ETSI Plugtests Report V1.0.0 (2021-11)

ETSI NFV&MEC IOP Plugtests
Remote
1- 15 October 2021

Keywords

Testing, Interoperability, NFV, MANO, VNF, VIM, MEC

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

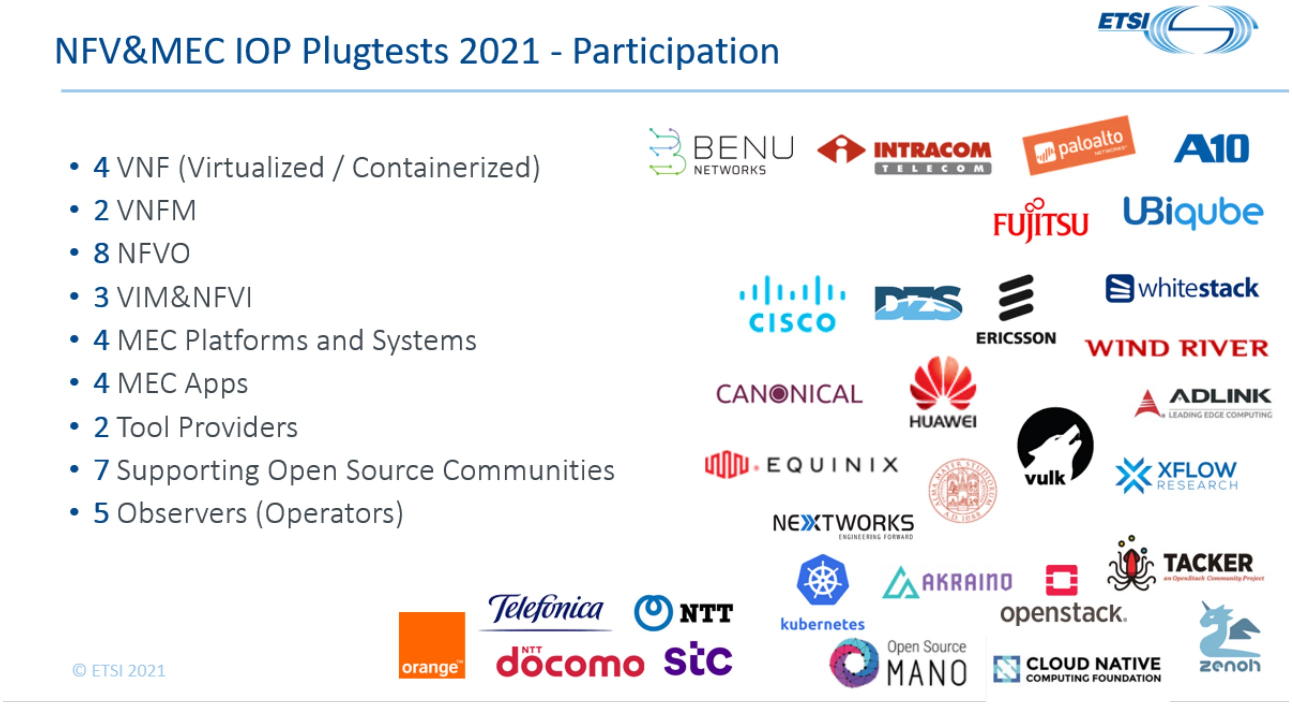
Executive summary

The NFV&MEC IOP Plugtests 2021 was organised by the ETSI Centre for Testing and Interoperability as part of the NFV Plugtests Programme. The event was held remotely due to the COVID-19 pandemics travel and hosting restrictions.

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

A total of 29 organisations and over 75 engineers were involved in the preparation of this 2 weeks event forming an engaged and diverse community of implementers testing together over 25 NFV and MEC solutions, such as:

- Virtual and Containerized Network Functions (VNFs)
- Management and Orchestration solutions: NFV Orchestrators and VNF Managers (NFVO, VNFM)
- NFV Platforms: NFV Infrastructure and Virtual and Infrastructure Managers (NFVI, VIM), in some cases, offering and managing Container Infrastructure Services (CIS, CISM)
- MEC Platforms, Systems and Applications
- Test and automation tools.



NFV&MEC IOP Plugtests 2021 - Participation

- 4 VNF (Virtualized / Containerized)
- 2 VNFM
- 8 NFVO
- 3 VIM&NFVI
- 4 MEC Platforms and Systems
- 4 MEC Apps
- 2 Tool Providers
- 7 Supporting Open Source Communities
- 5 Observers (Operators)

© ETSI 2021

Figure 1. NFV&MEC IOP Plugtests 2021 Participation

Different participating organizations and Functions Under Test (FUTs) 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 NFV&MEC Plugtests were:

- The remarkable progress in MEC IOP Testing, where for the first-time orchestration was included in the test configurations. The test plan for MEC interoperability has been significantly extended with regards to previous events and it is being contributed to ETSI MEC ISG as input for the MEC Interoperability Test Specification.

- The consolidation of the Test Automation Platform for MEC and NFV API Conformance Testing. The platform benefits now from a reliable CI/CD pipeline allowing to make the fixes in the Test Suites available to Plugtests participants in the matter of minutes.

The following sections describe in detail the preparation of the NFV&MEC 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.

This Plugtests Report is fed back to ETSI NFV and MEC Industry Specification Groups.

1 Introduction

The NFV&MEC IOP Plugtests 2021 was focused on verifying interoperability across different implementations of the main components of the NFV and MEC Architectural Frameworks, including:

- Virtual and Containerized Network Functions (VNF)
- Management and Orchestration (MANO) solutions, including integrated or standalone NFV Orchestrators (NFVO) and VNF Managers (VNFM)
- NFV infrastructure (NFVI) and Virtual Infrastructure Managers (VIM) in some cases providing Container Infrastructure Services and Services Management (CIS and CISM)
- MEC Applications, MEC Platforms and Systems

In addition, participants were offered the opportunity to test their NFV and MEC API implementations via a Test Automation Platform running standardized NFV and MEC API Conformance Test Suites.

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, MEC platforms and MANO solutions running remotely on participants' labs. VNF and MEC Apps Packages and images were made available in a local Repository and Docker Registry hosted at ETSI.These packages were uploaded to the different NFV / MEC platforms during the pre-testing phase. All the participants had access to the HIVE network, either over their company's site-to-site VPN or through a personal client-to-site VPN.

As in previous events, all the information required to organise, coordinate, and manage the NFV&MEC IOP Plugtests 2021 was compiled and shared with participants in the private NFV Plugtests Programme 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".

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

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

[NFV-TST010] ETSI GS NFV-TST 010: “Network Functions Virtualisation (NFV); API Conformance Testing Specification”

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

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

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

[NFV-SOL009] ETSI GS NFV-SOL 009: "Network Functions Virtualisation (NFV); Protocols and Data Models: RESTful protocols specification for the management of NFV-MANO"

[NFV-SOL011] ETSI GS NFV-SOL 011: "Network Functions Virtualisation (NFV); Protocols and Data Models; RESTful protocols specification for the Or-Or Reference Point"

[NFV-SOL012] ETSI GS NFV-SOL 012: "Network Functions Virtualisation (NFV); Protocols and Data Models; RESTful protocols specification for the Policy Management Interface".

[NFV-SOL013] ETSI GS NFV-SOL013: “Network Functions Virtualisation (NFV); Protocols and Data Models; Specification of common aspects for RESTful NFV MANO APIs”

[NFV-IF0005] ETSI GS NFV-IFA005: “Network Functions Virtualisation (NFV); Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification”

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

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

[NFVIFA008] ETSI GS NFV-IFA008: “Network Functions Virtualisation (NFV); Management and Orchestration; Ve-Vnfm reference point - Interface and Information Model Specification”

[NFV-IFA010] ETSI GS NFV-IFA010: “Network Functions Virtualisation (NFV); Management and Orchestration; Functional requirements specification”

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

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

[MEC009] ETSI GS MEC 009 V1.1.1: Mobile Edge Computing (MEC); General principles for Mobile Edge Service APIs

[MEC010-2] ETSI GS MEC 010-2 V2.1.1: “Multi-access Edge Computing (MEC); MEC Management; Part 2: Application lifecycle, rules and requirements management”

[MEC011] ETSI GS MEC 011 V2.1.1: “Multi-access Edge Computing (MEC); Mobile Edge Platform Application Enablement”

[MEC012] ETSI GS MEC 012 V2.1.1: “Multi-access Edge Computing (MEC); Radio Network Information API”

[MEC013] ETSI GS MEC 013 V2.1.1: “Multi-access Edge Computing (MEC); Location API”

[MEC014] ETSI GS MEC 014 V2.1.1: “Multi-access Edge Computing (MEC); UE Identity API”

[MEC015] ETSI GS MEC 015 V2.2.1: “Multi-Access Edge Computing (MEC); Traffic Management APIs”

[MEC016] ETSI GS MEC 016 V2.2.1: “Multi-access Edge Computing (MEC); Device application interface”

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

[MEC021] ETSI GS MEC 021 V2.1.1: “Multi-access Edge Computing (MEC); Application Mobility Service API”

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

[MEC028] ETSI GS MEC 028 V2.2.1: “Multi-access Edge Computing (MEC); WLAN Access Information API”

[MEC029] ETSI GS MEC 029 V2.1.1: “Multi-access Edge Computing (MEC); Fixed Access Information API”

[MEC030] ETSI GS MEC 030 V2.1.1: “Multi-access Edge Computing (MEC); V2X Information Service API”

[MEC-DEC032] ETSI GS MEC-DEC032: “Multi-access Edge Computing (MEC); API Conformance Test Specification”

[MEC-IOP-TP] NFV&MEC Plugtests 2021 - Test Plan for MEC Interoperability V1.1.0
https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/MEC_Plugtests_Test_Plan.pdf

[NFVMECPLU2020-R] ETSI NFV&MEC Plugtests Report 2020:
https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/ETSI_NFV&MEC_2020_Plugtests_Report_v1_0_0.pdf

[NFVMECAPI2021-R] ETSI NFV&MEC API Plugtests 2021 Report:
https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/ETSI_NFV&MEC_2021_API_Plugtests_Report_v1_0_0.pdf

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

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

[MEC-ROBOT-TS] Robot Test Suite for MEC API Conformance https://forge.etsi.org/rep/mec/gs032p3-robot-test-suite

[MEC-TTCN3-TS] TTCN-3 Test Suite for MEC API Conformance https://forge.etsi.org/rep/mec/gs032p3-ttcn-test-suite

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

[NFV-AUTO-IOP] Robot Test Suite for NFV Automated Interoperability https://forge.etsi.org/rep/plugtests/nfv/automated-interop

[MEC-ROBOT-ISSUE] Issue Tracker for the Robot Test Suite for MEC API Conformance https://forge.etsi.org/rep/mec/gs032p3-robot-test-suite/issues

[MEC-TTCN3-ISSUE] Issue Tracker for the TTCN-3 Test Suite for MEC API Conformance https://forge.etsi.org/rep/mec/gs032p3-ttcn-test-suite/issues

[CNFC-CNF-TS] CNCF CNF Test Suite v0.14.2 https://github.com/cncf/cnf-testsuite/releases/tag/v0.14.0

[CNCF-EVE011] CNCF CNF Testing comparison with NFV-EVE011 https://docbox.etsi.org/ISG/NFV/Open/Other/NFV(21)000177r1%20CNF%20Testsuite%20and%20EVE011%20Comparison.docx

[NFV-MEC-API-TR] NFV&MEC API Conformance Test Results per Test Case https://nfvwiki.etsi.org/images/NFV%26MEC_2021_API_Conformance_Results_per_Test_Case.pdf

3 Abbreviations

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

4 Technical and Project Management

4.1 Scope

The NFV&MEC Plugtests focused on testing several areas

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

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, and eventually Containerized Infrastructure Services
- One MANO solution, providing pre-integrated NFVO and VNFM or standalone NFVO functionality
- One VNFM solution, providing standalone VNFM functionality
- One Network Service, composed of VNFs/CNFs from one or several providers.

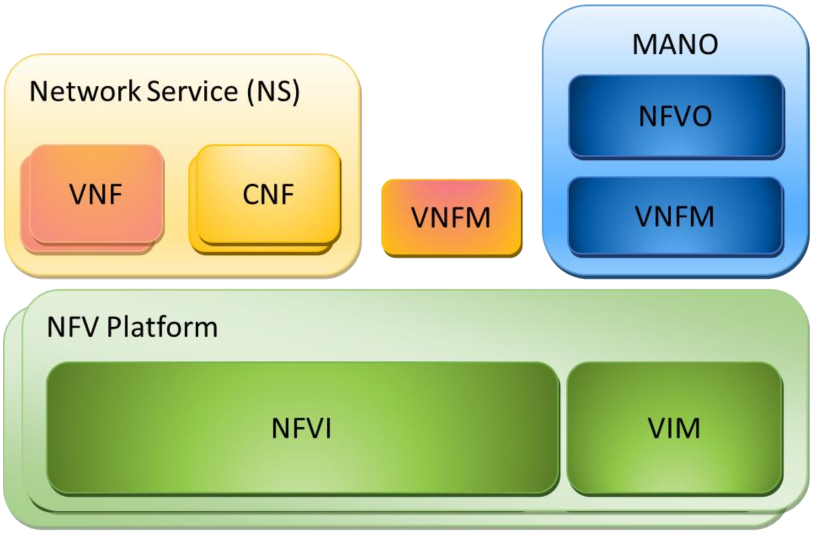


Figure 4.1.1-1. NFV Interoperability System Under Test

4.1.2 MEC Interoperability

The MEC Interoperability testing aimed at testing interoperability of MEC Applications execution on different MEC Platform and in different deployment types (i.e., MEC standalone and MEC in NFV). The MEC Interoperability Track proposed 4 groups of interoperability tests covering application lifecycle management, traffic and DNS management, MEC Service management, and MEC Location service. The main interoperable interfaces were between MEC Applications and MEC Platforms (the Mp1 reference point). The test configurations were derived from the generic Interoperability testing architecture reported in [MEC025].

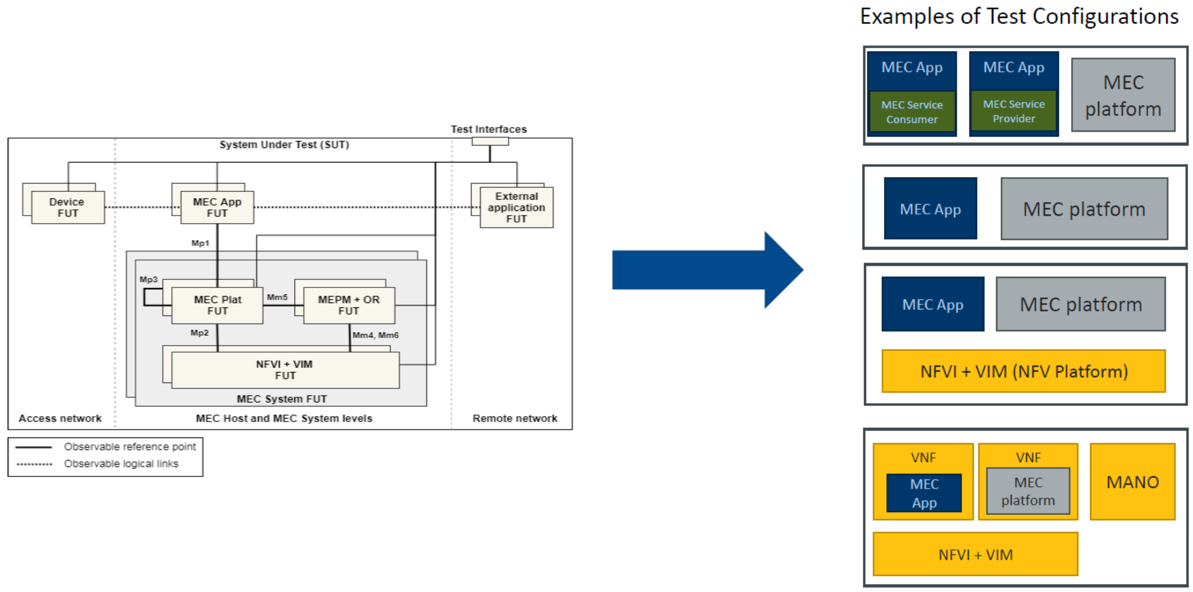


Figure 4.1.2-1. MEC Interoperability Testing

4.1.3 API Conformance

The main goal of the NFV and MEC API Conformance test sessions was to run individual Test Sessions between participating Functions Under Test and a Test Automation Platform (TAP) provided by the Plugtests Team. These sessions allowed to:

- validate the Robot Test Suites (run by the Test Automation Platform) for the NFV API Conformance Test Specification: [NFV-TST010] (v.2.4.1, v2.6.1, v2.7.1 and draft versions of v2.8.1, v3.3.1)
- validate the Robot Test Suites (run by the Test Automation Platform and manually by the Plugtests Team) from MEC API Conformance Test Specification: MEC DECODE
- validate the TTCN-3 Test Suites (run manually by the Plugtests Team or participants) from MEC API Conformance Test Specification: MEC DECODE
- assess the conformance level of participating Functions Under Test composed of VNFs, VNFMs, and NFVOs (operated by participants) with [NFV-SOL002], [NFV-SOL003], and [NFV-SOL005] APIs and OpenAPIs (v.2.4.1, v2.6.1, v2.7.1, v2.8.1 and v3.3.1)
- asses the conformance level of participating Function Under Test composed of MEC Platforms and Services (operated by participants) with [MEC010-2], [MEC011], [MEC012], [MEC013], [MEC014], [MEC015], [MEC016], [MEC021], [MEC028], [MEC029] and [MEC030] APIs and OpenAPI (v2.1.1 and v2.2.1).

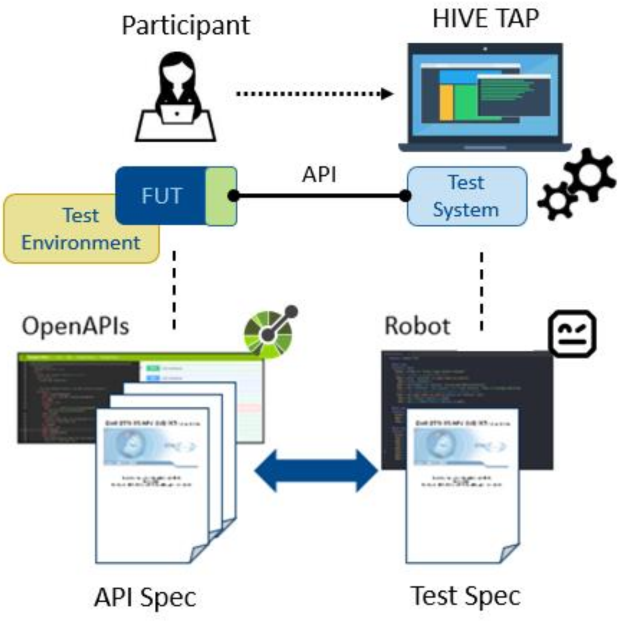


Figure 4.1.3-1. API Conformance Testing

4.2 Timeline

The NFV&MEC IOP Plugtests 2021 preparation started a few weeks after the NFV&MEC API Plugtests 2021, which was held in February. The preparation of the event ran through different phases as described in the figure below.

