directly (VduCp_A and VduCp_B) as external connections, we cannot pass a fixed floating IP. How will this NICs get a fixed IP using SOL003?

   b. Can we assume in this case, in SOL001, along with marking floating_ip_activated to true in a VduCp, internal connection points are exposed externally in substitution mapping & extVirtualLinks structure of SOL003 is used to pass fixed IPs?

It was acknowledged that this was a gap in SOL003 Release 2. For Release 3, there has been some changes in IFA007 in order to add some additional functionality to indirectly create the vnfLinkPort for an internal virtual link of the VNF, when such IntVL is provided as an "externally-managed virtual link". So, in Release 3 there will be:

- case 1: VNFM is responsible for creating its IntVL, in which case, the IP addressing is determined by the VNFM, so not provided via SOL003.

- case 2: NFVO is responsible for creating the IntVL and give it to the VNFM in the form of an "ExtManagedVirtualLink". In this case, a number of vnfLinkPort can be pre-configured and passed over the Or-Vnfm from the NFVO to the VNFM. If such ports have been created, then they have a pre-configured IP addressing. The VNFM can learn such addressing by "querying information of the virtualised resources corresponding to that vnfLinkPort".

Another proposal suggests that floating addresses could be provided in SOL003 on the VnfExtCp that is connected to the internal VL to which the VduCps also connect. In addition, the VduCps receive their default IP addresses, but these are not provided via SOL003, as these VduCps are not VnfExtCps.

3. Furthermore, SOL003 doesn't have a way to pass fixed private IPs for externally managed virtual link, in above case for intVL-A which is externally exposed as extVL.

   On this point it was outlined that the referring to intVl-A being externally exposed as extVL was not fully correct. There is no exposure of internal VLs as external VLs. Only the intCp (VduCp) can be exposed as vnfExtCp, so the question should refer to, "intVL-A which is handled as an externally-managed VL"?.

   This being clarified, the point was acknowledged as not addressed in Release 2 documentation. A possible shortcut is being studied for Release 3.

Finally, it was mentioned that there is ongoing work in NFV IFA and SOL WGs aiming to provide more precise details on the behaviour and expectation of the VNFM, NFVO and VIM for these and other Use Cases

10.1.1.2 SOL003 - macAddress in IpOverEthernetAddressInfodata

According to SOL 003, Table 5.5.3.10-1, the attribute macAddress shall be always present. In the following versions (GS SOL 003 v2.5.1 and v2.6.1) the attribute has cardinality 0 .. 1, since in some circumstances it is not possible to get the macAddress. On the newer version it requires that at least one between macAddress and ipAddresses shall be present.

10.1.1.3 SOL005 - Inconsistency on userDefinedData attribute

The attribute userDefinedData presents some inconsistency on the GS SOL 005 v2.4.1. In some cases, its data type is a KeyValuePair (eg. Table 5.5.2.3-1) others the data type is object (eg. Table 5.5.2.1-1). This is being address and fixed with contribution [NFVSOL(18)000233]

10.1.2 MEC Specifications

10.1.2.1 MEC Reference Architecture and Components

A major source of questions for VNF providers was the differences in scope of MEC Platforms vs MEC Apps. Indeed, some VNFs willing to participate in the MEC Track were network based VNFs, providing services that are transparent at an applicative level. Indeed, they had capabilities which were initially in the scope of a MEC Platform but somehow they couldn't be considered either MEC platforms, nor simple MEC Apps, but shared capabilities from both. While those VNFs may not match completely with the description of a MEC App, it might make a lot of sense to run them as network applications at the edge. A point to be clarified is whether the end goal is only to be closer to the end user or to also to benefit from the MEC infrastructure.

Some additional questions were raised about instantiation and provisioning; and whether the instantiation of MEC Platform should be different from a MEC App.

10.1.2.2 Mapping between MEC and NFV Information Models

A clear mapping between the MEC App Descriptor and the VNF Descriptors specified in ETSI NFV would be needed to facilitate MEC capabilities implementation in NFV environments and MEC-in-NFV deployments

10.1.2.3 MEC API and Specification Versioning

Some issues were raised concerning the API versions available on ETSI Forge with regards to the related MEC Specifications. For example, the Forge repository for [MEC011] has been recently updated to use OpenAPI (v3) instead of Swagger (v2) without a clear link to the appropriate version of the base specification, and this led to some confusion among those who were looking at integrating and testing using MP1 API.

10.1.2.4 MEC API Conformance Test Specifications

Availability of MEC API Conformance Test Specifications and Test Suites (especially MP1) will be key for future MEC testing events, as they are being really helpful in the NFV track to validate NFV API servers in parallel with the Interoperability testing.

10.2 Feedback on OpenAPIs

10.2.1 SOL002 - “extCpConfig” should be an array

As per GS SOL 002, Table 9.5.3.3-1, the attribute extCpConfig should be an array of CpConfiguration not an object. Bug #245 has been opened on the ETSI Bugzilla platform [BUGZILLA]

10.3 Feedback on the Test Plans

10.3.1 NFV Interoperability Test Plan – TST007

10.3.1.1 New and updated Test Descriptions

The Interoperability Test Plan for the 4th NFV Plugtests was developed starting from the test plans of previous NFV Plugtests [1NFVPLU-TP], [2NFVPLU-TP] and [3NFVPLU-TP] with the aim of improving them in terms of alignment to latest versions of NFV specifications (e.g. new SOL APIs) and maximization of execution rates for the different test cases and groups.