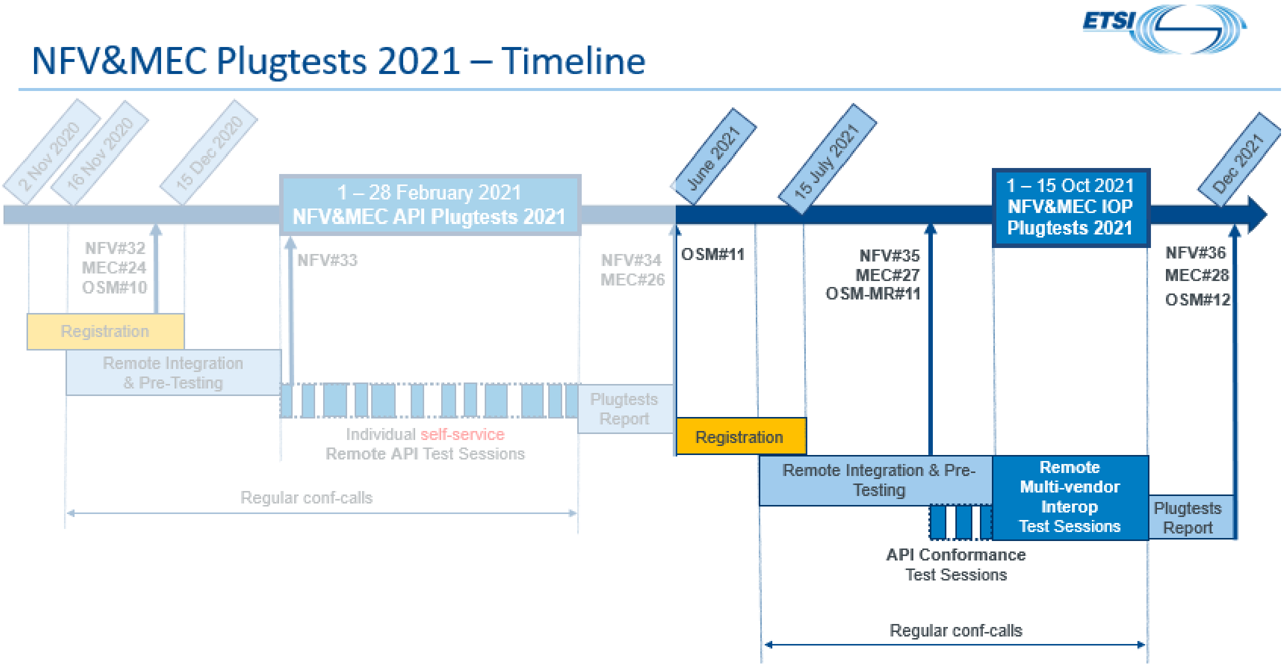


Figure 4.2-1. NFV&MEC IOP Plugtests 2021 timeline

Registration to Plugtests was open from June to mid-July 2021 to any organisation willing to participate with a Function Under Test, or a test tool. Registration was also open to observers (network operators, research, and academia) and supporting Open-Source communities willing to contribute to the event organization, discussions, and feedback gathering. Overall, over 75 engineers from 29 different organizations were involved in the Plugtests preparation and test sessions.

The following sections describe the different phases of the Plugtests preparation. During all these phases, weekly conf-calls were run among organisers and participants to discuss and track the remote-integration progress, anticipate and solve technical issues, review and discuss the test plans, prepare for the different types of test sessions, and discuss feedback and findings.

4.2.1 Remote Integration & Pre-Testing

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

1) FUT Documentation

Participants documented their Functions Under Test (FUTs), by filling in a form compiling the simplified Interoperability Features Statements (IFS) for each of their implementations. The simplified IFS Templates for each type of NFV FUT used for this Plugtests can be found in Annex A. The IFS Template for the MEC FUTs can be found in the Annex A of the MEC IOP Test Plan [MEC-IOP-TP]

Participants created a wiki page to compile information about their FUTs (description, diagrams, links to documentation.), as well as about the team(s) operating the FUT and representing the company during the preparation phase and the Test Sessions (email, Slack Id, time-zone etc.).

Participants providing VNFs complemented their documentation with diagrams and resource requirements.

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

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

2) Test Plan Discussion and Review

This Plugtests event relayed in several existing Test Specifications, guides and test suites published by ETSI NFV and MEC, such as [NFV-TST007] for NFV Interoperability, [NFV-TST010] for NFV API Conformance and, [MEC-DEC032] for MEC API Conformance. In addition, the Test Plan used in previous events to support MEC Interoperability Test Sessions, was re-discussed and improved. This resulting MEC IOP Test Plan [MEC-IOP-TP] has been contributed to ETSI MEC ISG for its adoption as GR MEC-DEC042 Guidelines for MEC Interoperability Testing.

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 connect their labs to HIVE when they register for an event, so that they can interact with other participants FUTs to configure complex multi-vendor Systems Under Test.

Participants are invited to maintain their connection to HIVE in between events, and use it to run additional testing, PoCs, demos and showcases. New participants that joined this Plugtests for the first time, were invited and guided by the Plugtests team to get their implementations available on HIVE.

At the end of this phase, over 22 remote sites were connected to HIVE, each of them with a dedicated subnet for the FUTs they host. 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 scarce time during the interop test sessions.

A site-to-site connection to HIVE was mandatory for participants providing NFV or MEC platforms and MANO solutions, and highly recommended for participants providing VNF, MEC Apps, VIM, and CISM software. The latter could also rely on client-to-site connection to HIVE, and the 3rd party lab running their software, as long as they had no additional 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.

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

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

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

4.2.2 Remote API Conformance Test Sessions

From 15 to 30 September, during the two weeks that preceded the Interop Test Sessions, participants were invited to run API Conformance Test sessions to assess the level of conformance of their implementations with NFV SOL and MEC APIs over the Test Automation Platform hosted in HIVE.

Upon request of the participants, API Conformance testing was also made possible during the 2 weeks of IOP Test Sessions.

4.2.3 Remote Interoperability Test Sessions

From 1 – 15 October, participants were invited to run collaboratively multi-vendor Interoperability test sessions, according to a test schedule provided by the Plugtests team. The IOP Test Sessions were scheduled taking into account participants time-zones and availability. A Sync call was run every other day to discuss progress, issues, findings and required arrangements.

4.3 Tools

4.3.1 Plugtests Wiki

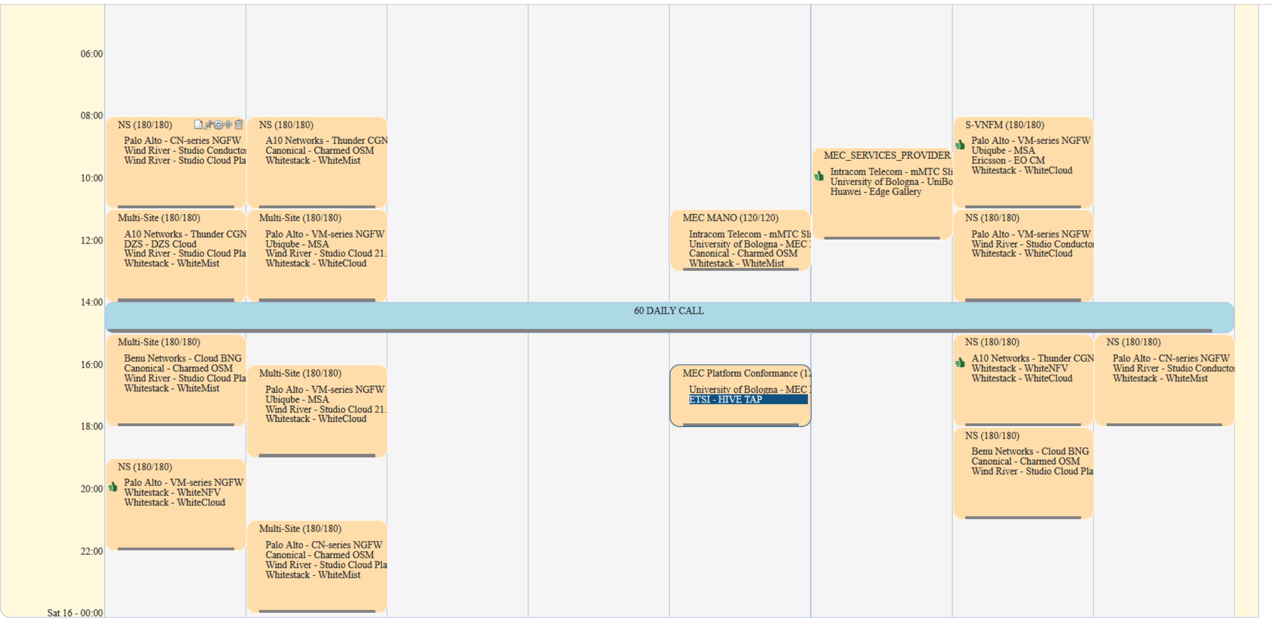
The NFV Plugtests Wiki is the main entry point for all the information concerning NFV&MEC Plugtests, from logistics aspects to testing procedures. Access to this Wiki is restricted to companies participating in the NFV Plugtests Programme.

4.3.2 Test Session Scheduler

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

- maximise the number of test sessions
- balance the amount of test sessions among participants
- consider supported features of the participating FUTs
- consider participants' time-zones and availability
- minimise the number of participants not involved in a test session at any given time

The picture below shows a partial view of a daily schedule. Each orange box corresponds to a specific Test Session, which depending on the targeted configuration included different components. In the NFV tracks, sessions included a Network Service with one or more VNFs from one or different providers, one MANO solution, one or more VIM&NFVI and one or more VNFMs. In the MEC track, each session included one MEC Platform and one or several MEC Apps. For each of these sessions a Test Session Report (TSR) was filed (see next clause). In addition to the pre-scheduled test sessions, participants were invited to request, run and report results for additional test sessions.



NS (180/180)
Palo Alto - CN-series NGFW
Wind River - Studio Conductor
Wind River - Studio Cloud Pla

Multi-Site (180/180)
A10 Networks - Thunder CGN
DZS - DZS Cloud
Wind River - Studio Cloud Pla
Whitestack - WhiteMist

Multi-Site (180/180)
Benu Networks - Cloud BNG
Canonical - Charmed OSM
Wind River - Studio Cloud Pla
Whitestack - WhiteMist

NS (180/180)
Palo Alto - VM-series NGFW
Whitestack - WhiteNFV
Whitestack - WhiteCloud

NS (180/180)
A10 Networks - Thunder CGN
Canonical - Charmed OSM
Whitestack - WhiteMist

Multi-Site (180/180)
Palo Alto - VM-series NGFW
Ubiqube - MSA
Wind River - Studio Cloud 21.
Whitestack - WhiteCloud

Multi-Site (180/180)
Palo Alto - VM-series NGFW
Ubiqube - MSA
Wind River - Studio Cloud 21.
Whitestack - WhiteCloud

Multi-Site (180/180)
Palo Alto - CN-series NGFW
Canonical - Charmed OSM
Wind River - Studio Cloud Pla
Whitestack - WhiteMist

60 DAILY CALL

MEC MANO (120/120)
Intracom Telecom - mMTC Sli
University of Bologna - MEC I
Canonical - Charmed OSM
Whitestack - WhiteMist

MEC Platform Conformance (12
University of Bologna - MEC I
ETSI - HIVE TAP

MEC_SERVICES_PROVIDER
Intracom Telecom - mMTC Sli
University of Bologna - UniBo
Huawei - Edge Gallery

S-VNFM (180/180)
Palo Alto - VM-series NGFW
Ubiqube - MSA
Ericsson - EO CM
Whitestack - WhiteCloud

NS (180/180)
Palo Alto - VM-series NGFW
Wind River - Studio Conductor
Whitestack - WhiteCloud

NS (180/180)
A10 Networks - Thunder CGN
Whitestack - WhiteNFV
Whitestack - WhiteCloud

NS (180/180)
Benu Networks - Cloud BNG
Canonical - Charmed OSM
Wind River - Studio Cloud Pla

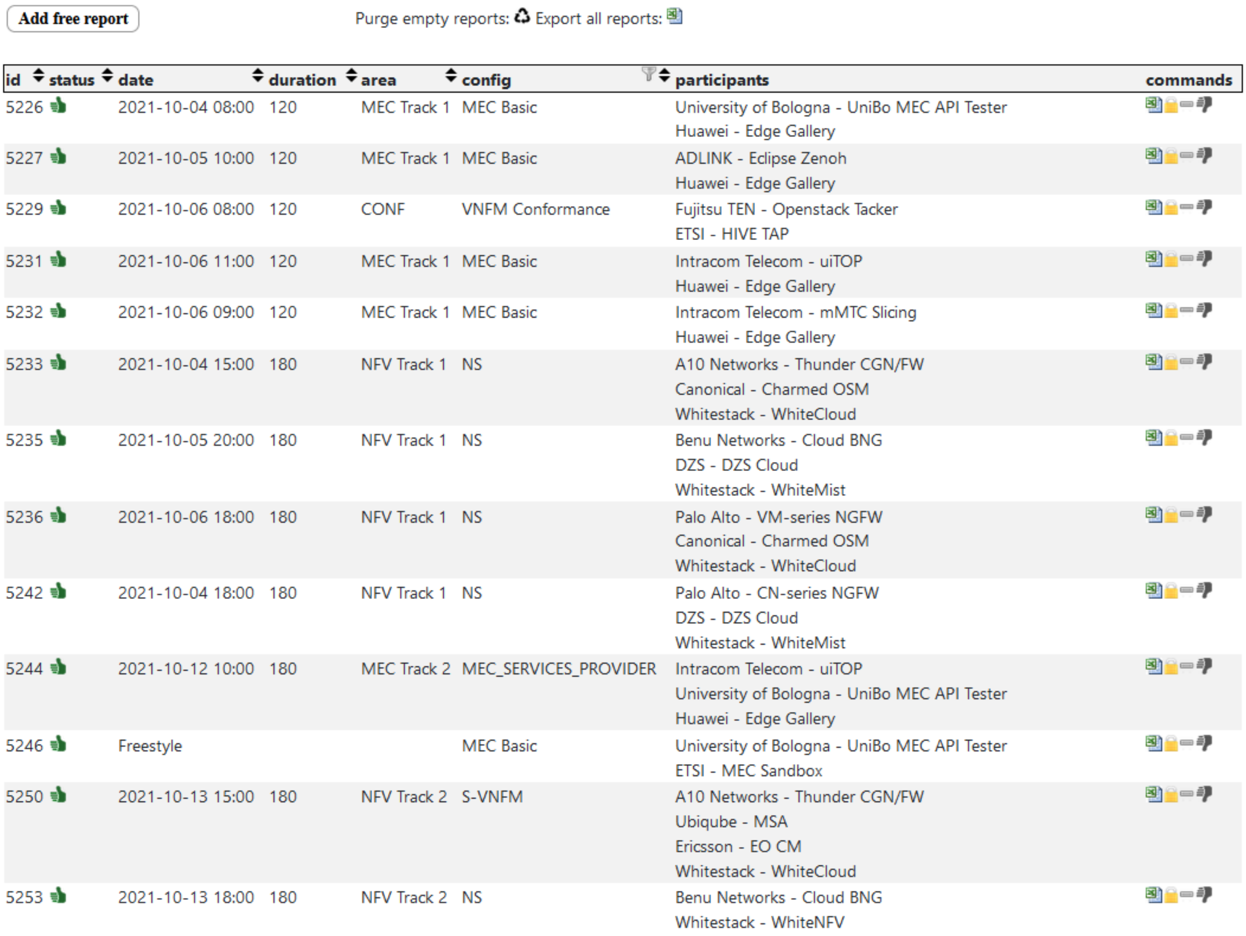
NS (180/180)
Palo Alto - CN-series NGFW
Wind River - Studio Conductor
Whitestack - WhiteMist

Figure 4.3.2-1. 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 or 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.



id	status	date	duration	area	config	participants	commands
5226	👍	2021-10-04 08:00	120	MEC Track 1	MEC Basic	University of Bologna - UniBo MEC API Tester Huawei - Edge Gallery	
5227	👍	2021-10-05 10:00	120	MEC Track 1	MEC Basic	ADLINK - Eclipse Zenoh Huawei - Edge Gallery	
5229	👍	2021-10-06 08:00	120	CONF	VNFM Conformance	Fujitsu TEN - Openstack Tacker ETSI - HIVE TAP	
5231	👍	2021-10-06 11:00	120	MEC Track 1	MEC Basic	Intracom Telecom - uiTOP Huawei - Edge Gallery	
5232	👍	2021-10-06 09:00	120	MEC Track 1	MEC Basic	Intracom Telecom - mMTC Slicing Huawei - Edge Gallery	
5233	👍	2021-10-04 15:00	180	NFV Track 1	NS	A10 Networks - Thunder CGN/FW Canonical - Charmed OSM Whitestack - WhiteCloud	
5235	👍	2021-10-05 20:00	180	NFV Track 1	NS	Benu Networks - Cloud BNG DZS - DZS Cloud Whitestack - WhiteMist	
5236	👍	2021-10-06 18:00	180	NFV Track 1	NS	Palo Alto - VM-series NGFW Canonical - Charmed OSM Whitestack - WhiteCloud	
5242	👍	2021-10-04 18:00	180	NFV Track 1	NS	Palo Alto - CN-series NGFW DZS - DZS Cloud Whitestack - WhiteMist	
5244	👍	2021-10-12 10:00	180	MEC Track 2	MEC_SERVICES_PROVIDER	Intracom Telecom - uiTOP University of Bologna - UniBo MEC API Tester Huawei - Edge Gallery	
5246	👍	Freestyle			MEC Basic	University of Bologna - UniBo MEC API Tester ETSI - MEC Sandbox	
5250	👍	2021-10-13 15:00	180	NFV Track 2	S-VNFM	A10 Networks - Thunder CGN/FW Ubiqube - MSA Ericsson - EO CM Whitestack - WhiteCloud	
5253	👍	2021-10-13 18:00	180	NFV Track 2	NS	Benu Networks - Cloud BNG Whitestack - WhiteNFV	

Figure 4.3.3-1. 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, on a per test case, test group, test session basis or for overall results. These stats are available in real time for all participants and organisers and allow for tracking the testing progress 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 supported or connected to the HIVE network:

5.1.1 VNFs

Table 5.1.1-1. VNFs Under Test

Organisation	Solution Name	VNF	CNF	Team Location(s)	Short Description
A10 Networks	Thunder CGN/FW	Y	Y	Germany	Full functional CGN and FW application
Benu Networks	Cloud vBNG	N	Y	USA	Virtual Broadband Network Gateway
Palo Alto Networks	VM-series NGFW	Y	N	USA, Canada, Poland	NGFW with GTP-C+GTP-U awareness and correlation
Palo Alto Networks	CN-series NGFW	N	Y	USA, Canada, Poland	NGFW with GTP-C+GTP-U awareness and correlation

5.1.2 VNFMs

Table 5.1.2-1. VNFMs Under Test

Organisations	Solution Name	Team Location(s)	Short Description
Fujitsu	Openstack Tacker	Japan	https://opendev.org/openstack/tacker
UBiqube	MSA	Fance / Ireland	Integrated Automation Platform (IAP)

5.1.3 Orchestrators

Table 5.1.3-1. Orchestrators Under Test

Organisations	Solution Name	VNFM	NFVO	VNF	CNF	Team Location(s)	Short Description
Canonical	Charmed OSM	Y	Y	Y	Y	EU, Canada	Canonical's Charmed Open Source MANO Release TEN
Cisco	NFVO	N	Y	Y	N	CA, USA	Cisco NFVO
DZS	DZS Cloud	Y	Y	Y	Y	India, Boston	https://dzsi.com/cloud-home
Ericsson	Cloud Manager	N	Y	Y	N	Hungary	Ericsson Orchestrator (Cloud Manager only)
Fujitsu	Openstack Tacker	Y	Y	Y	Y	Japan	https://opendev.org/openstack/tacker
Ubiqube	OpenMSA	Y	Y	Y	Y	France, Ireland	Integrated Automation Platform (IAP)
Whitestack	WhiteNFV	Y	Y	Y	Y	Peru, Chile	Whitestack's Open Source MANO Release TEN

	(OSM)						Distribution
Wind River	Studio Conductor	Y	Y	Y	Y	Germany	https://conductor.windriver.com/docs/21.05/

5.1.4 NFVI, VIM, CISM

Table 5.1.4-1. NFVI, VIM, CISM Under Test

Table5.1.4-1. NFVI, VIM, CISM Under Test**Organisations**	**Solution Name**	**LAB**	**OS**	**K8s**	**Team Location**	**Short Description**
Whitestack	WhiteCloud (Openstack) Whitemist (K8s)	OSM Labs	Y	Y	Peru, Chile	OpenStack Stein distribution + Kubernetes 1.16.x
Wind River	WR Studio Cloud Platform (K8s)	Wind River Alameda @OSM Labs	N	Y	CA, USA	WR Cloud Platform - Kubernetes 1.16.x
Wind River	WR Studio Openstack	Wind River Ismaning @OSM Labs	Y	N	Germany	WR Titanium Cloud - OpenStack Pike

5.1.5 MEC Apps

Table 5.1.5-1. MEC Apps Under Test

Table 5.1.5-1. MEC Apps Under Test**Organisations**	**Solution Name**	**Team(s)Location(s)**	**Short Description**
Adlink	Eclipse Zenoh	France	MEC Application containing an Eclipse Zenoh router. https://github.com/eclipse-zenoh
Intracom Telecom	uiTOP	Greece	IoT Device management and Edge Analytics for IOT devices for preparation and pre-processing of raw data from IoT sensors
Intracom Telecom	mMTC Slicing	Greece	In the mMTC slicing solution, a Host IoT GW is responsible for the management of the IoT devices in the field and also realizes IoT GW virtualization by mediating the access of the various vIoT GWs to the underlying infrastructure. The Host IoT GW VNF communicates with the MEC Platform for registering its service.
UniBo	Unibo MEC API Tester	Italy	The Unibo MEC API Tester is a web-based application that can be used to test the capability of a MEC Platform to support the MEC011 defined APIs

5.1.6 MEC Systems

Table 5.1.6-1. MEC Systems Under Test

Organisations	Solution Name	Team(s) Location(s)	Short Description
Equinix	Akraino	USA	The purpose of Public Cloud Edge Interface (PCEI) Blueprint family in Akraino is to specify a set of open APIs for enabling Multi-Domain Inter-working across functional domains that provide Edge capabilities/applications and require close cooperation between the Mobile Edge, the Public Cloud Core and Edge, the 3rd-Party Edge functions as well as the underlying infrastructure such as Data Centers and Networks.
Huawei	EdgeGallery	China	EdgeGallery includes MEP, MEC Manager (minimal management plane, including some functions of MEPM and MEAO), developer tool chain and IaaS/PaaS platform The main functions provided by EdgeGallery are third-party APP development, migration, optimization, integration verification, and simple self-service management.
UBiqube	MSA	France, Ireland	OpenMSA answers the need for multi-domain, multi-vendor service automation. It provides NFV & MEC orchestration ranging from VNF / CNF management up to day2 policy change management and analytics.
Unibo	Unibo MEP	Italy	Unibo MEP provides a containerized MEC Platform that can be used to test the interaction with MEC applications using MEC011
ETSI	MEC sandbox beta	France	MEC Sandbox provides the user with a choice of scenarios combining different network technologies (4G, 5G, Wi-Fi) and terminal types. Combining these assets in a geolocated environment, a user can gain hands-on experience on the behaviour and capabilities of the Location (MEC013), Radio Network Information (MEC012) and WLAN Information (MEC028) service APIs.

5.2 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.

Table 5.2-1. Technical Support

Organisation	Role
Baron	Technical Support
CIRI-ICT, University of Bologna	Technical Support
FScom	Technical Support
Nextworks	Technical Support
Sismondi	Technical Support
xFlow Research	Technical Support

5.3 Observers

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

Table 5.3-1. Observers

Organisation	Role
DOCOMO	Telecommunications service provider
NTT	Telecommunications service provider
Orange	Telecommunications service provider
STC	Telecommunications service provider
Telefonica	Telecommunications service provider

5.4 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:

Table 5.4-1. Supporting Open-Source communities

Community / Project	Role	Details
Akraino	MEC System	https://www.lfedge.org/projects/akraino/
CNCF	Test Suite	https://github.com/cncf/cnf-testsuite
Eclipse Zenoh	MEC App	https://github.com/eclipse-zenoh
ETSI OSM - Open Source MANO	Orchestrator	https://osm.etsi.org
Kubernetes	CISM	https://kubernetes.io/
Openstack	VIM&NFVI	https://www.openstack.org
Openstack Tacker	VNFM	https://opendev.org/openstack/tacker

6 Test Infrastructure

6.1 HIVE

The remote integration, pre-testing, API conformance and multi-vendor interoperability testing were enabled by the NFV Plugtests Programme's HIVE network

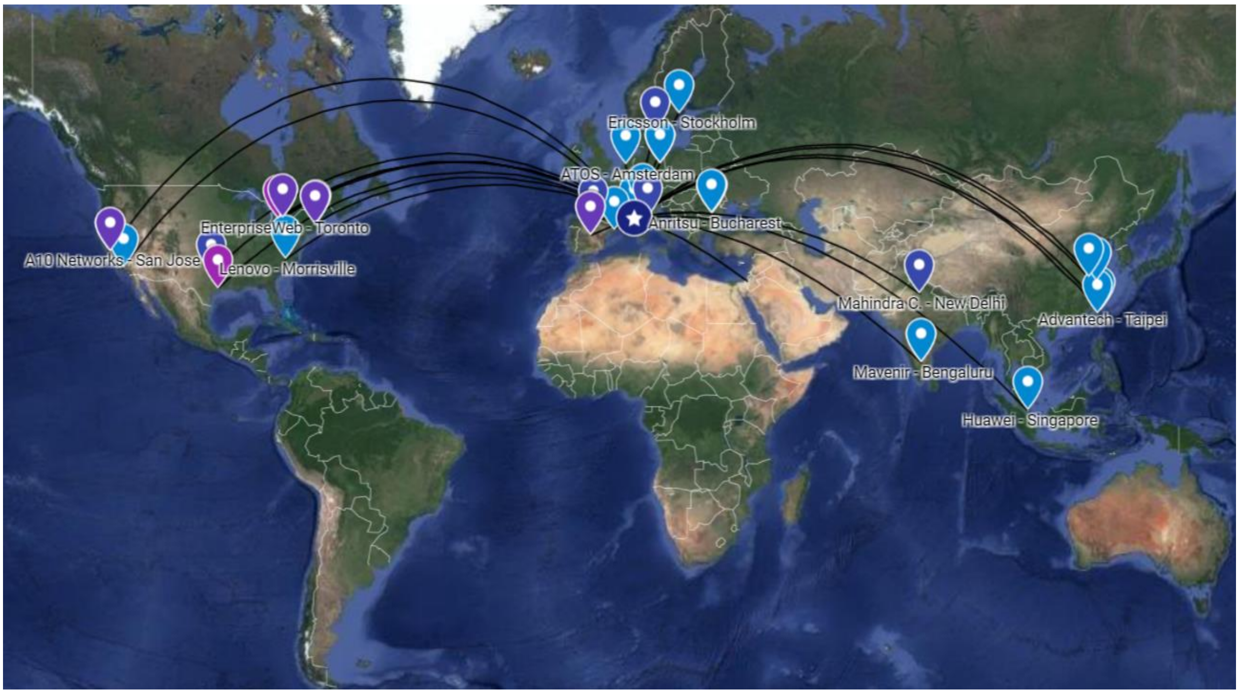


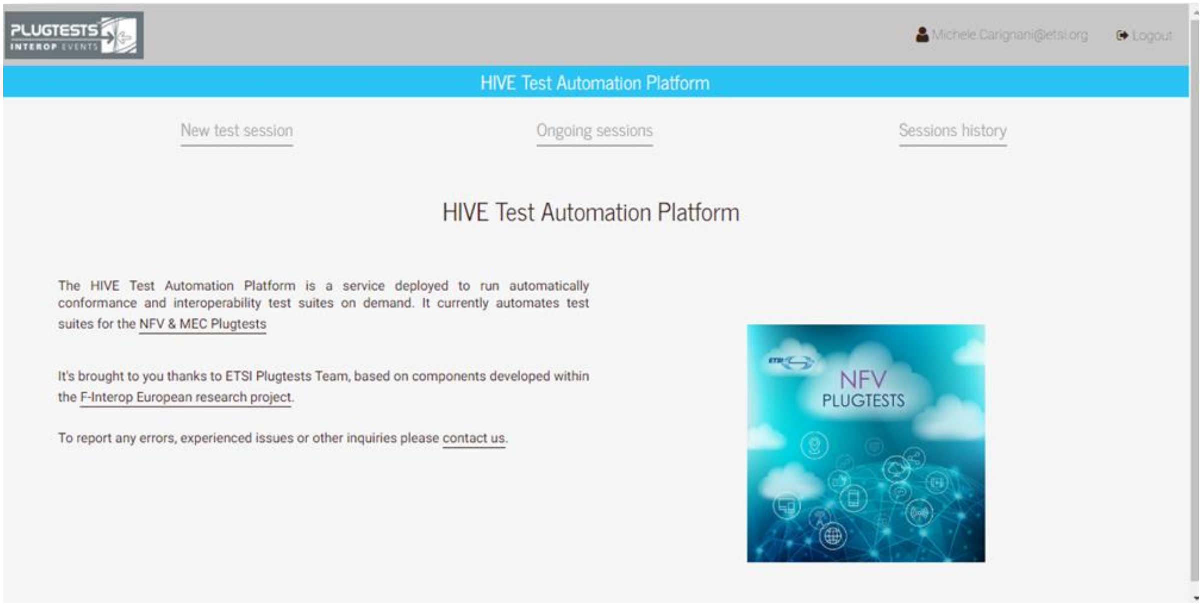
Figure 6.1-1. NFV Plugtests HIVE network

The NFV HIVE (Hub for Interoperability and Validation at ETSI) network securely interconnects participants' remote labs and Functions under Test and allows for remote multi-party interoperability testing and validation activities. Over 20 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.

6.2 Test Automation Platform

The API Conformance Test Sessions require a Test System acting as an API consumer and a Notification Endpoint for the NFV and MEC APIs exposed by NFV and MEC Functions Under Test. The main capabilities required by the test system are:

- Sending configurable HTTP(S) requests
- Allowing custom payloads to be exchanged
- Uploading custom YAML, JSON and ZIP files to be used as request payloads (when applicable)
- Automatically applying headers validation on the response payloads
- Automatically applying schema validation on the response payloads
- Receiving and validating notifications



HIVE Test Automation Platform

The HIVE Test Automation Platform is a service deployed to run automatically conformance and interoperability test suites on demand. It currently automates test suites for the NFV & MEC Plugtests

It's brought to you thanks to ETSI Plugtests Team, based on components developed within the F-Interop European research project.

To report any errors, experienced issues or other inquiries please contact us.

Figure 6.2-1. HIVE TAP - Test Automation Platform

The test system was deployed as a set of Testing Suites in the HIVE Test Automation Platform (TAP), able to run the Robot Framework developed for [NFV-TST010] and [MEC032]. The Platform orchestrates test session executions in all the required steps, including:

- Execution of connectivity pre-checks, to validate the correct networking between the HIVE TAP and the FUT
- Selection of the appropriate Test Tool based on the API Under Test, base specification, and its version
- Configuration of the test system w.r.t Implementation Conformance Statements and implementation details,
- Test selection and execution, with detailed interaction with the user, and
- Test reporting and logging, to enable results collection and issues resolution.

The execution of the tests was triggered on demand by the participants, in a self-service fashion. A demo and a detailed presentation of the new platform was provided to participants during the preparation of the event.

The HIVE TAP acts as a generic test orchestrator, able to transparently execute Test Suites implemented with different frameworks and tailored for different technologies. Ad-hoc components have been developed for NFV and MEC conformance testing, which can be reused in future activities and will be actively developed.

The Testing Tools used in the event were developed on the bases of the Tools already available for previous events, with several new features enabled:

- Extended support for NFV v2.8.1, v3.3.1, and MEC v.2.2.1
- Continuous build and deployment of API test fixes (~hundreds of test suites redeployments)
- A more precise and user-friendly process for selection of individual test groups and test cases
- Several architectural improvements for tool stability and GUI responsiveness

The usage of the HIVE TAP (with respect to the execution of sessions manually operated by a member of the Plugtests Team), led to a higher automation of the test execution which enabled a deeper learning of the Test Suites and the workflow of their execution. The learnings will be contributed back to ISG NFV and MEC (respectively in the TST and DECODE Working Groups) and a set of fixes has been made available in the code base during the event itself.

7 Test Procedures

7.1 Remote Integration Procedures

Remote integration procedures for different FUTs and FUT combinations were documented in the Plugtests WIKI and the progress was captured in a multi-dimensional tracking matrix which was reviewed regularly during the preparation calls.

Before being able to run cross-FUT pre-testing, each type of FUT went through the following steps

7.1.1 VNF, MEC App

1. **HIVE:** Connect to **HIVE** (site-to-site or client-to-site VPN)
2. **IFS:** Fill in the VNF/MEC App Interoperability Feature Statement form
3. **Upload**: Upload VNF / MEC App Package(s), descriptors, artifacts, and SW image(s) to the central VNF Repository / Docker Registry
4. **WIKI**: Create page additional details, including a schema describing the VNF / NS

7.1.2 VNFM, NFVO

- **HIVE:** Connect to **HIVE**. site-to-site VPN connectivity is required for MANO solutions
- **IFS:** Fill in the VNFM/NFVO Interoperability Feature Statements form
- **WIKI**: Create page, including instructions and credentials to access the MANO solution

7.1.3 NFVI, VIM, CISM, MEC Platform

1. **HIVE:** Connect to **HIVE** with site-to-site VPN
2. **VIM Ready**: Install, start and configure the platform
3. **IFS:** Fill in the VIM, CISM, MEC Platform Interoperability Feature Statements form
4. **WIKI**: Create page, including information required for MANO solutions to integrate: NBI IP@, tenants, credentials, VNF Management IP pool, etc...

7.2 NFV IOP Testing Procedure

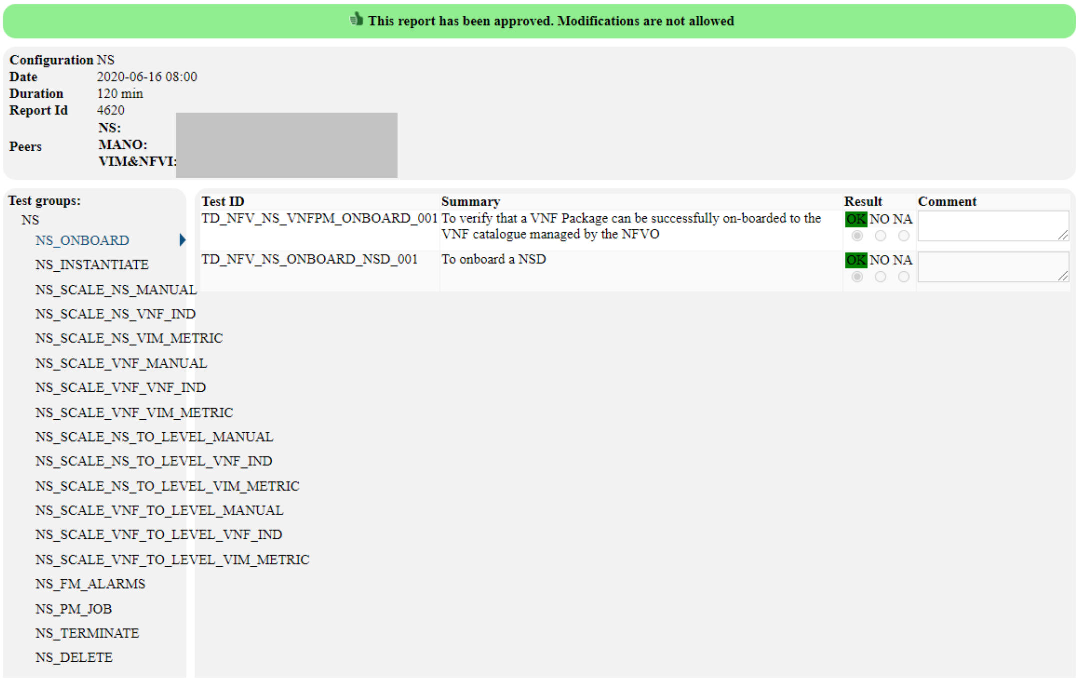
During the Plugtests event, a daily Test Session Schedule was produced with the Plugtests Scheduler. Test Sessions were organised in several parallel tracks, trying to achieve at least one pre-scheduled Test Session every day for each participant. Pre-scheduled test sessions could be completed with additional sessions addressing other test configurations.

Participants could choose to run these additional tests 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/CNFs FUTs representatives create and join a dedicated Slack channel.

2) One representative of the team opens the Test Session Report (as shown in the example below) and the NFV Interoperability Test Plan [NFV-TST007].



This report has been approved. Modifications are not allowed

Configuration	NS
Date	2020-06-16 08:00
Duration	120 min
Report Id	4620
Peers	**NS:** **MANO:** **VIM&NFVI:** [redacted]

Test groups:

NS
- NS_ONBOARD
- NS_INSTANTIATE
- NS_SCALE_NS_MANUAL
- NS_SCALE_NS_VNF_IND
- NS_SCALE_NS_VIM_METRIC
- NS_SCALE_VNF_MANUAL
- NS_SCALE_VNF_VNF_IND
- NS_SCALE_VNF_VIM_METRIC
- NS_SCALE_NS_TO_LEVEL_MANUAL
- NS_SCALE_NS_TO_LEVEL_VNF_IND
- NS_SCALE_NS_TO_LEVEL_VIM_METRIC
- NS_SCALE_VNF_TO_LEVEL_MANUAL
- NS_SCALE_VNF_TO_LEVEL_VNF_IND
- NS_SCALE_VNF_TO_LEVEL_VIM_METRIC
- NS_FM_ALARMS
- NS_PM_JOB
- NS_TERMINATE
- NS_DELETE

Test ID	Summary	Result	Comment
TD_NFV_NS_VNFPM_ONBOARD_001	To verify that a VNF Package can be successfully on-boarded to the VNF catalogue managed by the NFVO	OK NO NA ● ○ ○	
TD_NFV_NS_ONBOARD_NSD_001	To onboard a NSD	OK NO NA ● ○ ○	

Figure 7.2-1. Test Session Report

3) For each test in each Test Group of the Test Plan listed in the session opened in TRT:

a. The corresponding Test Descriptions (identified in TRT with Test ID and a summary description of the test to be executed) are considered for the target SUT Configuration (see the example below)

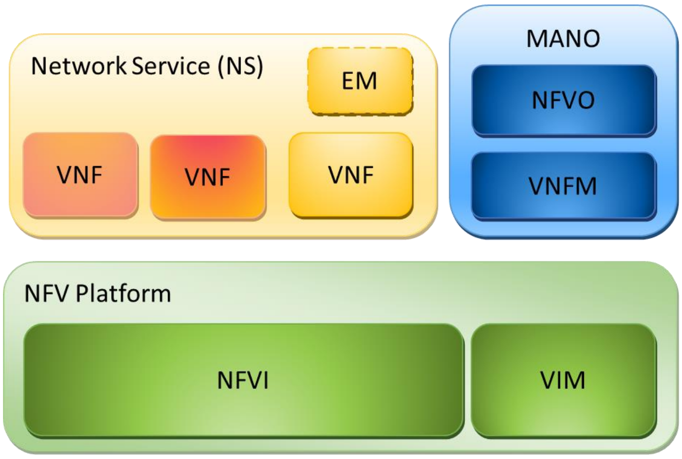


Figure 7.2-2. SUT Configuration example.

b. MANO, VIM&NFVI(s), and VNFs/CNFs providers jointly execute the different steps specified in the Test Description and evaluate interoperability through the different IOP Checks listed (see the Test Description below).