The 4th NFV Plugtests Test Plan has been developed in the Plugtests preparation phase, from February to May 2019, trying to involve the Plugtests participants as much as possible mostly to collect feedbacks and ideas for potential improvements with respect to existing test cases and test descriptions. The process of Test Plan improvement has been implemented following two main directions:

1. Alignment of the test cases with NFV SOL specifications
2. Explicit introduction of granting test steps for lifecycle management operations

Concerning the first point, the Interoperability Test Plan has been restructured and updated trying to align the test groups and test cases to the NFV SOL APIs, operations and workflows as specified in latest releases of ETSI NFV GS SOL002, SOL003 and SOL005. This process had the objective to move a first step towards the automation of the Interoperability Test Plan, being the NFV SOL APIs themselves well-defined RESTful interfaces with available OpenAPI representations. In addition, the improvement of the Interoperability Test Plan targeted a specific alignment of test triggers with NFV SOL005 API calls and operations, thus mapping the interoperability external actions to existing operations in the NFV specifications. As a consequence, the Interoperability Test Plan for this 4th NFV Plugtests includes some updates to existing test cases as well as some new ones to cover specific features (as shown in the table below). This mostly affected both Network Service and VNF Performance Management and Fault Management related test cases, where subscription and notification operations were included as a way to align the interoperability checks

with NFV SOL specifications. Similarly, new test descriptions were included to cover the VNF Indicator related operations in NFV SOL002 and SOL003.

For what concerns the second point, the introduction of granting test steps for lifecycle management operations had the objective to group under the same set of test cases multiple SUT configurations, thus having the same reference test cases for Multi-Vendor NS, Single-Vendor NS and S-VNFM (in both direct and indirect case). This allows to have a more compact Interoperability Test Plan, closer to the NFV SOL specifications. In addition, with the idea of being agnostic of the VNFM mode (i.e. direct or indirect), the lifecycle management test cases were updated to have support of both options in the granting interoperability check test step, as shown in the reference test case snapshot below (for NS instantiation case).

Step	Type	Description	Result
1	Stimulus	Trigger NS instantiation in NFVO (i.e. create new NS instance resource and instantiate it)	
2	IOP Check	Verify that the VNFM receives instantation requests for the VNFs composing the given NS	
3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for the VNFs composing the given NS in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the instantiation of the VNFs composing the given NS	

Figure 55. New granting test step in NS instantiation test case

In summary, the following table summarizes those 4th NFV Plugtests new or updated test descriptions with respect to the 3rd NFV Plugtests Test Plan [3NFVPLU-TP].

Test Description ID	Feature	Modification	4th NFV Plugtests Test Plan Clause
TD_NFV_NS_LCM_INSTANTIATE_001	LCM	Updated	6.1.2
TD_NFV_NS_LCM_SCALE_OUT_001 TD_NFV_NS_LCM_SCALE_IN_001 TD_NFV_NS_LCM_SCALE_OUT_002a TD_NFV_NS_LCM_SCALE_IN_002a TD_NFV_NS_LCM_SCALE_OUT_002b TD_NFV_NS_LCM_SCALE_IN_002b TD_NFV_NS_LCM_SCALE_OUT_003 TD_NFV_NS_LCM_SCALE_IN_003	LCM	Update	6.1.3
TD_NFV_NS_LCM_SCALE_OUT_VNF_001 TD_NFV_NS_LCM_SCALE_IN_VNF_001 TD_NFV_NS_LCM_SCALE_OUT_VNF_002a TD_NFV_NS_LCM_SCALE_IN_VNF_002a TD_NFV_NS_LCM_SCALE_OUT_VNF_002b TD_NFV_NS_LCM_SCALE_IN_VNF_002b TD_NFV_NS_LCM_SCALE_OUT_VNF_003 TD_NFV_NS_LCM_SCALE_IN_VNF_003 TD_NFV_NS_LCM_SCALE_OUT_VNF_004 TD_NFV_NS_LCM_SCALE_IN_VNF_004	LCM	Update	6.1.4
TD_NFV_FM_NS_ALARM_SUBSCRIPTION_CREATE_001 TD_NFV_FM_NS_ALARM_SUBSCRIPTION_DELETE_001	FM	New	6.1.8.1

TD_NFV_FM_NS_ALARM_QUERY_001	FM	New	6.1.8.2
TD_NFV_FM_NS_ALARM_NOTIFICATION_001 TD_NFV_FM_NS_ALARM_CLEAR_NOTIFICATION_001	FM	Updated	6.1.8.2
TD_NFV_PM_NS_CREATE_SUBSCRIPTION_001 TD_NFV_PM_NS_CREATE_SUBSCRIPTION_001 TD_NFV_PM_NS_DELETE_SUBSCRIPTION_001 TD_NFV_PM_NS_THRESHOLD_CROSSED_NOTIFICATION_001 TD_NFV_PM_NS_MONITORING_INFORMATION_NOTIFICATION_001	PM	New	6.1.9
TD_NFV_VNF_INDICATOR_QUERY_001 TD_NFV_VNF_INDICATOR_SUBSCRIPTION_001 TD_NFV_VNF_INDICATOR_NOTIFICATION_001	VNF_IND	New	6.1.10

Table 43. Input to NFV TST007

As for previous NFV Plugtests, the test plan will be contributed to ETSI NFV TST WG as input for the revision of TST007 Guidelines for Interoperability Testing.

10.3.1.2 Missing TDs for Updating VNF Configuration

Besides the improvements on top of the test plan coming from the 3rd NFV Plugtests, some useful and constructive feedbacks have been collected from the 4th NFV Plugtests participants that helped both before and during the Plugtests event to consolidate the test plan. In particular it is worth mentioning one specific feedback that has a direct impact on the next versions of Interoperability Test Plan.

It was noted that in the context of the update Network Service and update VNF operations, the Interoperability Test Plan was not covering the runtime VNF configuration update, sometimes identified as “Day2” configuration. This is a critical operation that allows an operator to update VNF configurations at runtime.

As a consequence, the following new test case is suggested to be added (under the Update VNF test group) in the next versions of the Interoperability Test Plan to cover the runtime configuration update for the instantiated VNFs.

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_UPDATE_VNF_CONFIGURATION_001			
Test Purpose	To verify that the configuration of a VNF running in a NS can be successfully updated			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_AUTO-LCM-VALIDATION			
References	[IFA013] Clause 7.3 [IFA007] Clause 7.2 [SOL005] Clause 6.3 [SOL003] Clause 5.4.10 [SOL002] Clause 9.4.2			
Applicability	• NFVO supports NS update operations for VNF configuration modification • VNFM supports VNF modification operations for configuration modification • VNF supports VNF configuration modification			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger the NS update in NFVO with an operator request, specifying the new modified configuration to be applied to the given VNF(s)	

	2	IOP Check	Verify that the VNFM receives a VNF modification request for each new configuration to be applied to the given VNF(s)	
	3	IOP Check	Verify that the VNFM issue the modified configuration request to the relevant VNF(s) instance(s)	
	4	IOP Check	Verify that the modified configuration has been successfully applied in the given VNF(s)	
IOP Verdict				

Table 44. New Interoperability TD for VNF configuration update

10.3.1.3 Additional Steps for Checking VNFM-NFVO IOP

During the Plugtests, some suggestions were made to include additional steps / IOP checks to make a more precise validation of the FNVM-NFVO Interoperability. These suggestions summarized below are for further study in the NFV TST WG and/or future Plugtests.