Table 7.2-1. Test Description example

Test Description: NS scale out with an operator action	
Identifier	TD_NFV_NSLCM_SCALE_OUT_001
Test Purpose	Verify that the NS can be successfully scaled out by adding VNF instances triggered by an operator action
Configuration	SUT Configuration 1
References	[NFV-IFA 005], [NFV-IFA 006], [NFV-IFA 007], [NFV-IFA 008], [NFV-IFA 010] and [NFV-IFA 013]

Test Description: NS scale out with an operator action				
Applicability	• NFVO/VNFM can generate "allocate compute resource" operation requests to the VIM (NFVO_CRM_ALLOCATE or VNFM_CRM_ALLOCATE) • NFVO/VNFM can generate "allocate network resource" operation requests to the VIM (NFVO_NRM_ALLOCATE or VNFM_NRM_ALLOCATE) • NFVO/VNFM can generate "allocate storage resource" operation requests to the VIM (NFVO_SRM_ALLOCATE or VNFM_SRM_ALLOCATE) • VIM supports "allocate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_ALLOCATE_BY_NFVO or VIM_CRM_ALLOCATE_BY_VNFM) • VIM supports "allocate network resource" operation requests from the NFVO/VNFM (VIM_NRM_ALLOCATE_BY_NFVO or VIM_NRM_ALLOCATE_BY_VNFM) • VIM supports "allocate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_ALLOCATE_BY_NFVO or VIM_SRM_ALLOCATE_BY_VNFM) • NFVO can generate "scale out by adding VNF instances" requests to the VNFM (NFVO_VNFLCM_NS_SCALE_OUT) • VNFM supports "scale out by adding VNF instances" requests from the NFVO (VNFM_VNFLCM_NS_SCALE_OUT)			
Pre-test conditions	• NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001) • NFVI has the required amount of consumable virtual resources to run the scaled-out NS			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger NS scale out by adding VNF instances to the NS in NFVO with an operator action	
	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 sends resource allocation request using resource provider ID through the NFVO to manage the instantiation of the VNFs composing the given NS	
	4	IOP Check	Verify that the additional VNF instance(s) have been deployed by querying the VNFM	
	5	IOP Check	Verify that the additional resources have been allocated by the VIM according to the descriptors	
	6	IOP Check	Verify that the additional VNF instance(s) are running and reachable via their management network	
	7	IOP Check	Verify that the additional VNF instances(s) have been configured according to the descriptors by querying the VNFM	
	8	IOP Check	Verify that the additional VNF instances(s), VL(s) and VNFFG(s) are connected according to the descriptors	
	9	IOP Check	Verify that the NFVO indicates the scaling operation result as successful	
	10	IOP Check	Verify that NS has been scaled out by running the end-to-end functional test factoring the VNF scale and capacity	
IOP Verdict				

c. The Test Result is reported to the Test Session Report in TRT, as follows:

i. OK: all IOP Checks were successful

ii. NO: 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 are executed and results recorded, all the involved participants review the Report and approve it.

7.3 MEC IOP Testing Procedure

The MEC Interoperability test sessions were organized among participants according to the capability of their FUTs and the test cases selected by the participants. The test sessions ran in parallel with sessions for other tracks.

Organized sessions were allocated in the Plugtests Scheduler for the coordination of participants. For each test session, a specific test configuration was assigned, and a set of multi-vendor components was selected for it.

Test sessions were executed remotely and facilitated by the use of communication channels such as live chats (with dedicated private spaces for individual sessions) and audio/video conferencing tools (provided either by the Plugtests Team or the participants). Given the novelty of the MEC Track, the Plugtests team participated in almost all the test sessions to provide support and coordinate the actions.

For each session, the same methodology of NFV Interoperability track was executed, i.e.:

- The test plan and the test descriptions applicable to the test session were reviewed,
- Each test description was jointly executed by the participants, by executing stimuli and verifications,
- Outcomes of each test session were reported and validated by the participants.

7.4 API Testing Procedure

The NFV API Conformance Test Sessions aimed at validating the conformance of the participating FUTs to the [NFV-SOL002], [NFV-SOL003] and [NFV-SOL005] API specifications, while validating the API Conformance Robot Test Suite as well. The Test System was run on the HIVE TAP, which provided the connectivity to the participating FUTs. The Test System executed the Robot Framework Test Suites developed for the NFV API Conformance Test Specification [NFV-TST010] - in different API specifications versions: published v2.4.1, v2.6.1 and v.2.7.1 and stable drafts v.2.8.1 and v.3.3.1 versions made available via the ETSI Forge [NFV-ROBOT-TS]. For MEC API conformance, the Test System executed the Robot Framework and TTCN-3 Test Suites developed for the MEC API Conformance Test Specification [MEC-DEC032-3], in both v2.1.1 and v2.2.1.

Each test session was executed on-demand and in a self-service fashion, to allow maximum flexibility and scalability of execution resources (e.g., w.r.t. time zone differences, number of parallel sessions etc.).

Within each test session, the user would – via the workflow implemented in HIVE TAP – execute the following steps:

1. Log into the HIVE TAP, with credentials created for each participating team;
2. Select the API (i.e., NFV Interface) to be tested, (e.g., NS LifecycleManagement over [NFV-SOL005]).
3. Fill in the configuration of the test system, (i.e., providing values for the variables defined in the Robot resource files). The variables were automatically collected from the Robot test suite and presented in a form for the user, who could fill them in individually or as a JSON data structure (enabling reuse of configuration settings);
4. After activation of the test session (which comprised initialization of a dedicated test environment and instantiation of the specifically configured test system), the user could execute and skip individual groups of tests within the Test Suite for the selected API. Groups were defined by the individual Robot files in the test suite.
5. After the execution of each group of tests, the user was presented with the possibility to download the detailed test reports as produced by the Robot executor, in both human readable (HTML) and machine readable (XML) formats.
6. User is presented with the option to commit or discard the results generated for the group of tests.
7. After the execution of all tests groups, user is presented with the option to download a zip file containing all results (HTML and XML) and the configuration (from step 3, as JSON file).
8. After execution of all Robot files the user was presented with the option to commit their test results

9. The user was presented with the possibility to restart or terminate the test sessions.

10. ‘Committed’ results were automatically collected by the platform to allow the generation of aggregated statistics. The user was allowed to execute, and commit results several times, even for a same API conformance test. When several commits occurred for a same API test group, the last committed result prevailed.

8 Test Plans Overview

8.1 NFV Interoperability

The NFV Interoperability test sessions were based on [NFV-TST007]. 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, results collection and analysis.

8.1.1 NS

The NS Configuration was based on the “SUT Configuration 1” in [NFV-TST007]. It involves one MANO solution, one VIM&NFVI and one Network Service (NS), including just one on several VNFs.

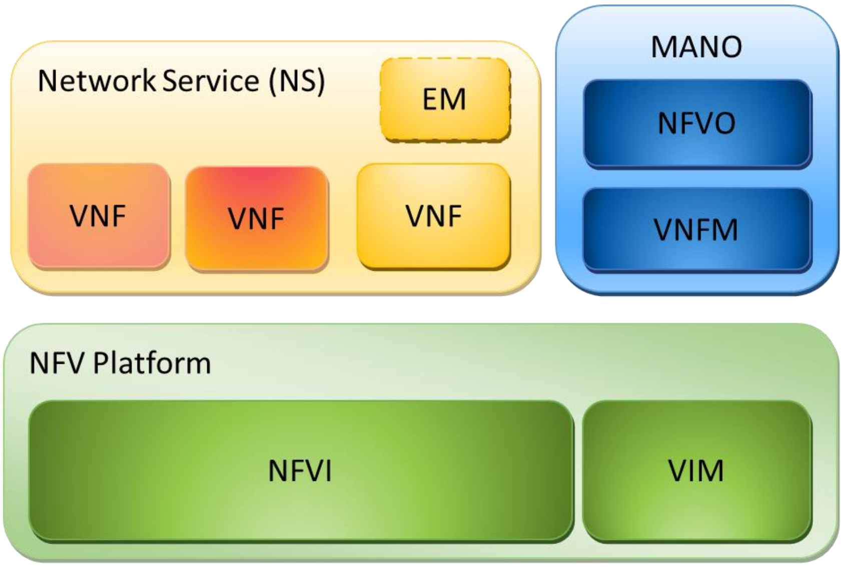


Figure 8.1.1-1: NS SUT Configurations

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

Table 8.1.1-1. NS Test Groups

Group	TST007 Test IDs
NS_ONBOARD	TD_NFV_VNFPM_ONBOARD_001
NS_INSTANTIATE	TD_NFV_NSLCM_INSTANTIATE_001
NS_SCALE_MANUAL	TD_NFV_NSLCM_SCALE_OUT_001 TD_NFV_NSLCM_SCALE_IN_001
NS_SCALE_VNF_IND	TD_NFV_NSLCM_SCALE_OUT_002 TD_NFV_NSLCM_SCALE_IN_002 TD_NFV_NSLCM_SCALE_OUT_004 TD_NFV_NSLCM_SCALE_IN_004
NS_SCALE_VIM_METRIC	TD_NFV_NSLCM_SCALE_OUT_003 TD_NFV_NSLCM_SCALE_IN_003

NS_SCALE_VNF_MANUAL	TD_NFV_NSLCM_SCALE_OUT_VNF_001 TD_NFV_NSLCM_SCALE_IN_VNF_001
NS_SCALE_VNF_VNF_IND	TD_NFV_NSLCM_SCALE_OUT_VNF_002 TD_NFV_NSLCM_SCALE_IN_VNF_002 TD_NFV_NSLCM_SCALE_OUT_VNF_004 TD_NFV_NSLCM_SCALE_IN_VNF_004
NS_SCALE_VNF_VIM_METRIC	TD_NFV_NSLCM_SCALE_OUT_VNF_003 TD_NFV_NSLCM_SCALE_IN_VNF_003
NS_SCALE_TO_LEVEL_MANUAL	TD_NFV_NSLCM_SCALE_TO_LEVEL_001
NS_SCALE_TO_LEVEL_VNF_IND	TD_NFV_NSLCM_SCALE_TO_LEVEL_002
NS_SCALE_TO_LEVEL_VIM_METRIC	TD_NFV_NSLCM_SCALE_TO_LEVEL_003
NS_SCALE_VNF_TO_LEVEL_MANUAL	TD_NFV_NSLCM_SCALE_TO_LEVEL_VNF_001
NS_SCALE_VNF_TO_LEVEL_VNF_IND	TD_NFV_NSLCM_SCALE_TO_LEVEL_VNF_002
NS_SCALE_VNF_TO_LEVEL_VIM_METRIC	TD_NFV_NSLCM_SCALE_TO_LEVEL_VNF_003
NS_UPDATE_VNF_OPERATE	TD_NFV_NSLCM_UPDATE_START_001 TD_NFV_NSLCM_UPDATE_STOP_001
NS_UPDATE_VNF_CONFIG	TD_NFV_NSLCM_UPDATE_VNF_CONFIG_001
NS_UPDATE_VNF_DF	TD_NFV_NSLCM_UPDATE_VNF_DF_001
NS_FM_SUBSCRIPTION	TD_NFV_FM_NS_ALARM_SUBSCRIPTION_CREATE_001 TD_NFV_FM_NS_ALARM_SUBSCRIPTION_DELETE_001 TD_NFV_FM_NS_ALARM_NOTIFICATION_001 TD_NFV_FM_NS_ALARM_CLEAR_NOTIFICATION_001
NS_FM_ALARMS	TD_NFV_FM_NS_ALARM_QUERY_001
NS_PM_JOB	TD_NFV_PM_NS_MONITORING_JOB_CREATE_001 TD_NFV_PM_NS_PERFORMANCE_METRICS_QUERY_001 TD_NFV_PM_NS_MONITORING_INFO_NOTIFICATION_001 TD_NFV_PM_NS_MONITORING_JOB_DELETE_001
NS_PM_THRESHOLD	TD_NFV_PM_NS_THRESHOLD_CREATE_001 TD_NFV_PM_NS_THRESHOLD_NOTIFICATION_001 TD_NFV_PM_NS_THRESHOLD_DELETE_001
NS_TERMINATE	TD_NFV_NSLCM_TERMINATE_001
NS_DELETE	TD_NFV_VNFPM_DELETE_001

The full list and detailed steps of these NFV IOP Test Descriptions can be found in [NFV-TST007].

8.1.2 NS CNF

The NS-CNF group was based on the “SUT Configuration 1” in [NFV-TST007]. It involves one MANO solution, one VIM&NFVI and one Network Service (NS), including one or several CNFs.

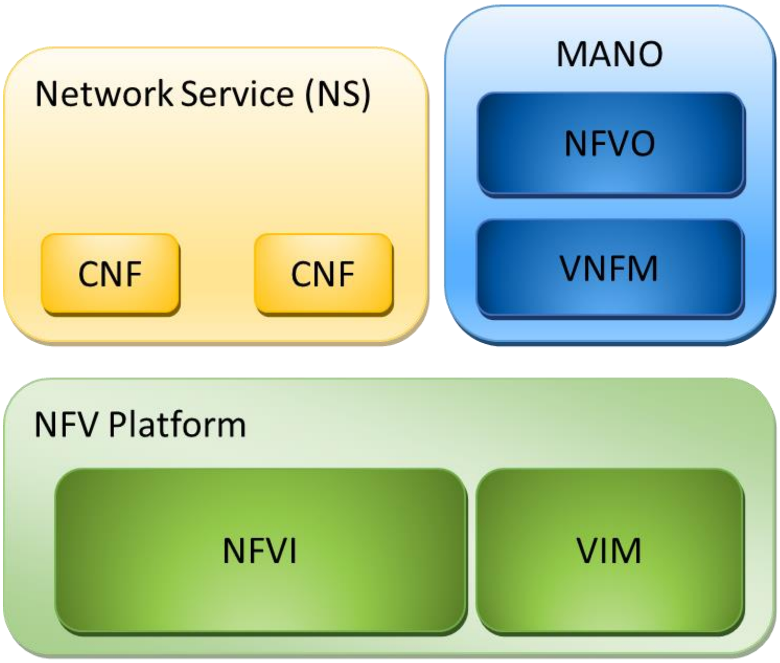


Figure 8.1.2-1: NS-CNF SUT Configurations

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

Table 8.1.2-1. NS-CNF Test Groups

Group	TST007 Test IDs
NS-CNF_ONBOARD	TD_NFV_VNFPM_ONBOARD_001
NS-CNF_INSTANTIATE	TD_NFV_NSLCM_INSTANTIATE_001
NS-CNF_SCALE_MANUAL	TD_NFV_NSLCM_SCALE_OUT_001 TD_NFV_NSLCM_SCALE_IN_001
NS-CNF_SCALE_VNF_IND	TD_NFV_NSLCM_SCALE_OUT_002 TD_NFV_NSLCM_SCALE_IN_002 TD_NFV_NSLCM_SCALE_OUT_004 TD_NFV_NSLCM_SCALE_IN_004
NS-CNF_SCALE_VIM_METRIC	TD_NFV_NSLCM_SCALE_OUT_003 TD_NFV_NSLCM_SCALE_IN_003
NS-CNF_SCALE_VNF_MANUAL	TD_NFV_NSLCM_SCALE_OUT_VNF_001 TD_NFV_NSLCM_SCALE_IN_VNF_001
NS-CNF_SCALE_VNF_VNF_IND	TD_NFV_NSLCM_SCALE_OUT_VNF_002 TD_NFV_NSLCM_SCALE_IN_VNF_002 TD_NFV_NSLCM_SCALE_OUT_VNF_004 TD_NFV_NSLCM_SCALE_IN_VNF_004
NS-CNF_SCALE_VNF_VIM_METRIC	TD_NFV_NSLCM_SCALE_OUT_VNF_003 TD_NFV_NSLCM_SCALE_IN_VNF_003

NS-CNF_SCALE_TO_LEVEL_MANUAL	TD_NFV_NSLCM_SCALE_TO_LEVEL_001
NS-CNF_SCALE_TO_LEVEL_VNF_IND	TD_NFV_NSLCM_SCALE_TO_LEVEL_002
NS-CNF_SCALE_TO_LEVEL_VIM_METRIC	TD_NFV_NSLCM_SCALE_TO_LEVEL_003
NS-CNF_SCALE_VNF_TO_LEVEL_MANUAL	TD_NFV_NSLCM_SCALE_TO_LEVEL_VNF_001
NS-CNF_SCALE_VNF_TO_LEVEL_VNF_IND	TD_NFV_NSLCM_SCALE_TO_LEVEL_VNF_002
NS-CNF_SCALE_VNF_TO_LEVEL_VIM_METRIC	TD_NFV_NSLCM_SCALE_TO_LEVEL_VNF_003
NS-CNF_UPDATE_VNF_OPERATE	TD_NFV_NSLCM_UPDATE_START_001 TD_NFV_NSLCM_UPDATE_STOP_001
NS-CNF_UPDATE_VNF_CONFIG	TD_NFV_NSLCM_UPDATE_VNF_CONFIG_001
NS-CNF_UPDATE_VNF_DF	TD_NFV_NSLCM_UPDATE_VNF_DF_001
NS-CNF_FM_SUBSCRIPTION	TD_NFV_FM_NS_ALARM_SUBSCRIPTION_CREATE_001 TD_NFV_FM_NS_ALARM_SUBSCRIPTION_DELETE_001 TD_NFV_FM_NS_ALARM_NOTIFICATION_001 TD_NFV_FM_NS_ALARM_CLEAR_NOTIFICATION_001
NS-CNF_FM_ALARMS	TD_NFV_FM_NS_ALARM_QUERY_001
NS-CNF_PM_JOB	TD_NFV_PM_NS_MONITORING_JOB_CREATE_001 TD_NFV_PM_NS_PERFORMANCE_METRICS_QUERY_001 TD_NFV_PM_NS_MONITORING_INFO_NOTIFICATION_001 TD_NFV_PM_NS_MONITORING_JOB_DELETE_001
NS-CNF_PM_THRESHOLD	TD_NFV_PM_NS_THRESHOLD_CREATE_001 TD_NFV_PM_NS_THRESHOLD_NOTIFICATION_001 TD_NFV_PM_NS_THRESHOLD_DELETE_001
NS-CNF_TERMINATE	TD_NFV_NSLCM_TERMINATE_001
NS-CNF_DELETE	TD_NFV_VNFPM_DELETE_001

The full list and detailed steps of these NFV IOP Test Descriptions can be found in [NFV-TST007].