- Create VNF (was the NFVO able to create a single VNF [ID])

  - External catalog (Is the catalog provided by the NFVO, defined by Sol003, used by the VNFM to fetch the VNF package during VNF lifecycle operations.)

  - Correct lifecycle notifications (LCNs) are sent by VNFM

- Instantiate VNF

  - Correct operationOccurence is created (phase changes (starting->processing->completed, etc.) are in line with sol003)

  - Correct LCNs are sent

  - Grant request was sent in line with VNFD

    - Synchronous grant response was created by NFVO in line with VNFD and grant request

    - Asynchronous grant response was created by NFVO in line with VNFD and grant request, VNFM was able to poll for the grant response

  - InstantiatedVnfInfo is matching the instantiation request, the grant response, the VNFD and the actual resources on VIM

- Scale VNF

  - Operation supported

  - Correct operationOccurence is created (phase changes (starting->processing->completed, etc.) are in line with sol003)

  - Correct LCNs are sent

  - Grant request was sent in line with VNFD

    - Synchronous grant response was created by NFVO in line with VNFD and grant request

    - Asynchronous grant response was created by NFVO in line with VNFD and grant request, VNFM was able to poll for the grant response

  - InstantiatedVnfInfo was updated to match the grant response, the VNFD and the actual resources on VIM

- Heal VNF

  - Operation supported

  - Correct operationOccurence is created (phase changes (starting->processing->completed, etc.) are in line with sol003)

  - Correct LCNs are sent

  - (Optonal) In case virtual resource are modified/removed/added, grant request was sent in line with VNFD

    - Synchronous grant response was created by NFVO in line with VNFD and grant request

    - Asynchronous grant response was created by NFVO in line with VNFD and grant request, VNFM was able to poll for the grant response

  - InstantiatedVnfInfo was updated to match the grant response (if there was any), the VNFD and the actual resources on VIM

- Terminate VNF

  - Correct operationOccurence is created (phase changes (starting->processing->completed, etc.) are in line with sol003)

  - Correct LCNs are sent

  - Grant request was sent in line with VNFD

    - Synchronous grant response was created by NFVO in line with VNFD and grant request

    - Asynchronous grant response was created by NFVO in line with VNFD and grant request, VNFM was able to poll for the grant response

  - InstantiatedVnfInfo was updated to match the (now 0) resources on the VIM

  - Resources created during instantiation are actually removed from VIM.

- Modify VNF

  - Operation supported

  - Correct operationOccurence is created (phase changes (processing->completed, etc.) are in line with sol003)

  - Correct LCNs are sent

  - Operation supported after VNF has been created, but before VNF has been instantiated

  - Changes initiated from NFVO are actually reflected instantly or during the next resource changing LCM operation

- Delete VNF

  - Correct LCNs are sent

  - VNF record is removed from VNFM

- Subscribe to VNF LCNs (NFVO is able to subscribe to LCNs using sol003)

  - Subscription filters are supported and considered, when sending out the LCNs

  - Unsubscribing works as well (initiated by NFVO)

10.3.2 NFV API Conformance Test Plan – TST010

The Test Plan for the NFV API Conformance Test Sessions during the 4rd NFV Plugtests was based on the test plan developed for the Remote NFV API Plugtests.

The main additional features were:

- Extending the APIs (i.e. NSD Management) and operations (e.g. VNF instance instantiation and termination) in scope,
- Automated check of message payloads via schema validation

The API validation track already highlighted several parts of the specifications that are prone to misinterpretation and implementation errors. This indicates that – even rather simple - syntactic checks on the API operations are already very valuable to provide insights on some potentially critical issues and raise awareness within the organizations participating to the API testing.

The learnings of this track will be contributed to ETSI NFV as input to TST010 NFV Conformance Testing.

Overall, the 4th NFV Plugtests allowed to identify and file 25 issues on the NFV API Conformance Test Suites. The table below summarises all the issues and indicates the impacted SOL Specification and the number under which the issue was filed in the [ISSUE-TRACKER] set up for that purpose in the ETSI Forge.

Issue	Description	SOL002	SOL003	SOL005
#66	Script Error in VNFConfiguration-API/Configuration.robot/Set new VNF Configuration Test Case	X		
#67	Schema validation failure in VNFConfiguration-API/DELETE VNF Configuration - Method not implemented Test Case	X		
#68	SOL002/VNFIndicator-API/Subscriptions.robot and IndividualSubscription.robot failing due to Discrepency in the Specifications and the schema definition for VnfIndicatorSubscription object	X		
#69	Script error in VNFIndicator-API/IndividualSubscription.robot	X		
#70	Wrong SETUP on Instantiate a vnfInstance Conflict	X		
#71	Wrong variable on Scale a vnfInstance Conflict (Not-Instantiated)	X		
#72	Missing GET Not Found (Wrong URI) option on NSDescriptors.robot			X
#73	BuiltIn Catenate keyword need a separator in order to avoid space between words	X	X	X
#74	Change variable in DELETE Single Network Service Descriptor			X
#75	Synchronous vs Asynchronous	X	X	X
#76	Synchronous PUT NSD Content should return 204 instead of 200			X
#77	REST library not supporting multipart payload.	X	X	X
#78	Error loading json in PNFContent			X
#79	Typos on Subscriptions.robot on NSD-Management-API			X
#80	Typo in IndividualVNFPackages			X
#81	Keyword "Check Postcondition VNF Package Subscription Is Set" should			X

	be checked			
#82	ETag should not be checked on PATCH Operations		X	
#83	Missing schema in SOL003/VNFLifeCycleManagement-API		X	
#84	Keyword "Check resource instantiated" in SOL003/VNFLifeCycleManagement-API should be parametrized		X	
#85	Wrong URI in SOL002/VNFIndicator-API/VNFIndicators.robot	X		
#86	Unable to generate 404 in all operations in LCM		X	X
#87	Update on Subscriptions in SOL005/VNFPackageManagement-API when duplications are tested			X
#89	Header Location should not be checked in Individual VNF LCM operation occurrence		X	X
#90	Content-Type when errors occur should be application/problem+json	X	X	X
#91	Checking Location header on Operation Tasks should be removed		X	X

Table 45. Issues Found and Reported in NFV TST010 Robot Test Suites

10.3.3 MEC Interoperability Test Plan

The following feedback was captured on the MEC Interoperability Test Plan

- Additional details required for provisioning and routing (how to trigger, etc..)
- Test Automation via APIs would be helpful for stimuli and IOP checks
- Better distinction between MEC App and MEC Platform roles required in NFV Deployments
- A common Application Descriptor and details on how it is consumed by different components would be needed.
- MP1 was identified as a critical interoperability point: additional details and/or dedicated TDs would be welcome
- Additional TDs for multi-site testing would be welcome in a future event

While in general the available TDs were found quite simple by most participants, it was agreed that they were a good starting point for triggering MEC interoperability testing activities and gathering interest. It can be expected that the level of complexity will grow in future events.

Annex A – NFV Interoperability Results per Test Case

	Interoperability		Not Executed	Totals	
	OK	NO	NA	Run	Results
TD_NFV_MULTIVENDOR_ONBOARD_VNF_PKG_001	23	0	0	23	23
TD_NFV_MULTIVENDOR_ONBOARD_NSD_001	23	0	0	23	23
TD_NFV_MULTIVENDOR_NS_LCM_INSTANTIATE_001	22	0	0	22	22
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_OUT_001	7	1	4	8	12
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_IN_001	7	1	4	8	12
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_OUT_002a	0	1	7	1	8
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_IN_002a	0	1	7	1	8
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_OUT_002b	1	1	6	2	8
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_IN_002b	1	1	6	2	8
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_OUT_003	2	0	4	2	6
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_IN_003	2	0	4	2	6
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_OUT_VNF_001	3	1	4	4	8
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_IN_VNF_001	2	1	4	3	7
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_OUT_VNF_002a	0	0	3	0	3
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_IN_VNF_002a	0	0	3	0	3
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_OUT_VNF_002b	2	0	1	2	3
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_IN_VNF_002b	2	0	1	2	3
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_OUT_VNF_003	0	0	3	0	3
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_IN_VNF_003	0	0	3	0	3
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_OUT_VNF_004	0	0	3	0	3
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_IN_VNF_004	0	0	3	0	3
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_TO_LEVEL_001	1	0	0	1	1
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_TO_LEVEL_002	0	0	1	0	1
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_TO_LEVEL_003	0	0	1	0	1
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_TO_LEVEL_VNF_001	1	0	0	1	1
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_TO_LEVEL_VNF_002	1	0	1	1	2
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_TO_LEVEL_VNF_003	0	0	2	0	2
TD_NFV_MULTIVENDOR_NS_LCM_SCALE_TO_LEVEL_VNF_004	0	0	2	0	2
TD_NFV_MULTIVENDOR_NS_LCM_UPDATE_STOP_VNF_001	7	2	4	9	13
TD_NFV_MULTIVENDOR_NS_LCM_UPDATE_START_VNF_001	7	2	4	9	13
TD_NFV_MULTIVENDOR_NS_FM_ALARM_ SUBSCRIPTION_CREATE_001	2	2	9	4	13

TD_NFV_MULTIVENDOR_NS_FM_ALARM_SUBSCRIPTION_DELETE_002	2	2	9	4	13
TD_NFV_MULTIVENDOR_NS_FM_ALARM_NOTIFICATION_001	0	2	9	2	11
TD_NFV_MULTIVENDOR_NS_FM_ALARM_CLEAR_NOTIFICATION_001	0	2	9	2	11
TD_NFV_MULTIVENDOR_NS_FM_ALARM_QUERY_001	0	2	9	2	11
TD_NFV_MULTIVENDOR_NS_PM_CREATE_MONITORING_JOB_001	3	0	5	3	8
TD_NFV_MULTIVENDOR_NS_PM_PERFORMANCE_METRICS_QUERY_001	3	0	5	3	8
TD_NFV_MULTIVENDOR_NS_PM_CREATE_THRESHOLD_001	1	2	5	3	8
TD_NFV_MULTIVENDOR_NS_PM_CREATE_SUBSCRIPTION_001	1	0	7	1	8
TD_NFV_MULTIVENDOR_NS_PM_CREATE_SUBSCRIPTION_002	1	0	7	1	8
TD_NFV_MULTIVENDOR_NS_PM_THRESHOLD_CROSSED_NOTIFICATION_001	2	0	6	2	8
TD_NFV_MULTIVENDOR_NS_PM_MONITORING_INFORMATION_NOTIFICATION_001	1	0	7	1	8
TD_NFV_MULTIVENDOR_NS_PM_DELETE_SUBSCRIPTION_002	1	0	7	1	8
TD_NFV_MULTIVENDOR_NS_PM_DELETE_MONITORING_JOB_001	0	2	6	2	8
TD_NFV_MULTIVENDOR_NS_PM_DELETE_THRESHOLD_001	0	2	6	2	8
TD_NFV_MULTIVENDOR_NS_VNF_INDICATOR_QUERY_001	0	1	5	1	6
TD_NFV_MULTIVENDOR_NS_VNF_INDICATOR_SUBSCRIPTION_001	0	0	6	0	6
TD_NFV_MULTIVENDOR_NS_VNF_INDICATOR_NOTIFICATION_001	0	0	6	0	6
TD_NFV_MULTIVENDOR_NS_LCM_TERMINATE_001	22	0	0	22	22
TD_NFV_MULTIVENDOR_NS_TEARDOWN_DELETE_NSD_001	22	0	0	22	22
TD_NFV_MULTIVENDOR_NS_TEARDOWN_DELETE_VNF_PKG_001	22	0	0	22	22
TD_NFV_SINGLEVENDOR_ONBOARD_VNF_PKG_001	27	1	1	28	29
TD_NFV_SINGLEVENDOR_ONBOARD_NSD_001	27	0	1	27	28
TD_NFV_SINGLEVENDOR_NS_LCM_INSTANTIATE_001	25	2	1	27	28
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_OUT_001	12	5	8	17	25
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_IN_001	12	5	8	17	25
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_OUT_002a	0	4	14	4	18
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_IN_002a	0	4	14	4	18
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_OUT_002b	2	4	12	6	18
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_IN_002b	2	4	12	6	18
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_OUT_003	0	0	5	0	5
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_IN_003	0	0	5	0	5
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_OUT_VNF_001	3	1	1	4	5
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_IN_VNF_001	3	1	1	4	5
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_OUT_VNF_002a	0	0	3	0	3
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_IN_VNF_002a	0	0	3	0	3

TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_OUT_VNF_002b	1	0	2	1	3
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_IN_VNF_002b	1	0	2	1	3
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_OUT_VNF_003	0	0	3	0	3
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_IN_VNF_003	0	0	3	0	3
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_OUT_VNF_004	0	0	3	0	3
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_IN_VNF_004	0	0	3	0	3
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_TO_LEVEL_001	4	4	8	8	16
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_TO_LEVEL_002	0	0	7	0	7
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_TO_LEVEL_003	0	0	4	0	4
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_TO_LEVEL_VNF_001	1	0	1	1	2
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_TO_LEVEL_VNF_002	0	0	2	0	2
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_TO_LEVEL_VNF_003	0	0	3	0	3
TD_NFV_SINGLEVENDOR_NS_LCM_SCALE_TO_LEVEL_VNF_004	0	3	13	3	16
TD_NFV_SINGLEVENDOR_NS_LCM_UPDATE_STOP_VNF_001	9	5	8	14	22
TD_NFV_SINGLEVENDOR_NS_LCM_UPDATE_START_VNF_001	9	5	8	14	22
TD_NFV_SINGLEVENDOR_NS_FM_ALARM_SUBSCRIPTION_CREATE_001	2	0	7	2	9
TD_NFV_SINGLEVENDOR_NS_FM_ALARM_SUBSCRIPTION_DELETE_002	2	0	7	2	9
TD_NFV_SINGLEVENDOR_NS_FM_ALARM_NOTIFICATION_001	0	0	6	0	6
TD_NFV_SINGLEVENDOR_NS_FM_ALARM_CLEAR_NOTIFICATION_001	0	0	6	0	6
TD_NFV_SINGLEVENDOR_NS_FM_ALARM_QUERY_001	0	0	6	0	6
TD_NFV_SINGLEVENDOR_NS_PM_CREATE_MONITORING_JOB_001	0	0	3	0	3
TD_NFV_SINGLEVENDOR_NS_PM_PERFORMANCE_METRICS_QUERY_001	0	0	3	0	3
TD_NFV_SINGLEVENDOR_NS_PM_CREATE_THRESHOLD_001	0	0	3	0	3
TD_NFV_SINGLEVENDOR_NS_PM_CREATE_SUBSCRIPTION_001	1	0	6	1	7
TD_NFV_SINGLEVENDOR_NS_PM_CREATE_SUBSCRIPTION_002	1	0	6	1	7
TD_NFV_SINGLEVENDOR_NS_PM_THRESHOLD_CROSSED_NOTIFICATION_001	1	0	6	1	7
TD_NFV_SINGLEVENDOR_NS_PM_MONITORING_INFORMATION_NOTIFICATION_001	1	0	6	1	7
TD_NFV_SINGLEVENDOR_NS_PM_DELETE_SUBSCRIPTION_002	1	0	6	1	7
TD_NFV_SINGLEVENDOR_NS_PM_DELETE_MONITORING_JOB_001	0	0	3	0	3
TD_NFV_SINGLEVENDOR_NS_PM_DELETE_THRESHOLD_001	0	0	3	0	3
TD_NFV_SINGLEVENDOR_NS_VNF_INDICATOR_QUERY_001	0	2	3	2	5
TD_NFV_SINGLEVENDOR_NS_VNF_INDICATOR_SUBSCRIPTION_001	0	2	3	2	5
TD_NFV_SINGLEVENDOR_NS_VNF_INDICATOR_NOTIFICATION_001	0	2	3	2	5
TD_NFV_SINGLEVENDOR_NS_LCM_TERMINATE_001	27	1	1	28	29
TD_NFV_SINGLEVENDOR_NS_TEARDOWN_DELETE_NSD_001	25	1	1	26	27