8.1.3 Specific VNFM

The Specific VNFM group leverages the “SUT Configuration 1” as described in [NFV-TST007].

This configuration involved one MANO solution (providing a standalone NFVO), one VIM&NFVI, one VNF and a standalone VNF Manager. The Specific VNFM and the NFVO in the MANO were requested to both support the same mode (direct or indirect) for resource management

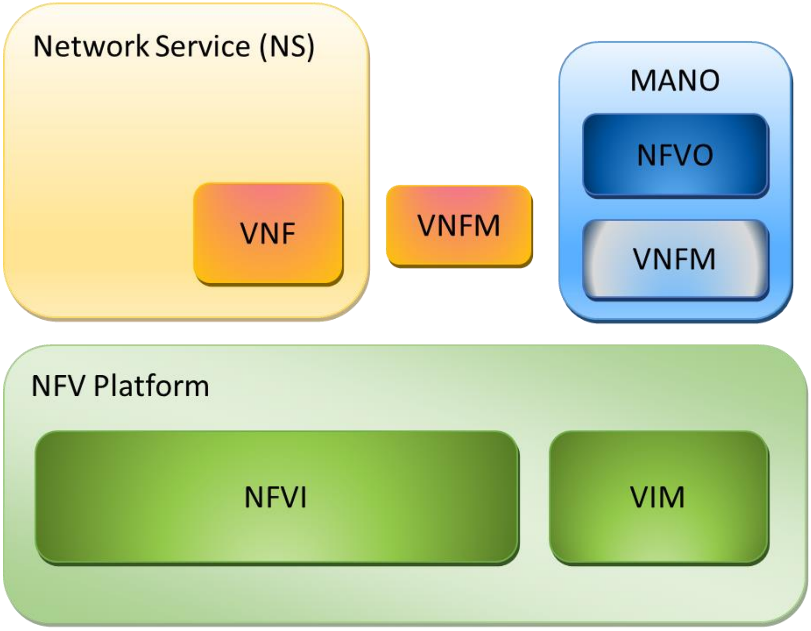


Figure 8.1.3-1: Specific VNFM SUT Configuration

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

Table 8.1.3-1. Specific VNFM Test Groups

Group	TST007 Test IDs
S-VNFM_NS_ONBOARD	TD_NFV_VNFPM_ONBOARD_001
S-VNFM_NS_INSTANTIATE	TD_NFV_NSLCM_INSTANTIATE_001
S-VNFM_NS_SCALE_MANUAL	TD_NFV_NSLCM_SCALE_OUT_001 TD_NFV_NSLCM_SCALE_IN_001
S-VNFM_NS_SCALE_VNF_IND	TD_NFV_NSLCM_SCALE_OUT_002 TD_NFV_NSLCM_SCALE_IN_002 TD_NFV_NSLCM_SCALE_OUT_004 TD_NFV_NSLCM_SCALE_IN_004
S-VNFM_NS_SCALE_VIM_METRIC	TD_NFV_NSLCM_SCALE_OUT_003 TD_NFV_NSLCM_SCALE_IN_003
S-VNFM_NS_SCALE_VNF_MANUAL	TD_NFV_NSLCM_SCALE_OUT_VNF_001 TD_NFV_NSLCM_SCALE_IN_VNF_001
S-VNFM_NS_SCALE_VNF_VNF_IND	TD_NFV_NSLCM_SCALE_OUT_VNF_002 TD_NFV_NSLCM_SCALE_IN_VNF_002 TD_NFV_NSLCM_SCALE_OUT_VNF_004 TD_NFV_NSLCM_SCALE_IN_VNF_004
S-VNFM_NS_SCALE_VNF_VIM_METRIC	TD_NFV_NSLCM_SCALE_OUT_VNF_003 TD_NFV_NSLCM_SCALE_IN_VNF_003
S-VNFM_NS_SCALE_TO_LEVEL_MANUAL	TD_NFV_NSLCM_SCALE_TO_LEVEL_001
S-VNFM_NS_SCALE_TO_LEVEL_VNF_IND	TD_NFV_NSLCM_SCALE_TO_LEVEL_002
S-VNFM_NS_SCALE_TO_LEVEL_VIM_METRIC	TD_NFV_NSLCM_SCALE_TO_LEVEL_003
S-VNFM_NS_SCALE_VNF_TO_LEVEL_MANUAL	TD_NFV_NSLCM_SCALE_TO_LEVEL_VNF_001

S-VNFM_NS_SCALE_VNF_TO_LEVEL_VNF_IND	TD_NFV_NSLCM_SCALE_TO_LEVEL_VNF_002
S-VNFM_NS_SCALE_VNF_TO_LEVEL_VIM_METRIC	TD_NFV_NSLCM_SCALE_TO_LEVEL_VNF_003
S-VNFM_NS_UPDATE_VNF_OPERATE	TD_NFV_NSLCM_UPDATE_START_001 TD_NFV_NSLCM_UPDATE_STOP_001
S-VNFM_NS_UPDATE_VNF_CONFIG	TD_NFV_NSLCM_UPDATE_VNF_CONFIG_001
S-VNFM_NS_UPDATE_VNF_DF	TD_NFV_NSLCM_UPDATE_VNF_DF_001
S-VNFM_NS_FM_SUBSCRIPTION	TD_NFV_FM_NS_ALARM_SUBSCRIPTION_CREATE_001 TD_NFV_FM_NS_ALARM_SUBSCRIPTION_DELETE_001 TD_NFV_FM_NS_ALARM_NOTIFICATION_001 TD_NFV_FM_NS_ALARM_CLEAR_NOTIFICATION_001
S-VNFM_NS_FM_ALARMS	TD_NFV_FM_NS_ALARM_QUERY_001
S-VNFM_NS_PM_JOB	TD_NFV_PM_NS_MONITORING_JOB_CREATE_001 TD_NFV_PM_NS_PERFORMANCE_METRICS_QUERY_001 TD_NFV_PM_NS_MONITORING_INFO_NOTIFICATION_001 TD_NFV_PM_NS_MONITORING_JOB_DELETE_001
S-VNFM_NS_PM_THRESHOLD	TD_NFV_PM_NS_THRESHOLD_CREATE_001 TD_NFV_PM_NS_THRESHOLD_NOTIFICATION_001 TD_NFV_PM_NS_THRESHOLD_DELETE_001
S-VNFM_NS_TERMINATE	TD_NFV_NSLCM_TERMINATE_001
S-VNFM_NS_DELETE	TD_NFV_VNFPM_DELETE_001

The full list and detailed steps of these NFV IOP Test Descriptions can be found in [NFV-TST007].

8.2 NFV API Conformance

This NFV API Conformance test plan was based on the Robot Framework Test Cases developed for [NFV-TST010] NFV API Conformance Test Specification, addressing FUT API Conformance to [NFV-SOL002], [NFV-SOL003] and [NFV-SOL005] specifications. In particular, for this Plugtests, five NFV API Conformance test specifications versions were made available to the participants for their tests:

- [NFV-TST010] v2.4.1, with NFV API conformance tests for [NFV-SOL002], [NFV-SOL003] and [NFV-SOL005] v2.4.1
- [NFV-TST010] v2.6.1 with NFV API conformance tests for [NFV-SOL002], [NFV-SOL003] and [NFV-SOL005] v2.6.1
- [NFV-TST010] v2.7.1 with NFV API conformance tests for [NFV-SOL002], [NFV-SOL003] and [NFV-SOL005] v2.7.1
- Stable version of [NFV-TST010] v2.8.1, with preliminary NFV API conformance tests for [NFV-SOL002], [NFV-SOL003] and [NFV-SOL005] v2.8.1
- Stable version of [NFV-TST010] v3.3.1, with preliminary NFV API conformance tests for [NFV-SOL002], [NFV-SOL003], [NFV-SOL005], [NFV-SOL009], [NFV-SOL011] and [NFV-SOL012] v3.3.1

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

The following clauses summarise the test cases in scope for this Plugtests, grouped by FUT type. As none of the participants bringing NFs (as detailed in Table 1) were providing support or implementation of the [NFV-SOL002] APIs exposed by the VNFs (i.e., VNF Configuration and VNF Indicator APIs), the next clauses refer to VNFM and NFVO FUT types only.

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

- V2.4.1 https://forge.etsi.org/rep/nfv/api-tests/tree/2.4.1-fix-plu
- V2.6.1 https://forge.etsi.org/rep/nfv/api-tests/tree/2.6.1-fix-plu
- V2.7.1 https://forge.etsi.org/rep/nfv/api-tests/tree/2.7.1-fix-plu
- V2.8.1 https://forge.etsi.org/rep/nfv/api-tests/tree/2.8.1-fix-plu
- V3.3.1 https://forge.etsi.org/rep/nfv/api-tests/tree/3.3.1-fix-plu

8.2.1 VNFM

The VNFM APIs were tested following the test configuration shown in the figure below. In particular, two set of APIs were in scope for this NFV&MEC API Plugtests:

- NFV-SOL002 APIs, exposed by the VNFM and consumed by the test system (HIVE TAP) acting as VNF/EM
- NFV-SOL003 APIs, exposed by the VNFM and consumed by the test system (HIVE TAP) acting as NFVO

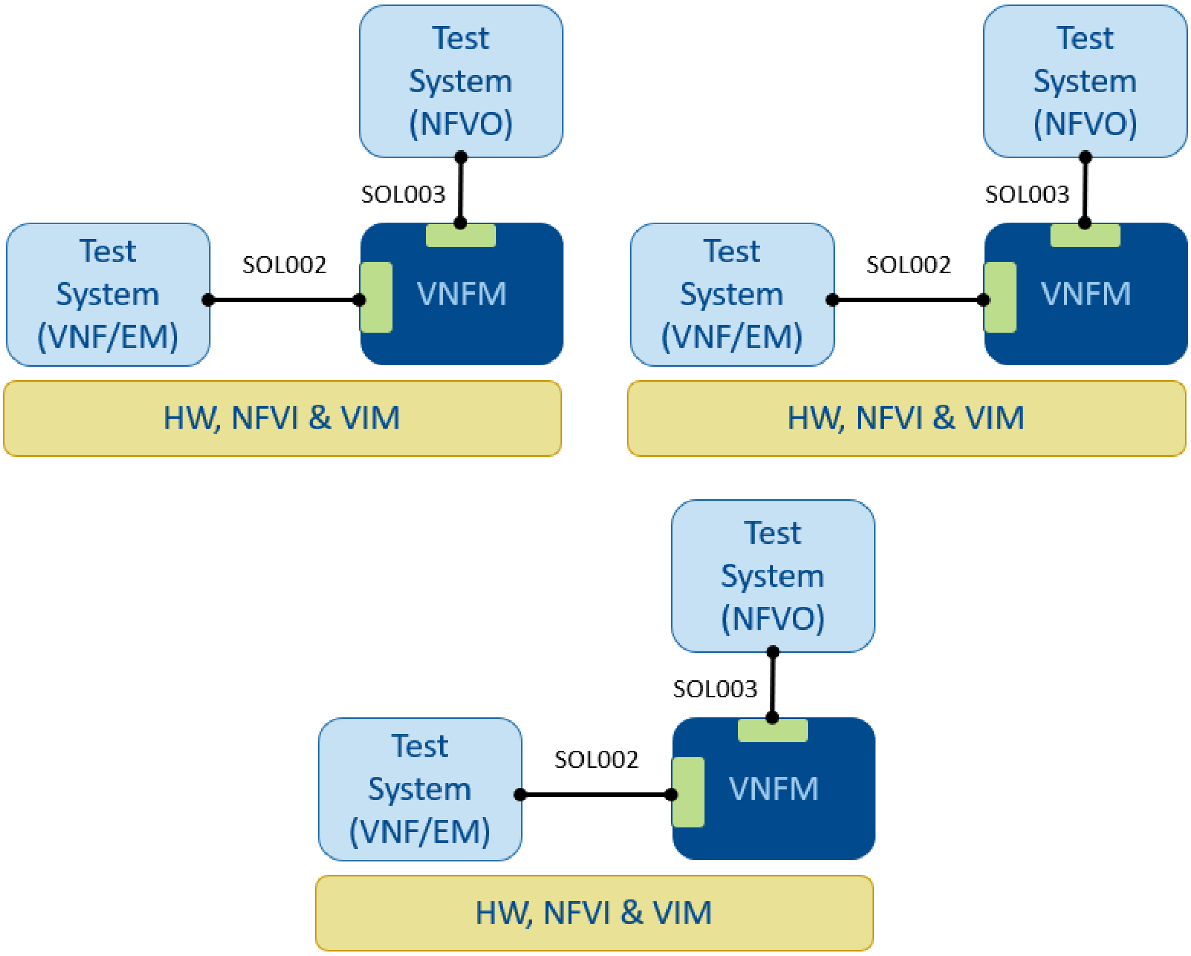


Figure 8.2.1-1: VNFM APIs Test Configuration

8.2.1.1 NFV-SOL002

The following subset of the [NFV-TST010] v2.6.1 Test Suites for [NFV-SOL002] APIs exposed by VNFMs was run during this Plugtests.

Table 8.2.1.1-1. VNFM SOL002 API tests suites

VNFM SOL002 API	Version	[NFV-TST010] Clause
VNF Life Cycle Management API	v2.6.1	6.3.5 (Annex E)

8.2.1.2 NFV-SOL003

The following subset of the [NFV-TST010] v2.6.1 Test Suites for [NFV-SOL003] APIs exposed by VNFMs was run during this Plugtests.

Table 8.2.1.2-1. VNFM SOL003 API tests suites

VNFM SOL003 API	Version	[NFV-TST010] Clause
VNF Life Cycle Management API	v2.6.1	7.3.1 (Annex F)

8.2.2 NFVO

The NFVO APIs were tested following the test configuration shown in the figure below. In particular, two set of APIs were in scope for this Plugtests:

- [NFV-SOL003] APIs, exposed by the NFVO and consumed by the Robot Framework test system acting as VNFM
- [NFV-SOL005] APIs, exposed by the NFVO and consumed by the Robot Framework test system acting as OSS/BSS

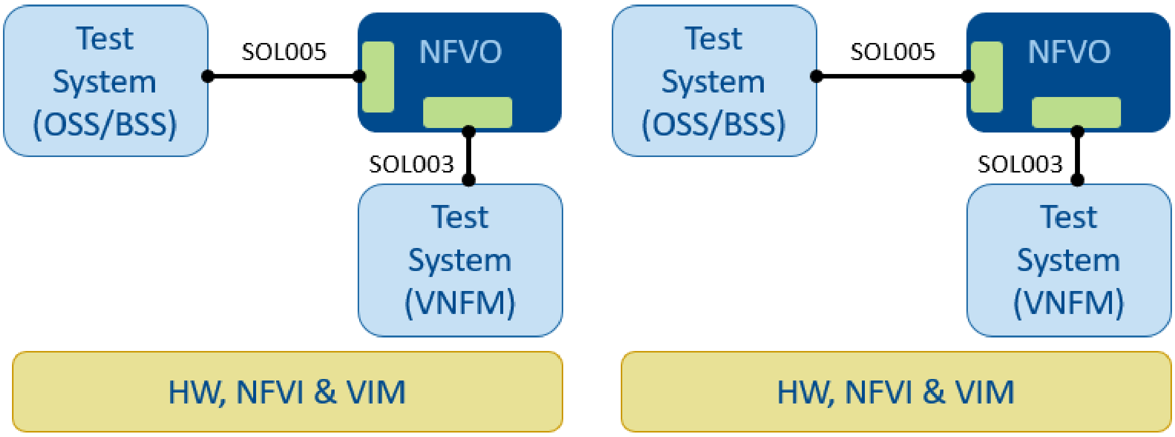


Figure 8.2.2-1: NFVO APIs Test Configuration

8.2.2.1 NFV-SOL005

The following subset of the [NFV-TST010] v2.4.1 and v2.6.1 Test Suites for [NFV-SOL005] APIs exposed by NFVOs was run during this Plugtests.

Table 8.2.2.1-1. NFVO SOL005 API tests suites

NFVO SOL005 API	Version	[NFV-TST010] Clause
NS Lifecycle Management API	v2.4.1	5.3.2
VNF Package Management API	v2.6.1	5.3.5

8.3 MEC Interoperability

The MEC Interoperability test sessions were based on a dedicated test plan developed during the Plugtests preparation [MEC-IOP-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.

8.3.1 MEC Basic

The SUT_MEC_BASIC configuration includes a single MEC application along with a MEC platform. In those tests, the term “MEC Platform” is used to indicate any of the following components: MEC platform, MEC orchestrator or MEC platform manager. The providers of other components of the MEC system such as MEO or MEPM are out of scope. The MEC application runs – together with the MEC Platform - on the MEC host or the NFVI.

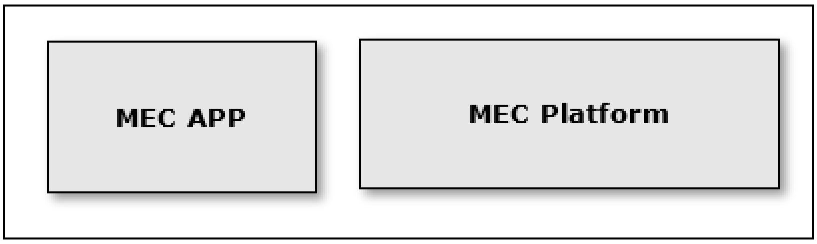


Figure 8.3.1-1: Test configuration SUT_MEC_BASIC

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

Table 8.3.1-1. MEC Basic Test Groups

Group	Test IDs	# TDs
MEC Application Lifecycle	TD_MEC_APP_ONBOARD TD_MEC_APP_START TD_MEC_APP_STOP TD_MEC_APP_STATUS TD_MEC_APP_CHANGE	5
MEC Services	TD_MEC_SVC_QUERY TD_MEC_SVC_REGISTER TD_MEC_SVC_UPDATE TD_MEC_SVC_DEREGISTER TD_MEC_SVC_QUERYTIME TD_MEC_SVC_TRANSPORTS	7
MEC Traffic	TD_MEC_NTW_ACTIVATE TD_MEC_NTW_UPDATE TD_MEC_NTW_DEACTIVATE TD_MEC_NTW_DNS_ACTIVATE	5

	TD_MEC_NTW_DNS_DEACTIVATE	

The complete list of Test Cases can be found in [MEC-IOP-TP] clause 8

8.3.2 MEC Services with Single App

The SUT_MEC_SERVICES_SINGLE_APP test configuration is similar to the configuration SUT_MEC_BASIC, with a difference on the integration between the two elements. In this configuration, one (1) MEC application runs within the MEC Host alongside the MEC platform. The configuration focuses on the capabilities around MEC Services such as the capability of applications and the platform to provide and register. The service is registered and available for discovery through the service registry in the MEC platform.