TD_NFV_SINGLEVENDOR_NS_TEARDOWN_DELETE_VNF_PKG_001	26	1	1	27	28
TD_NFV_S-VNFM_ONBOARD_VNF_PKG_001	7	0	0	7	7
TD_NFV_S-VNFM_ONBOARD_NSD_001	4	0	1	4	5
TD_NFV_S-VNFM_NS_LCM_INSTANTIATE_001	6	1	0	7	7
TD_NFV_S-VNFM_NS_LCM_SCALE_OUT_001	0	0	6	0	6
TD_NFV_S-VNFM_NS_LCM_SCALE_IN_001	0	0	6	0	6
TD_NFV_S-VNFM_NS_LCM_SCALE_OUT_002a	0	0	5	0	5
TD_NFV_S-VNFM_NS_LCM_SCALE_IN_002a	0	0	5	0	5
TD_NFV_S-VNFM_NS_LCM_SCALE_OUT_002b	0	0	5	0	5
TD_NFV_S-VNFM_NS_LCM_SCALE_IN_002b	0	0	5	0	5
TD_NFV_S-VNFM_NS_LCM_SCALE_OUT_003	0	0	2	0	2
TD_NFV_S-VNFM_NS_LCM_SCALE_IN_003	0	0	2	0	2
TD_NFV_S-VNFM_NS_LCM_SCALE_OUT_VNF_001	3	0	0	3	3
TD_NFV_S-VNFM_NS_LCM_SCALE_IN_VNF_001	3	0	0	3	3
TD_NFV_S-VNFM_NS_LCM_SCALE_OUT_VNF_002a	0	0	1	0	1
TD_NFV_S-VNFM_NS_LCM_SCALE_IN_VNF_002a	0	0	1	0	1
TD_NFV_S-VNFM_NS_LCM_SCALE_OUT_VNF_002b	0	0	1	0	1
TD_NFV_S-VNFM_NS_LCM_SCALE_IN_VNF_002b	0	0	1	0	1
TD_NFV_S-VNFM_NS_LCM_SCALE_OUT_VNF_003	0	0	1	0	1
TD_NFV_S-VNFM_NS_LCM_SCALE_IN_VNF_003	0	0	1	0	1
TD_NFV_S-VNFM_NS_LCM_SCALE_OUT_VNF_004	0	0	1	0	1
TD_NFV_S-VNFM_NS_LCM_SCALE_IN_VNF_004	0	0	1	0	1
TD_NFV_S-VNFM_NS_LCM_SCALE_TO_LEVEL_002	0	0	1	0	1
TD_NFV_S-VNFM_NS_LCM_SCALE_TO_LEVEL_003	0	0	2	0	2
TD_NFV_S-VNFM_NS_LCM_SCALE_TO_LEVEL_VNF_002	0	0	1	0	1
TD_NFV_S-VNFM_NS_LCM_SCALE_TO_LEVEL_VNF_003	0	0	1	0	1
TD_NFV_S-VNFM_NS_LCM_UPDATE_STOP_VNF_001	0	0	6	0	6
TD_NFV_S-VNFM_NS_LCM_UPDATE_START_VNF_001	0	0	6	0	6
TD_NFV_S-VNFM_NS_FM_ALARM_SUBSCRIPTION_CREATE_001	0	1	1	1	2
TD_NFV_S-VNFM_NS_FM_ALARM_SUBSCRIPTION_DELETE_002	0	0	2	0	2
TD_NFV_S-VNFM_NS_LCM_TERMINATE_001	5	0	0	5	5
TD_NFV_S-VNFM_NS_TEARDOWN_DELETE_NSD_001	4	0	1	4	5
TD_NFV_S-VNFM_NS_TEARDOWN_DELETE_VNF_PKG_001	6	0	0	6	6
TD_NFV_AUTOLCM_ONBOARD_VNF_PKG_001	3	0	2	3	5
TD_NFV_AUTOLCM_ONBOARD_NSD_001	4	0	1	4	5
TD_NFV_AUTOLCM_NS_LCM_INSTANTIATE_001	5	0	0	5	5

TD_NFV_AUTOLCM_NS_LCM_SCALE_OUT_001	3	0	1	3	4
TD_NFV_AUTOLCM_NS_LCM_SCALE_IN_001	1	2	1	3	4
TD_NFV_AUTOLCM_NS_LCM_SCALE_OUT_002a	0	0	3	0	3
TD_NFV_AUTOLCM_NS_LCM_SCALE_IN_002a	0	0	3	0	3
TD_NFV_AUTOLCM_NS_LCM_SCALE_OUT_002b	0	0	3	0	3
TD_NFV_AUTOLCM_NS_LCM_SCALE_IN_002b	0	0	3	0	3
TD_NFV_AUTOLCM_NS_LCM_SCALE_OUT_003	1	0	1	1	2
TD_NFV_AUTOLCM_NS_LCM_SCALE_IN_003	1	0	1	1	2
TD_NFV_AUTOLCM_NS_LCM_SCALE_OUT_VNF_001	1	0	0	1	1
TD_NFV_AUTOLCM_NS_LCM_SCALE_IN_VNF_001	1	0	0	1	1
TD_NFV_AUTOLCM_NS_LCM_UPDATE_STOP_VNF_001	0	0	2	0	2
TD_NFV_AUTOLCM_NS_LCM_UPDATE_START_VNF_001	0	0	2	0	2
TD_NFV_AUTOLCM_NS_FM_ALARM_SUBSCRIPTION_CREATE_001	0	0	2	0	2
TD_NFV_AUTOLCM_NS_FM_ALARM_SUBSCRIPTION_DELETE_002	0	0	2	0	2
TD_NFV_AUTOLCM_NS_FM_ALARM_NOTIFICATION_001	0	0	2	0	2
TD_NFV_AUTOLCM_NS_FM_ALARM_CLEAR_NOTIFICATION_001	0	0	2	0	2
TD_NFV_AUTOLCM_NS_FM_ALARM_QUERY_001	0	0	2	0	2
TD_NFV_AUTOLCM_NS_LCM_TERMINATE_001	5	0	0	5	5
TD_NFV_AUTOLCM_TEARDOWN_DELETE_NSD_001	5	0	0	5	5
TD_NFV_AUTOLCM_TEARDOWN_DELETE_VNF_PKG_001	3	0	2	3	5

Annex B – NFV API Conformance Results per Test Case

	API Conformance		Not Executed	Totals	
	OK	NO	NA	Run	Results
TD_API_VNF_SOL002_VNF_CONF_1	2	0	0	2	2
TD_API_VNF_SOL002_VNF_CONF_2	2	0	0	2	2
TD_API_VNF_SOL002_VNF_CONF_3	2	0	0	2	2
TD_API_VNF_SOL002_VNF_CONF_4	2	0	0	2	2
TD_API_VNF_SOL002_VNF_CONF_5	2	0	0	2	2
TD_API_VNF_SOL002_VNF_CONF_6	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_1	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_2	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_3	1	1	0	2	2
TD_API_VNF_SOL002_VNF_IND_4	0	2	0	2	2
TD_API_VNF_SOL002_VNF_IND_5	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_6	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_7	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_8	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_10	1	0	0	1	1
TD_API_VNF_SOL002_VNF_IND_9	1	0	0	1	1
TD_API_VNF_SOL002_VNF_IND_11	0	1	0	1	1
TD_API_VNF_SOL002_VNF_IND_12	1	0	0	1	1
TD_API_VNF_SOL002_VNF_IND_13	1	0	0	1	1
TD_API_VNF_SOL002_VNF_IND_14	1	0	0	1	1
TD_API_VNF_SOL002_VNF_IND_15	1	0	0	1	1
TD_API_VNF_SOL002_VNF_IND_16	1	0	0	1	1
TD_API_VNF_SOL002_VNF_IND_17	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_18	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_19	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_20	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_21	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_22	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_23	1	1	0	2	2
TD_API_VNF_SOL002_VNF_IND_24	1	1	0	2	2
TD_API_VNF_SOL002_VNF_IND_25	1	1	0	2	2

TD_API_VNF_SOL002_VNF_IND_26	1	1	0	2	2
TD_API_VNF_SOL002_VNF_IND_27	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_28	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_29	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_30	1	1	0	2	2
TD_API_VNF_SOL002_VNF_IND_31	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_32	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_33	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_34	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_35	2	0	0	2	2
TD_API_VNF_SOL002_VNF_IND_36	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_FM_1	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_FM_2	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_FM_3	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_FM_4	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_FM_5	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_FM_6	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_FM_7	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_PM_20	0	1	0	1	1
TD_API_VNFM_SOL003_VNF_PM_21	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_PM_22	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_PM_23	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_PM_24	0	1	0	1	1
TD_API_VNFM_SOL003_VNF_PM_25	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_PM_26	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_PM_27	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_PM_41	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_PM_42	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_PM_43	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_PM_44	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_PM_45	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_PM_46	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_PM_47	0	0	1	0	1
TD_API_VNFM_SOL003_VNF_PM_48	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_PM_49	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_PM_50	1	0	0	1	1

TD_API_NFVO_SOL003_VNF_PCKGM_1	2	2	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_2	3	1	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_3	2	2	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_4	2	2	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_5	2	2	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_6	1	3	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_7	2	2	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_8	1	1	2	2	4
TD_API_NFVO_SOL003_VNF_PCKGM_9	2	0	2	2	4
TD_API_NFVO_SOL003_VNF_PCKGM_10	2	2	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_11	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_12	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_13	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_14	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_15	3	1	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_16	3	1	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_17	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_18	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_19	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_20	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_31	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_32	0	0	1	0	1
TD_API_NFVO_SOL003_VNF_PCKGM_33	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_34	0	0	1	0	1
TD_API_NFVO_SOL003_VNF_PCKGM_35	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_36	0	1	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_37	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_38	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_39	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_40	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_1	3	0	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_2	3	0	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_3	2	1	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_4	1	1	1	2	3
TD_API_VNFM_SOL003_VNF_LCM_5	3	0	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_6	3	0	0	3	3

TD_API_VNFM_SOL003_VNF_LCM_7	3	0	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_8	2	1	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_9	3	0	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_10	3	0	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_11	3	0	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_12	1	1	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_13	1	0	1	1	2
TD_API_VNFM_SOL003_VNF_LCM_14	2	1	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_15	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_16	3	0	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_17	3	0	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_18	3	0	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_19	3	0	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_20	3	0	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_21	3	0	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_22	3	0	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_23	1	1	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_24	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_26	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_27	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_28	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_29	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_30	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_31	1	1	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_32	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_33	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_34	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_35	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_36	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_37	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_38	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_39	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_40	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_41	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_42	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_43	2	0	0	2	2

TD_API_VNFM_SOL003_VNF_LCM_44	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_45	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_46	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_48	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_49	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_50	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_51	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_52	0	1	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_56	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_57	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_58	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_59	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_60	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_61	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_62	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_64	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_65	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_66	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_67	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_68	3	0	0	3	3
TD_API_VNFM_SOL003_VNF_LCM_69	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_70	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_71	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_72	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_73	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_74	2	0	0	2	2
TD_API_VNFM_SOL003_VNF_LCM_75	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_76	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_77	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_78	0	0	1	0	1
TD_API_VNFM_SOL003_VNF_LCM_79	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_80	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_81	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_82	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_83	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_84	1	0	0	1	1

TD_API_VNFM_SOL003_VNF_LCM_85	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_86	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_87	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_103	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_119	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_120	0	1	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_121	0	0	1	0	1
TD_API_VNFM_SOL003_VNF_LCM_122	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_123	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_124	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_125	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_126	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_127	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_128	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_129	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_130	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_131	1	0	0	1	1
TD_API_VNFM_SOL003_VNF_LCM_132	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_21	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_22	2	0	2	2	4
TD_API_NFVO_SOL003_VNF_PCKGM_23	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_24	2	0	2	2	4
TD_API_NFVO_SOL003_VNF_PCKGM_25	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_26	0	4	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_27	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_28	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_29	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_30	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_41	3	1	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_42	3	1	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_43	3	1	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_44	2	1	0	3	3
TD_API_NFVO_SOL003_VNF_PCKGM_45	3	1	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_46	0	4	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_47	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_48	4	0	0	4	4

TD_API_NFVO_SOL003_VNF_PCKGM_49	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_50	4	0	0	4	4
TD_API_NFVO_SOL003_VNF_PCKGM_51	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_52	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_53	0	1	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_54	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_55	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_56	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_57	0	0	1	0	1
TD_API_NFVO_SOL003_VNF_PCKGM_58	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_59	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_60	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_61	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_62	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_63	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_64	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_65	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_66	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_PCKGM_67	1	0	0	1	1
TD_API_NFVO_SOL003_VNF_LCM_GRANT_1	0	2	1	2	3
TD_API_NFVO_SOL003_VNF_LCM_GRANT_2	1	0	2	1	3
TD_API_NFVO_SOL003_VNF_LCM_GRANT_3	1	1	0	2	2
TD_API_NFVO_SOL003_VNF_LCM_GRANT_4	3	0	0	3	3
TD_API_NFVO_SOL003_VNF_LCM_GRANT_5	3	0	0	3	3
TD_API_NFVO_SOL003_VNF_LCM_GRANT_6	3	0	0	3	3
TD_API_NFVO_SOL003_VNF_LCM_GRANT_7	3	0	0	3	3
TD_API_NFVO_SOL003_VNF_LCM_GRANT_8	2	0	0	2	2
TD_API_NFVO_SOL003_VNF_LCM_GRANT_9	1	1	0	2	2
TD_API_NFVO_SOL003_VNF_LCM_GRANT_10	0	0	2	0	2
TD_API_NFVO_SOL003_VNF_LCM_GRANT_11	0	1	0	1	1
TD_API_NFVO_SOL003_VNF_LCM_GRANT_12	2	0	0	2	2
TD_API_NFVO_SOL003_VNF_LCM_GRANT_13	2	0	0	2	2
TD_API_NFVO_SOL003_VNF_LCM_GRANT_14	2	0	0	2	2
TD_API_NFVO_SOL005_NSD_MGMT_1	1	3	0	4	4
TD_API_NFVO_SOL005_NSD_MGMT_2	2	2	0	4	4
TD_API_NFVO_SOL005_NSD_MGMT_3	2	2	0	4	4