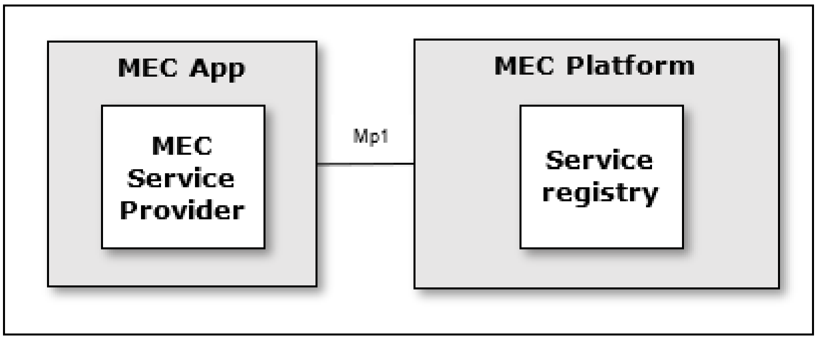


Figure 8.3.2-1: Test configuration SUT_MEC_SERVICE_SINGLE_APP

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

Table 8.3.2-1. MEC Services (Single App) Test Groups

Group	Test IDs	# TDs
MEC Application Lifecycle	TD_MEC_APP_ONBOARD TD_MEC_APP_START TD_MEC_APP_STOP TD_MEC_APP_STATUS TD_MEC_APP_CHANGE	5
MEC Services	TD_MEC_SVC_QUERY TD_MEC_SVC_REGISTER TD_MEC_SVC_UPDATE TD_MEC_SVC_DEREGISTER TD_MEC_SVC_CONSUME TD_MEC_SVC_QUERYTIME TD_MEC_SVC_TRANSPORTS	7
MEC Location API	TD_MEC_LOC_UE_LKP_1 TD_MEC_LOC_UE_LKP_2 TD_MEC_LOC_UE_INF_LKP_1 TD_MEC_LOC_UE_INF_LKP_2 TD_MEC_LOC_UE_SUB_1 TD_MEC_LOC_UE_SUB_2 TD_MEC_LOC_INF_SUB_1 TD_MEC_LOC_INF_SUB_2 TD_MEC_LOC_RNL TD_MEC_LOC_TRACK TD_MEC_LOC_DIST_1	18

	TD_MEC_LOC_DIST_2 TD_MEC_LOC_DIST_SUB_1 TD_MEC_LOC_DIST_SUB_2 TD_MEC_LOC_DIST_SUB_3 TD_MEC_LOC_AREA_SUB_1 TD_MEC_LOC_AREA_SUB_2 TD_MEC_LOC_AREA_SUB_3	

The complete list of Test Cases can be found in [MEC-IOP-TP] clause 8.

8.3.3 MEC Services with Multiple Apps

The SUT_MEC_SERVICES_MULTI_APP configuration is similar to the configuration SUT_MEC_SERVICES_SINGLE_APP, with a difference on the integration between both elements. In this configuration, two (2) MEC applications run together alongside the MEC Platform. The configuration focuses on the capabilities around MEC Services such as the capability of applications and the platform to provide, discover or consume MEC services.

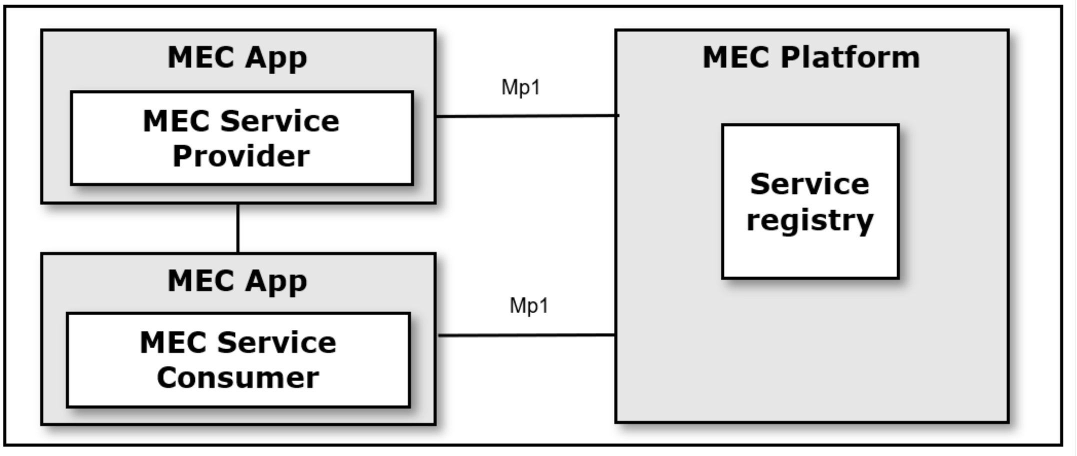


Figure 8.3.3-1: Test configuration SUT_MEC_SERVICE_MULTI_APP

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

Table 8.3.3-1. MEC Services (Multi App) Test Groups

Group	Test IDs	# TDs
MEC Application Lifecycle	TD_MEC_APP_ONBOARD TD_MEC_APP_START TD_MEC_APP_STOP TD_MEC_APP_STATUS TD_MEC_APP_CHANGE	5
MEC Services	TD_MEC_SVC_QUERY TD_MEC_SVC_REGISTER TD_MEC_SVC_UPDATE TD_MEC_SVC_DEREGISTER TD_MEC_SVC_CONSUME TD_MEC_SVC_QUERYTIME TD_MEC_SVC_TRANSPORTS	7
MEC Traffic	TD_MEC_NTW_ACTIVATE TD_MEC_NTW_UPDATE TD_MEC_NTW_DEACTIVATE TD_MEC_NTW_DNS_ACTIVATE TD_MEC_NTW_DNS_DEACTIVATE	5

MEC Location API	TD_MEC_LOC_UE_LKP_1 TD_MEC_LOC_UE_LKP_2 TD_MEC_LOC_UE_INF_LKP_1 TD_MEC_LOC_UE_INF_LKP_2 TD_MEC_LOC_UE_SUB_1 TD_MEC_LOC_UE_SUB_2 TD_MEC_LOC_INF_SUB_1 TD_MEC_LOC_INF_SUB_2 TD_MEC_LOC_RNL TD_MEC_LOC_TRACK TD_MEC_LOC_DIST_1 TD_MEC_LOC_DIST_2 TD_MEC_LOC_DIST_SUB_1 TD_MEC_LOC_DIST_SUB_2 TD_MEC_LOC_DIST_SUB_3 TD_MEC_LOC_AREA_SUB_1 TD_MEC_LOC_AREA_SUB_2 TD_MEC_LOC_AREA_SUB_3	18

The complete list of Test Cases can be found in [MEC-IOP-TP] clause 8.

8.3.4 MEC in NFV Platforms

The SUT_MEC_NFVI configuration, the MEC platform and the MEC application(s) are hosted and executed on a third party NFV Infrastructure. The focus is on interoperability of virtualization technologies and VIM APIs in a multivendor scenario.

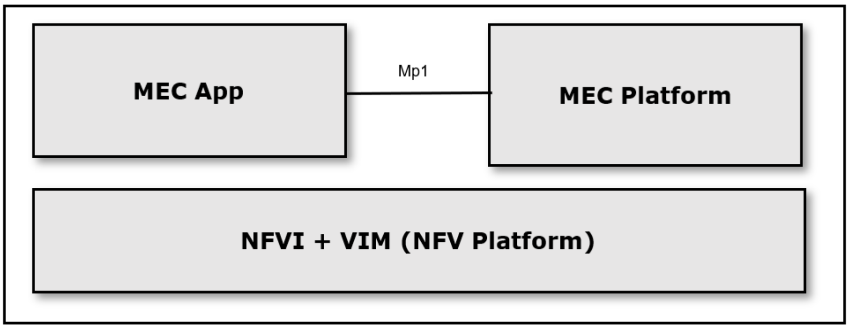


Figure 8.3.4-1: Test configuration SUT_MEC_NFVI

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

Table 8.3.4-1. MEC in NFV Test Groups

Group	Test IDs	# TDs
MEC Application Lifecycle	TD_MEC_APP_ONBOARD TD_MEC_APP_START TD_MEC_APP_STOP TD_MEC_APP_STATUS TD_MEC_APP_CHANGE	5
MEC Traffic	TD_MEC_NTW_ACTIVATE TD_MEC_NTW_UPDATE TD_MEC_NTW_DEACTIVATE TD_MEC_NTW_DNS_ACTIVATE TD_MEC_NTW_DNS_DEACTIVATE	5

The complete list of Test Cases can be found in [MEC-IOP-TP] clause 8.

8.3.5 MEC in NFV Platforms orchestrated by MANO

The SUT_MEC_MANO focuses on the MEC-in-NFV scenario. In this scenario the MEC application(s) and the MEC platform are packaged as VNFs and are managed by a third-party MANO platform in an NFV infrastructure. The availability of other components of the MEC system (such as MEAO, MEPM and specific VNFM) is out of scope.

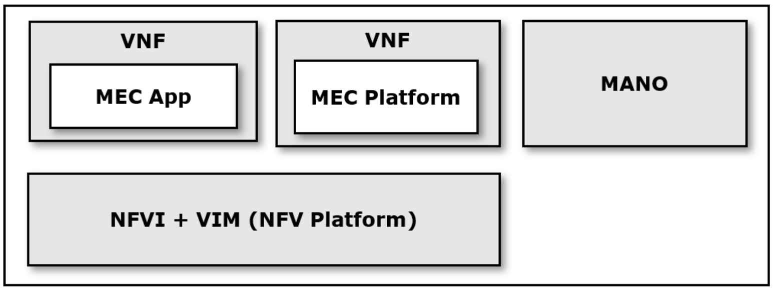


Figure 8.3.5-1: Test configuration SUT_MEC_MANO

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

Table 8.3.5-1. MEC MANO Test Groups

Group	Test IDs	# TDs
MEC Application Lifecycle	TD_MEC_APP_START TD_MEC_APP_STOP TD_MEC_APP_STATUS TD_MEC_APP_CHANGE	5
MEC Services	TD_MEC_SVC_QUERY TD_MEC_SVC_REGISTER TD_MEC_SVC_UPDATE TD_MEC_SVC_DEREGISTER	4
MEC Traffic	TD_MEC_NTW_ACTIVATE TD_MEC_NTW_UPDATE TD_MEC_NTW_DEACTIVATE TD_MEC_NTW_DNS_ACTIVATE TD_MEC_NTW_DNS_DEACTIVATE	5

The complete list of Test Cases can be found in [MEC-IOP-TP] clause 8.

8.4 MEC API Conformance

The test plan for the MEC API Conformance test track was based upon the latest stable drafts of ETSI GS [MEC-DEC032].

The work item provides a database of test purposes for MEC APIs and their implementation in TTCN-3 and Robot Framework. The Plugtests participants were able to run the test suites against their FUTs over the HIVE TAP, that implements the robot framework version of the test suites. Furthermore, they could run individual tests using both Robot Framework and TTCN-3 outside HIVE TAP by themselves, or through the Plugtests team.

Based on the capabilities and selections of the participating FUTs, the API Tests prepared and executed during the NFV&MEC IOP Plugtests 2021 targeted the MEC specifications [MEC010-2], [MEC011], [MEC012], [MEC013], [MEC014], [MEC015], [MEC016], [MEC021], [MEC029] and [MEC030]. The test suites were publicly available at [MEC-ROBOT-TS] and [MEC-TTCN3-TS]. The test suite structure followed [MEC-DEC032], Clause 5.

9 Results

9.1 NFV Interoperability

9.1.1 Overall Results

During the Plugtests, a total of 11 NFV interoperability Test Sessions were run with different combinations of the Functions Under Test (FUTs) in scope: VNFs, CNFs, MANOs, and VIM&NFVIs.

The following sections provide an overview of the reported results: overall, and on a per test group& test case basis. To facilitate the analysis, results are presented as follows:

Table 9.1.1-1: Results Interpretation

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

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:

Table 9.1.1-2: NFV IOP Overall Results

Overall Results	Number of Test Sessions	Interoperability (TCs Run)		TCs Not Run	TCs Totals	
		OK	NO	NA	Run	Total
	11	54	3	24	57	81

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 148 test cases, organised in different groups as described in clause 8.1. Through the 11 Test Sessions run, a total of 81 Test Results were reported. This figure includes both the executed and non-executed test cases. Overall, a total of 57 individual test cases were executed and results (OK or NO as per table above) reported for them. The table below provides a summary of NFV Interoperability results for each test group.

Table 9.1.1-3. NFV IOP Overall results per Group

	Interoperability		Not Run	Totals		%			
	OK	NO	N/A	Run	Results	% Run	% OK	% NO	% N/A
NS	34	0	20	34	54	62,96%	100,00%	0,00%	37,04%
NS-CNF	16	0	2	16	18	88,89%	100,00%	0,00%	11,11%
S-VNFM	4	3	2	7	9	77,78%	57,14%	42,86%	22,22%
TOTAL	54	3	24	57	81	70,37%	94,74%	5,26%	29,63%

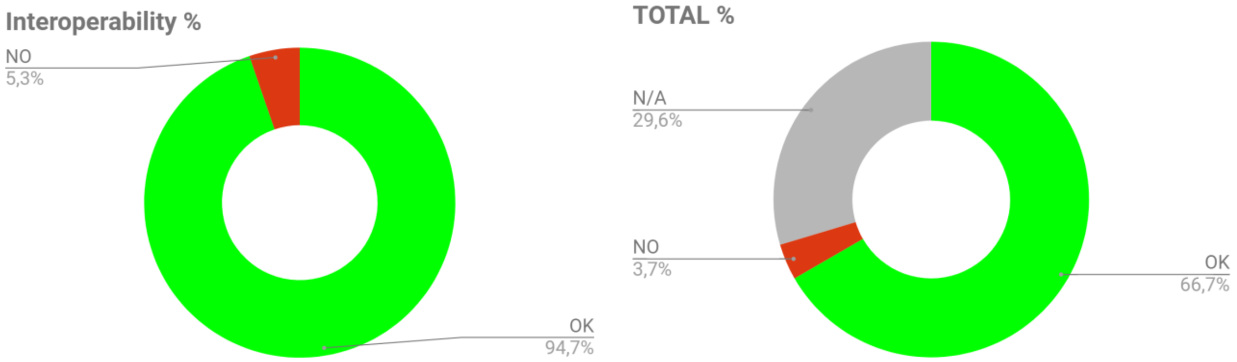


Figure 9.1.1-1. NFV IOP Overall results (%)

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 NS

The table and figure below provide an overview of the results for the NS group. Overall, 5 NS test sessions were run.

Table 9.1.2.1-1. Results for NS Sub-Group

	Interoperability		Not Run	Totals		%			
	OK	NO	NA	Run	Results	% Run	% OK	% NO	% NA
NS_ONBOARD	5	0	0	5	5	100,00%	100,00%	0,00%	0,00%
NS_INSTANTIATE	5	0	0	5	5	100,00%	100,00%	0,00%	0,00%
NS_SCALE_MANUAL	6	0	2	6	8	75,00%	100,00%	0,00%	25,00%
NS_SCALE_VIM_METRIC	2	0	6	2	8	25,00%	100,00%	0,00%	75,00%
NS_SCALE_VNF_MANUAL	4	0	4	4	8	50,00%	100,00%	0,00%	50,00%
NS_SCALE_VNF_VIM_METRIC	2	0	6	2	8	25,00%	100,00%	0,00%	75,00%
NS_SCALE_VNF_TO_LEVEL_MANUAL	2	0	2	2	4	50,00%	100,00%	0,00%	50,00%
NS_TERMINATE	4	0	0	4	4	100,00%	100,00%	0,00%	0,00%

NS_DELETE	4	0	0	4	4	100,00%	100,00%	0,00%	0,00%
TOTAL	34	0	20	34	54	62,96%	100,00%	0,00%	37,04%

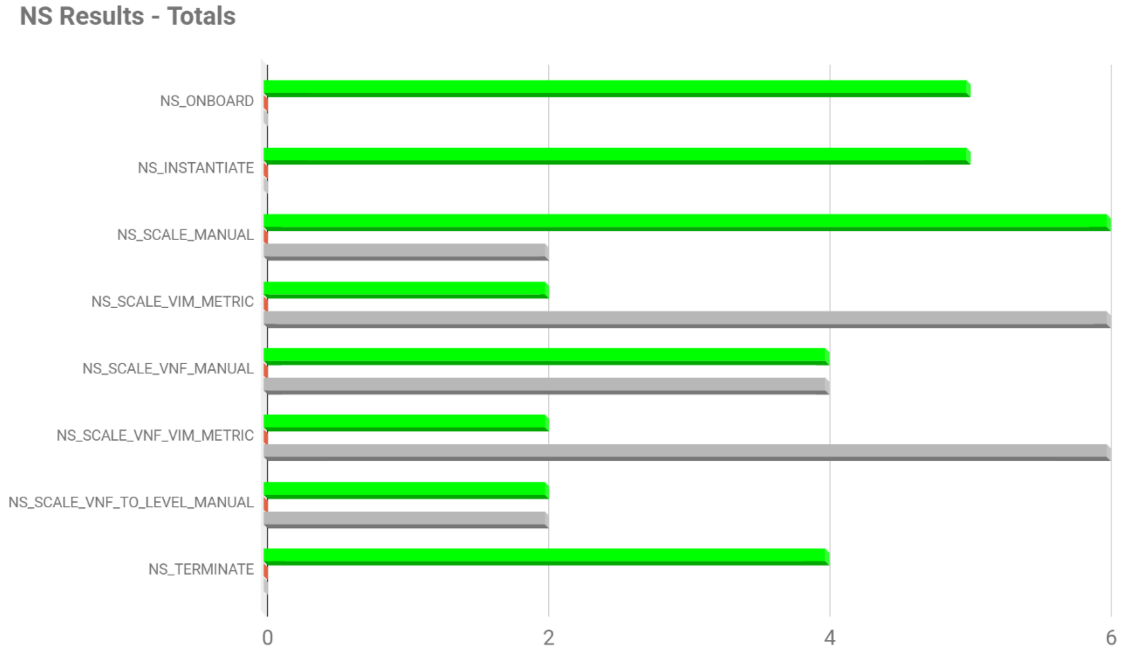


Figure 9.1.2.1-1. Results per NS Sub-Group – Totals

9.1.2.2 NS CNF

The table and figure below provide an overview of the results for the NS-CNF group. Overall, 3 NS-CNF test sessions were run.

Table 9.1.2.2-1. Results per NS-CNF Sub-Group

	Interoperability		Not Run	Totals		%			
	OK	NO	NA	Run	Results	% Run	% OK	% NO	% NA
NS-CNF_ONBOARD	3	0	0	3	3	100,00%	100,00%	0,00%	0,00%
NS-CNF_INSTANTIATE	3	0	0	3	3	100,00%	100,00%	0,00%	0,00%
NS-CNF_SCALE_MANUAL	4	0	2	4	6	66,67%	100,00%	0,00%	33,33%
NS-CNF_UPDATE_VNF_OPERATE	2	0	0	2	2	100,00%	100,00%	0,00%	0,00%
NS-CNF_TERMINATE	2	0	0	2	2	100,00%	100,00%	0,00%	0,00%
NS-CNF_DELETE	2	0	0	2	2	100,00%	100,00%	0,00%	0,00%
TOTAL	16	0	2	16	18	88,89%	100,00%	0,00%	11,11%

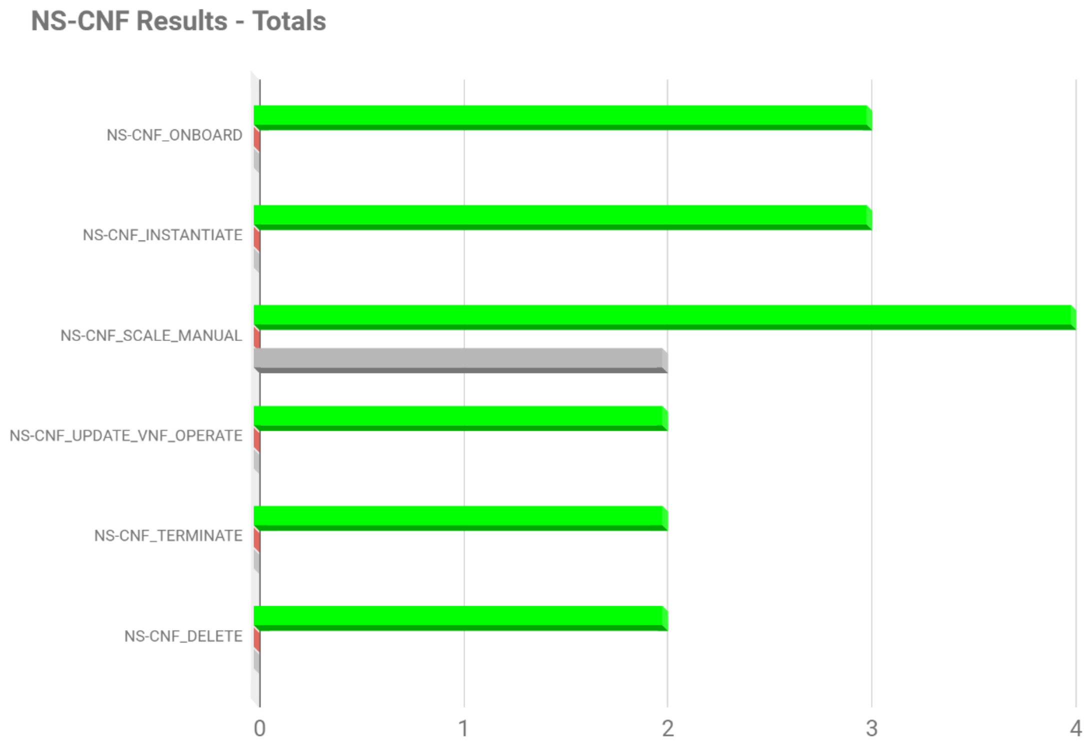


Figure 9.1.2.2-1. Results per NS-CNF Sub-Group – Totals

9.1.2.3 Specific VNFM

The table and figure below provide an overview of the results for the S-VNFM group. Overall, 3 Test Sessions involving different combinations of FUTs were run, and results reported as follows:

Table 9.1.2.3-1. Results per Specific VNFM Sub-Group

	Interoperability		Not Run	Totals		%			
	OK	NO	NA	Run	Results	% Run	% OK	% NO	% NA
S-VNFM_NS_ONBOARD	2	1	0	3	3	100,00%	66,67%	33,33%	0,00%
S-VNFM_NS_INSTANTIATE	0	2	1	2	3	66,67%	0,00%	100,00%	33,33%
S-VNFM_NS_DELETE	2	0	1	2	3	66,67%	100,00%	0,00%	33,33%
TOTAL	4	3	2	7	9	77,78%	57,14%	42,86%	22,22%

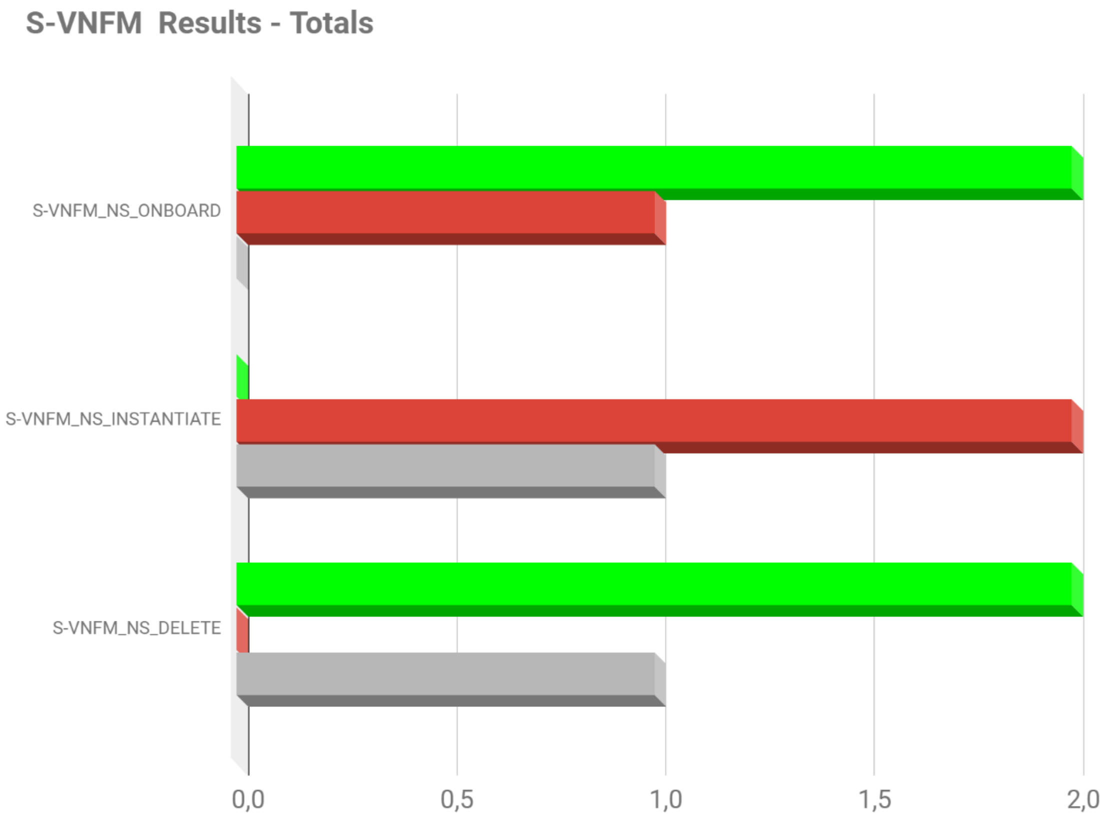


Figure 9.1.2.3-1. Results per S-VNFM Sub-Group - Totals

9.1.3 Results per Test Case

The Table below provides an overview of the results for each NFV Test Case run. The information reported in the table below is the consolidation of test results from the different tested configurations, i.e., NS, NS CNF and Specific VNFM test groups.

Table 9.1.3-1: NFV IOP results per Test Case

	Interoperability		Run	Totals	
	OK	NO	NA	Run	Results
TD_NFV_VNFPM_ONBOARD_001	10	1	0	11	11
TD_NFV_VNFPM_DELETE_001	8	0	1	8	9
TD_NFV_NSLCM_INSTANTIATE_001	8	2	1	10	11
TD_NFV_NSLCM_SCALE_OUT_001	5	0	2	5	7
TD_NFV_NSLCM_SCALE_OUT_003	1	0	4	1	5
TD_NFV_NSLCM_SCALE_IN_001	5	0	2	5	7
TD_NFV_NSLCM_SCALE_IN_003	1	0	4	1	5
TD_NFV_NSLCM_SCALE_OUT_VNF_001	2	0	6	2	8
TD_NFV_NSLCM_SCALE_OUT_VNF_003	1	0	4	1	5
TD_NFV_NSLCM_SCALE_IN_VNF_001	2	0	6	2	8
TD_NFV_NSLCM_SCALE_IN_VNF_003	1	0	4	1	5
TD_NFV_NSLCM_SCALE_TO_LEVEL_VNF_001	2	0	6	2	8
TD_NFV_NSLCM_UPDATE_START_001	1	0	4	1	5
TD_NFV_NSLCM_UPDATE_STOP_001	1	0	4	1	5
TD_NFV_NSLCM_TERMINATE_001	6	0	3	6	9

9.2 NFV API Conformance Results

During the NFV&MEC Plugtests 2021 event, several NFV API Conformance Test Sessions were run, as described in 8.2, in an automated and on-demand fashion. Out of all executed test sessions, all executed tests have been reported and for each of the tests the best result has been used in the final calculation of the outcomes reported below. The API conformance tests were executed for two different Functions Under Test (FUTs): VNFMs and NFVOs. A total of 3 FUTs (2 VNFMS and 1 NFVO) participated to the NFV API Conformance sessions. The table below provides the overall results (aggregated data) for all the NFV API Conformance tests run during the NFV&MEC Plugtests 2021, from all participating organisations.

Table 9.2-1: NFV API Conformance overall results

Overall Results	API Conformance (TCs Run)		TCs Totals
	OK	NO	Total
	210	75	285

For each remote Test Session, depending on the involved FUT and the features to be tested, the involved participants were able to select different number of test cases.

Overall, the test plan included more than 1500 NFV API Conformance test cases, organised in different groups as described in clause 8.2. The test plan was based on ETSI GS [NFV-TST010], in several versions including 2.4.1, 2.6.1, 2.7.1, 2.8.1 and 3.3.1. Participants were free to select the version of the test suite according to their implementations.

With respect to the previous NFV Plugtests event, a larger number of tests was made available. Through the Test Sessions run, a total of 285 Test Results were executed and reported.

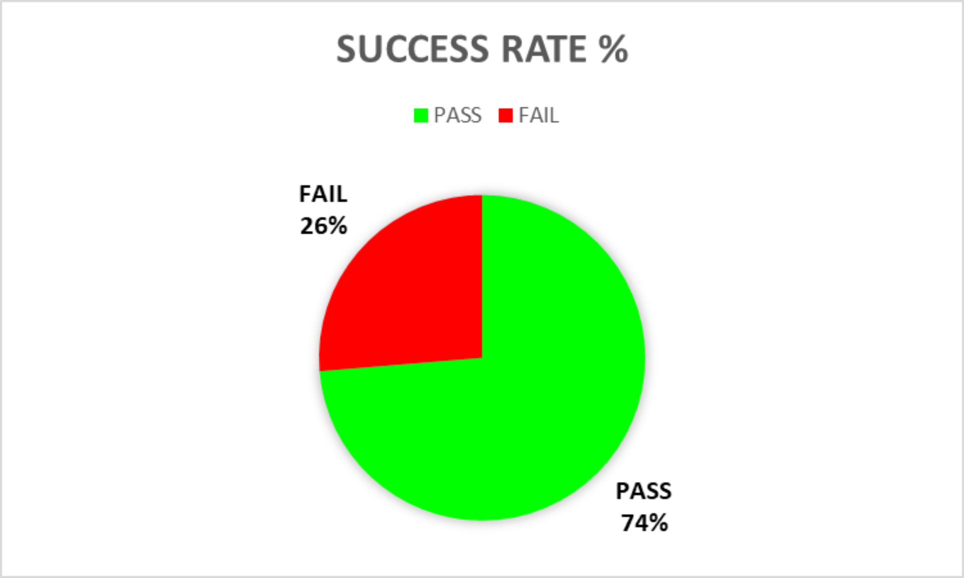


Figure 9.2-1. NFVAPI Conformance Overall results (%)

The next clauses present more detailed results per SOL Specification (and for each version), 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 NFV Specification

The tables and figures below provide an overview of the results for the API conformance per SOL specifications, i.e. [NFV-SOL002], [NFV-SOL003] and [NFV-SOL005]. Overall, the [NFV-SOL003] APIs were those with the higher number of Test Cases run.

Table 9.2.1-1: Test Results summary per Specification

	API Conformance		Totals	%	
	OK	NO	Results	% OK	% NO
NFV-SOL002	1	0	1	100%	0%
NFV-SOL003	155	55	210	74%	26%
NFV- SOL005	54	20	74	73%	27%
TOTAL	210	75	285	74%	26%

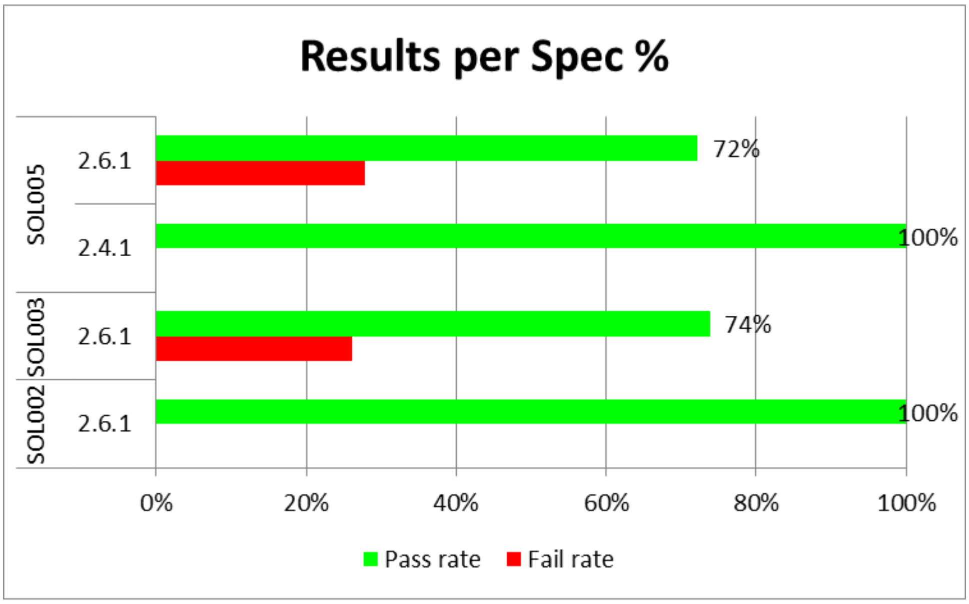


Figure 9.2.1-1. Test results per Specification - %

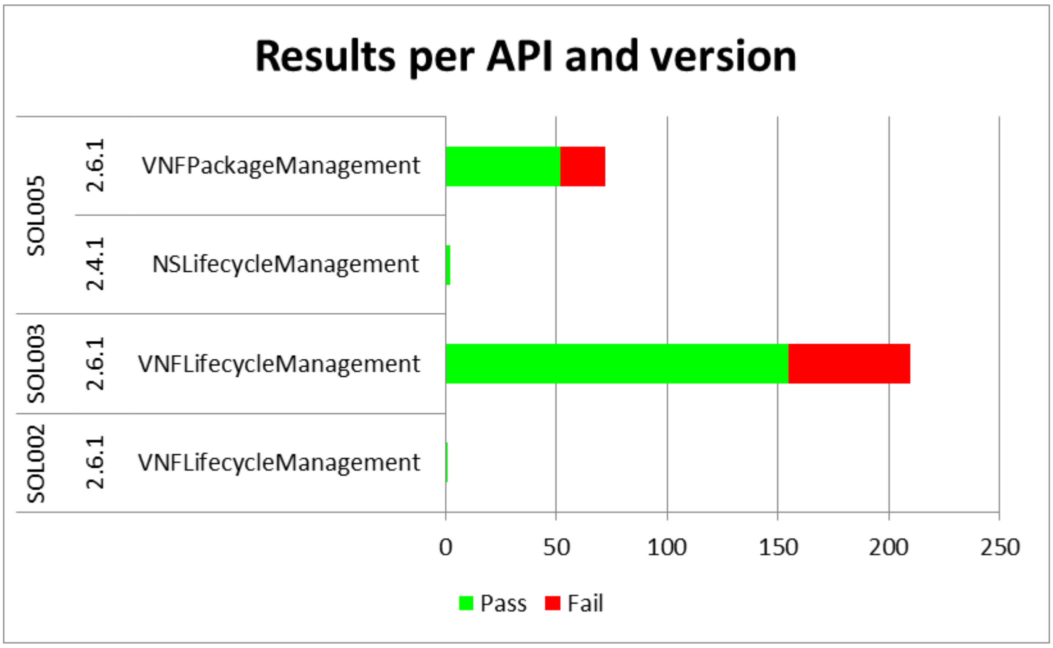


Figure 9.2.1-2. Test results per API and version

9.2.2 Results per Test Case

The full list of NFV API Conformance results per Test Case is provided in [NFV-MEC-API-TR].

9.3 MEC Interoperability

9.3.1 Overall Results

As part of the MEC Track a total of 10 Test Sessions were run, combining 4 different SUT configurations.

Table 9.3.1-1: MEC IOP Overall Results

Overall Results	Number of Test Sessions	Interoperability (TCs Run)		TCs Not Run	TCs Totals	
		OK	NO	NA	Run	Total
	10	61	0	67	61	128

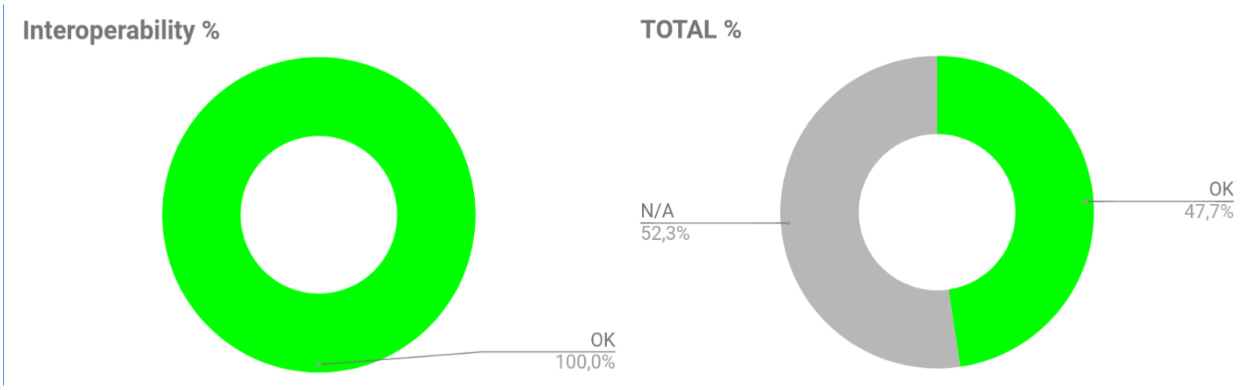


Figure 9.3.1-1 MEC IOP Overall results (%)

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

Table 9.3.1-2: MEC IOP Overall results per group

	Interoperability		Not Run	Totals		Totals (%)			
	OK	NO	N/A	Run	Results	% Run	% OK	% NO	% N/A
MEC Basic	17	0	8	17	25	68,00%	100,00%	0,00%	32,00%
MEC Services Provider	8	0	16	8	24	33,3%	100,00%	0,00%	66,7%
MEC Services Consumer	28	0	39	28	67	41,8%	100,00%	0,00%	58,2%
MEC MANO	8	0	4	8	12	66,7%	100,00%	0,00%	33,3%

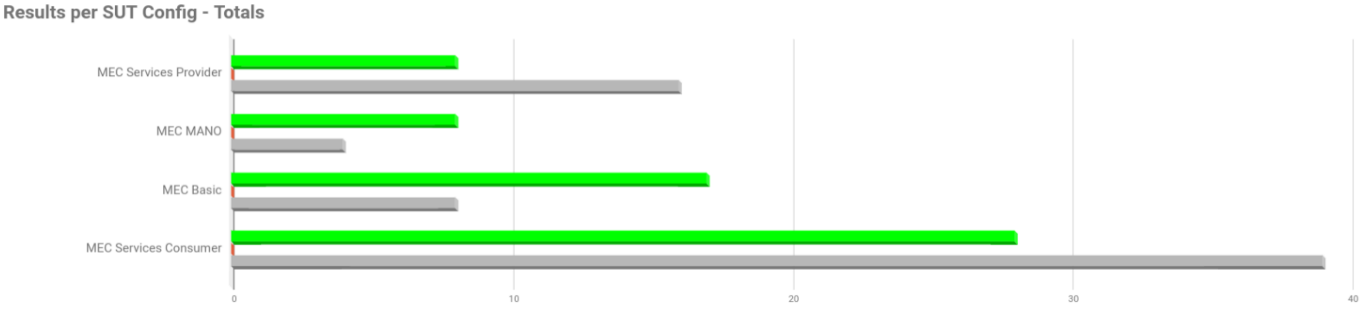


Figure 9.3.1-1. MEC IOP results per group - Totals

9.3.2 Results per Test Case

The Table below provides an overview of the results for each MEC Test Case run.