TD_API_NFVO_SOL005_NSD_MGMT_4	3	1	0	4	4
TD_API_NFVO_SOL005_NSD_MGMT_5	3	1	0	4	4
TD_API_NFVO_SOL005_NSD_MGMT_6	2	1	0	3	3
TD_API_NFVO_SOL005_NSD_MGMT_7	1	3	0	4	4
TD_API_NFVO_SOL005_NSD_MGMT_8	1	3	0	4	4
TD_API_NFVO_SOL005_NSD_MGMT_9	2	2	0	4	4
TD_API_NFVO_SOL005_NSD_MGMT_10	1	3	0	4	4
TD_API_NFVO_SOL005_NSD_MGMT_11	4	0	0	4	4
TD_API_NFVO_SOL005_NSD_MGMT_12	4	0	0	4	4
TD_API_NFVO_SOL005_NSD_MGMT_13	4	0	0	4	4
TD_API_NFVO_SOL005_NSD_MGMT_14	1	2	0	3	3
TD_API_NFVO_SOL005_NSD_MGMT_15	2	1	0	3	3
TD_API_NFVO_SOL005_NSD_MGMT_16	2	1	0	3	3
TD_API_NFVO_SOL005_NSD_MGMT_17	2	1	0	3	3
TD_API_NFVO_SOL005_NSD_MGMT_18	1	2	0	3	3
TD_API_NFVO_SOL005_NSD_MGMT_19	2	0	0	2	2
TD_API_NFVO_SOL005_NSD_MGMT_20	3	0	0	3	3
TD_API_NFVO_SOL005_NSD_MGMT_21	3	0	0	3	3
TD_API_NFVO_SOL005_NSD_MGMT_22	3	0	0	3	3
TD_API_NFVO_SOL005_NSD_MGMT_23	3	0	0	3	3
TD_API_NFVO_SOL005_NSD_MGMT_24	2	0	0	2	2
TD_API_NFVO_SOL005_NSD_MGMT_25	1	0	1	1	2
TD_API_NFVO_SOL005_NSD_MGMT_26	2	0	0	2	2
TD_API_NFVO_SOL005_NSD_MGMT_27	1	0	1	1	2
TD_API_NFVO_SOL005_NSD_MGMT_28	2	0	0	2	2
TD_API_NFVO_SOL005_NSD_MGMT_29	1	1	0	2	2
TD_API_NFVO_SOL005_NSD_MGMT_30	1	0	1	1	2
TD_API_NFVO_SOL005_NSD_MGMT_31	2	0	0	2	2
TD_API_NFVO_SOL005_NSD_MGMT_32	1	1	0	2	2
TD_API_NFVO_SOL005_NSD_MGMT_33	2	0	0	2	2
TD_API_NFVO_SOL005_NSD_MGMT_34	2	0	0	2	2
TD_API_NFVO_SOL005_NSD_MGMT_35	2	0	0	2	2
TD_API_NFVO_SOL005_NSD_MGMT_36	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_37	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_38	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_39	1	0	0	1	1

TD_API_NFVO_SOL005_NSD_MGMT_40	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_41	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_42	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_43	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_44	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_45	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_46	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_47	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_48	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_49	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_50	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_51	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_52	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_53	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_54	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_55	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_56	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_57	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_58	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_59	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_60	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_61	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_62	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_63	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_64	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_65	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_66	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_67	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_68	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_69	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_70	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_71	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_72	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_73	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_74	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_75	1	0	0	1	1

TD_API_NFVO_SOL005_NSD_MGMT_76	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_77	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_78	1	0	0	1	1
TD_API_NFVO_SOL005_NSD_MGMT_79	1	0	0	1	1
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_1	1	2	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_2	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_3	2	1	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_4	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_5	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_6	1	2	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_7	1	2	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_8	2	0	1	2	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_9	2	0	1	2	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_10	2	1	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_11	1	2	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_12	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_13	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_14	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_15	2	1	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_16	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_17	1	2	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_18	1	2	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_19	2	1	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_20	2	1	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_21	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_22	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_23	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_24	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_25	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_26	1	1	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_27	2	1	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_28	2	1	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_29	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_30	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_31	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_32	3	0	0	3	3

TD_API_NFVO_SOL005_VNF_PCKG_MGMT_33	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_34	2	0	1	2	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_35	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_36	2	0	1	2	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_37	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_38	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_39	1	0	0	1	1
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_40	1	0	0	1	1
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_41	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_42	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_43	3	0	0	3	3
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_44	1	0	0	1	1
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_45	1	0	0	1	1
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_46	1	0	0	1	1
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_47	1	0	0	1	1
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_48	1	0	0	1	1
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_49	1	0	0	1	1
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_50	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_51	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_52	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_53	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_54	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_55	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_56	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_57	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_58	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_59	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_60	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_61	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_62	1	1	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_63	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_64	1	0	0	1	1
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_65	1	0	1	1	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_66	1	0	0	1	1
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_67	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_68	2	0	0	2	2

TD_API_NFVO_SOL005_VNF_PCKG_MGMT_69	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_71	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_72	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_73	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_74	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_75	2	0	0	2	2
TD_API_NFVO_SOL005_VNF_PCKG_MGMT_76	2	0	0	2	2

History

Document history		
V0.1.0	11/07/2019	Shared for Plugtests Participants review