Table 9.3.2-1: MEC IOP results per Test Case

	Interoperability		Not Run	Totals	
	OK	**NO**	**NA**	**Run**	**Results**
TD_MEC_NTW_ACTIVATE	0	4	2	4	6
TD_MEC_APP_ONBOARD	5	0	1	5	6
TD_MEC_APP_START	5	0	1	5	6
TD_MEC_APP_STOP	5	0	1	5	6
TD_MEC_APP_STATUS	5	0	1	5	6
TD_MEC_APP_CHANGE	1	0	5	1	6
TD_MEC_SVC_QUERY	9	0	0	9	9
TD_MEC_SVC_REGISTER	9	0	0	9	9
TD_MEC_SVC_UPDATE	0	0	9	0	9
TD_MEC_SVC_DEREGISTER	9	0	0	9	9
TD_MEC_SVC_CONSUME	2	0	1	2	3
TD_MEC_SVC_TRANSPORTS	2	0	7	2	9
TD_MEC_SVC_QUERYTIME	3	0	6	3	9
TD_MEC_NTW_ACTIVATE	1	0	7	1	8
TD_MEC_NTW_UPDATE	1	0	7	1	8
TD_MEC_NTW_DEACTIVATE	1	0	7	1	8
TD_MEC_NTW_DNS_ACTIVATE	1	0	7	1	8
TD_MEC_NTW_DNS_DEACTIVATE	1	0	7	1	8
TD_MEC_LOC_UE_LKP_1	0	0	0	0	0
TD_MEC_LOC_UE_LKP_2	0	0	0	0	0
TD_MEC_LOC_UE_INF_LKP_1	0	0	0	0	0
TD_MEC_LOC_UE_INF_LKP_2	0	0	0	0	0
TD_MEC_LOC_UE_SUB_1	0	0	0	0	0
TD_MEC_LOC_UE_SUB_2	0	0	0	0	0
TD_MEC_LOC_INF_SUB_1	1	0	0	1	1
TD_MEC_LOC_INF_SUB_2	0	0	0	0	0
TD_MEC_LOC_RNL	0	0	0	0	0
TD_MEC_LOC_TRACK	0	0	0	0	0
TD_MEC_LOC_DIST_1	0	0	0	0	0
TD_MEC_LOC_DIST_2	0	0	0	0	0
TD_MEC_LOC_DIST_SUB_1	0	0	0	0	0
TD_MEC_LOC_DIST_SUB_2	0	0	0	0	0
TD_MEC_LOC_DIST_SUB_3	0	0	0	0	0
TD_MEC_LOC_AREA_SUB_1	0	0	0	0	0

TD_MEC_LOC_AREA_SUB_2	0	0	0	0	0
TD_MEC_LOC_AREA_SUB_3	0	0	0	0	0

9.4 MEC API Conformance Results

During the NFV&MEC Plugtests 2021 event, several MEC API Conformance Test Sessions were run, as described 8.4, in an automated and on-demand fashion. Out of all executed test sessions, all executed tests have been reported and for each of the tests the best result has been used in the final calculation of the outcomes reported below. The API conformance tests were executed for Functions Under Test (FUTs) of type “MEC Platform”. A total of 2 MEC Platforms participated to the MEC API Conformance sessions.

The table below provides the overall results (aggregated data) for all the MEC API Conformance tests run during the NFV&MEC Plugtests 2021, from all participating organisations.

Table 9.4-1: NFV API Conformance overall results

Overall Results	API Conformance (TCs Run)		TCs Totals
	OK	NO	Total
	41	14	55

For each remote Test Session, depending on the involved FUT and the features to be tested, the involved participants were able to select different number of test cases.

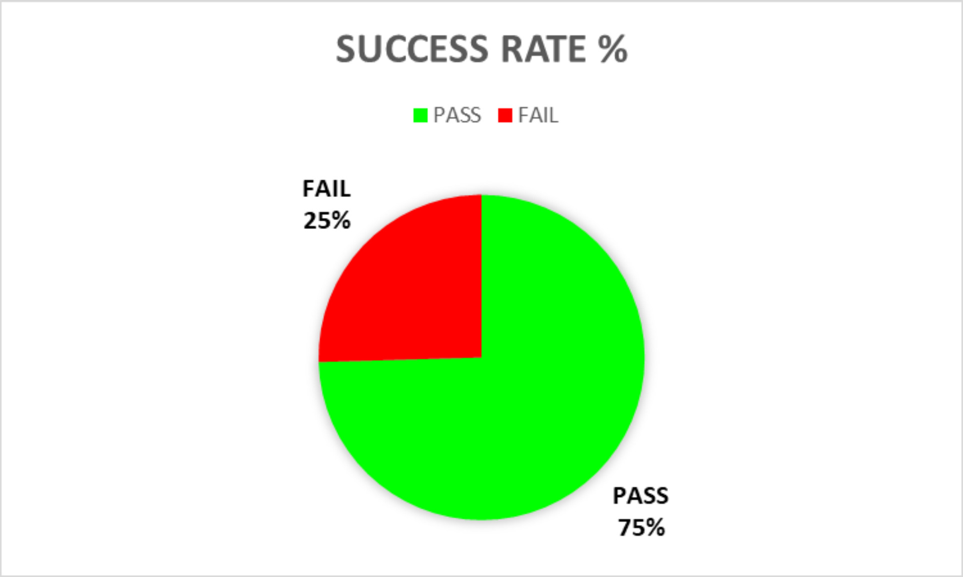


Figure 9.4-1. MEC API Conformance Overall results (%)

The next clauses present more detailed results per MEC Specification and per test group and will allow to identify the areas and APIs with higher execution and conformance rates.

9.4.1 Results per MEC Specification

The tables and figures below provide an overview of the results for the API conformance per MEC specifications, i.e. [MEC011], [MEC012] and [MEC013]. Overall, the [MEC011] APIs have been those with the higher number of Test Cases run and had the highest success rate.

Table 9.4.1-1: Test Results summary per-MEC Specification

	API Conformance		Totals	%	
	OK	NO	Results	% OK	% NO
MEC010-2	0	4	4	0%	100%
MEC011	30	10	40	75%	25%
MEC013	11	0	11	100%	0%
TOTAL	41	14	55	75%	25%

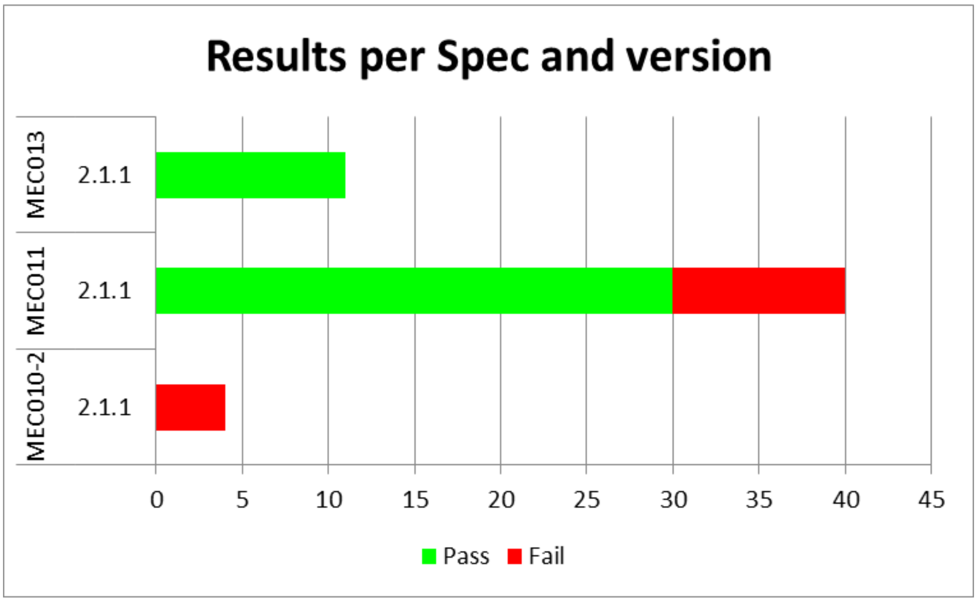


Figure 9.4.1-1. Test results per Specification and version

9.4.2 Results per Test Case

The full list of MEC API Conformance results per Test Case is provided in [NFV-MEC-API-TR].

9.4.3 MEC Sandbox validation

The ETSI MEC Sandbox provides the user with a choice of scenarios combining different network technologies (4G, 5G, Wi-Fi) and terminal types. Combining these assets in a geolocated environment, a user can gain hands-on experience on the behaviour and capabilities of the Location [MEC013], Radio Network Information [MEC012], WLAN Information [MEC028], Edge Platform Application Enablement [MEC011] and Application Mobility [MEC021] service APIs.

NOTE: Due to the BETA version status, the following limitations applied:

1. Supported version of [MEC011] was 2.1.1, not 2.2.1,
2. Not all functionalities from [MEC011] were available for testing (e.g., DNS, CONFTASK, MSL not supported yet)

The TTCN-3 implementation of the [MEC-DEC032] test suite [MEC-TTCN3-TS] was used to validate the MEC standards. The table below summarizes the result of this validation:

Table 9.4.3-1: TTCN-3 tests against MEC Sandbox

	API Conformance			Totals	%		
	OK	NO	N/A	Results	% OK	% NO	%N/A
MEC011	19	4	22	55	34%	7%	59%
MEC012	12	2	17	31	39%	6%	55%
MEC013	24	6	0	30	80%	20%	0%
MEC028	13	6	1	20	65%	30%	5%
TOTAL	68	18	50	136	50%	13%	37%

Using the same principles, the Robot Test Suites [MEC-ROBOT-TS] was used to validate the [MEC011], [MEC013] and [MEC028] specifications implemented in the MEC Sandbox. The following table summarizes the results.

Table 9.4.3-1: Robot tests against MEC Sandbox

	API Conformance			Totals	%		
	OK	Results	N/A	Results	% OK	% NO	% N/A
MEC011	31	7	17	55	56%	13%	31%
MEC013	33	0	0	33	100%	0%	0%
MEC028	17	2	9	28	61%	7%	32%
TOTAL	81	9	26	116	70%	8%	22%

10 Plugtests Outcome

As for previous events, the NFV&MEC IOP Plugtests 2021 allowed to identify a number of potential interoperability issues and inconsistencies in NFV and MEC specifications. It also triggered interesting discussions among participants, organisers, observers and representatives 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 and 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 NFV Specifications

10.1.1 NFV-SOL013 - IOP Issue with authentication methods

During the NFV&MEC API Plugtests 2021 a usability issue with authentication and authorization in the [NFV-ROBOT-TS] was identified, reported, and fixed. In particular, the [NFV-ROBOT-TS] included the name of such HTTP header “Authorization” as hard-coded into the low-level code of the tests, bringing a lack of flexibility in the usage of custom HTTP header name to transmit the authentication tokens for API requests. However, [NFV-SOL013] specifies the header name for authorization to be required, this being the name “Authorization” as the only option. Therefore, the usability issue was targeting an improvement in flexibility and usability of the Test Suites, and not the correctness of authentication mechanisms to be used. After a discussion within NFV-TST Working Group, it was decided to update the behaviour of the [NFV-ROBOT-TS] in order to allow users to set the HTTP header name as a configurable parameter.

Similarly, during this NFV&MEC IOP Plugtests, a related interoperability issue occurred, concerning the incompatibility between NFV FUTs (NFVO and VNFM in the specific case) in interoperating through [NFV-SOL003] APIs due to the use of different authorization and authentication methods. In practice, even if both NFV FUTs were successfully executing the stand-alone NFV API conformance tests, with one of them using a custom Authentication HTTP header (i.e., not compliant with [NFV-SOL013]), it was not possible to run any IOP testing.

This IOP issue confirmed that the usage of an Authentication HTTP Header name different from the one specified in [NFV-SOL013] (i.e., “Authorization”) is to be considered not conformant with the NFV API specifications (even if its customization is granted as an exception for NFV API conformance), and as such it brings interoperability issues when integrating with [NFV-SOL013] conformant solutions.

10.1.2 NFV-SOL001, NFV-SOL006 Descriptors

In previous events, some concern was raised by participants, mostly VNF vendors, about the lack of uniformity and the number of variations and customizations in VNF Descriptors across MANO solutions. Concrete concerns were expressed on the ability of NFV Descriptors to provide interoperability for complex operations, where often, MANO specific Descriptor extensions seemed to be required to achieve successful results. While this year a significant improvement was seen in the adoption of standardized descriptors, some concerns still remain about the effort required in some cases to adapt descriptors for their reuse across different MANO solutions.

10.1.3 NFV-TST007 Interoperability Test Plan

The NFV Interoperability testing was run following the NS related Test Descriptions in [NFV-TST007] (mainly clauses 7.3, 7.5, 7.6 and 7.7). The complete list of Test Descriptions in scope of this Plugtests is reported in clause 8.1. Moreover the [NFV-TST007] IFS templates have been significantly revised with the aim of simplifying them and be used to properly configure the Test Reporting Tool. The related NFV IFS templates offered to the Plugtests participants in order to capture the capabilities of their FUTs in this Plugtests are provided in Annex A.

10.1.4 NFV-TST010 API Conformance Spec and Test Suite

The NFV API Conformance Test Sessions run during the NFV&MEC IOP Plugtests 2021 were based on [NFV-TST010] v2.4.1 and v2.6.1

The main improvement with respect to previous Plugtests was the additional support of v2.7.1 and preliminary versions v2.8.1, v3.3.1 of [NFV-TST010].

Overall, the NFV&MEC Plugtests 2021 allowed to identify and file 7 new 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.

Table 10.1.4-1. Issues Reported fort NFV TST010 Robot Test Suites

Issue	Description	SOL002	SOL003	SOL005
#189	The 'descriptor_id' is fetched as None, causing validation failure	X	X	X
#190	Wrong Problem Details JSON Object set in Keywords Check HTTP Response Body Json Schema Is ProblemDetails SOL005 VNFPackageManagementAPI v2.6.1			X
#191	Test NSFaultManagement-API.Alarms Incorrect Schema validation			X
#192	Wrong Parameter name in IndividualVNFInstance.robot SOL003 VNFLifecysleManagementAPI		X	
#193	Unexpected GET method runs after http code 303 returned in Subscriptions.robot SOL003 VNFLifecycleManagementAPI		X	
#194	Keywords not running when testing Individual VNFInstances SOL003 VNFLifecycleMangementAPI V2.6.1		X	
#195	Bad use of Format String when replacing value of callbackUri attribute	X	X	X

10.2 Feedback on MEC Specifications

10.2.1 MEC012 – Duplicated values in enumerated

A request for an editorial change was reported to MEC-DECODE working group on ETSI GS [MEC012] V2.1.1 (2019-12) Table 6.6.3-1: Enumeration Trigger. The issue is a duplicated value in the enumerated data structure. The same issue was also detected in later version of this standard.

At the same time, a change request was on the ETSI doc2oas tool, which is used to automatically generate OpenAPI definitions from ETSI Specifications (Issue#4). The doc2oas tool generates a baseline OAS3 data model (yaml) from a specification document (docx). It parses the Data object tables from the specification document and creates the corresponding data model objects to the OAS3 yaml file.

10.2.2 MEC042 – MEC Interoperability Test Plan

A dedicated MEC Interoperability Test Plan [MEC-IOP-TP] was developed by the Plugtests team to support MEC IOP Test Sessions. The Test Plan will be contributed to ISG MEC DEC WG.

The following feedback was captured on the MEC Interoperability Test Plan:

- Availability in the test plan of examples and conventions for MEC App descriptors may ease integration and foster participation of MEC App providers.

10.2.3 MEC-DEC032 - API Conformance Spec and Test Suites

The MEC API Conformance Test Plan run during the NFV&MEC IOP Plugtests 2021 were based on MEC [DEC032]. The test suite was implemented in TTCN-3 [MEC-TTCN3-TS] and Robot Framework [MEC-ROBOT-TS].

Overall, the NFV&MEC IOP Plugtests 2021 allowed to identify and fix 7 new issues on the MEC API Conformance Test Suites, where 3 of them were related to the TTCN-3 implementation and the other 4 to the Robot Framework implementation. The tables below summarise all the issues and indicates the impacted MEC Test Specifications and the number under which the issue was filed in the [MEC-TTCN-3-ISSUE] and [MEC-ROBOT-ISSUE] set up for that purpose in the ETSI Forge.

Table 10.2.6-1: Issues reported for MEC DEC032 TTCN-3 Test Suites

Issue	Description	Base Spec
#20	Reorganize ATS (use one unique file)	[MEC012]
#21	Add not implemented TPs	[MEC011]
#23	Rename MEC-011 TTCN-3 modules	[MEC013]

Table 10.2.6-2: Issues reported for MEC DEC032 Robot Test Suites

Issue	Description	Base Spec
#42	Failure running TC_MEC_MEC013_SRV_UEDISTSUB_001_OK	[MEC013]
#43	MEC028 - Measurements endpoint using wrong rest methods	[MEC028]
#44	MEC028 - Measurements endpoint - missing brackets	[MEC028]
#45	TP_MEC_MEC011_SRV_APPSUB_002_BR: Testcase is expecting wrong response code	[MEC011]

10.3 Other Outcome

10.3.1 CNCF CNF Test Suite comparison with NFV-EVE011

Building up on the NFV&MEC API Plugtests run in February, see [NFVMECAPI2021-R], the work on the CNF Testing [CNCF-CNF-TS] comparison with the classification of Cloud Native Network Functions defined in [NFV-EVE011] continued. The outcome of this activity has been made public by the NFV ISG at [CNCF-EVE011] and was discussed with participants during the wrap up of the NFV&MEC IOP Plugtests 2021.

10.3.2 ETSI MEC Sandbox

The following issues, found during the API Conformance tests against the MEC Sandbox, were reported to the MEC Sandbox team:

Table 10.4.2-1: Issues reported for MEC Sandbox

Issue	Description	Base Spec
NA-1606	Creating a subscription requires Location in response header	[MEC011]
NA-1607	GET /service/<serviceId> should return ServiceInfo object	[MEC011]
NA-1608	subscriptionType missing from SubscriptionLinkList	[MEC011]
MECSTF-354	Misleading error message	[MEC013]

10.3.3 ETSI MEC Test System

The following issues were found and fixed on the ETSI MEC TTCN-3 Test System during the NFV&MEC IOP Plugtests 2021 event:

Issue	Description
Issue#18	JSON encoding/decoding exception on UserTrackingSubscription
Issue#19	JSON codec encoding issue for PeriodicNotificationSubscription, ZonalTrafficSubscription and DistanceNotificationSubscription
Issue#22	JSON codec encoding issues MEC-011

Annex A – NFV Interoperability Feature Statements

A.1 NFV NFVO IFS

Table A-1: NFV NFVO IFS

IFS MANO	
IFS_VNF	NFVO supports NS composed by one or more VNFs
IFS_CNF	NFVO supports NS composed by one or more CNFs
IFS_MULTISITE	NFVO supports multi-site deployments
IFS_NS_SCALE	NFVO supports manual NS scale out/in by adding/removing VNF instances
IFS_NS_SCALE_IND	NFVO supports automated NS scale out/in triggered by VNF Indicators
IFS_NS_SCALE_PM	NFVO supports automated NS scale out/in triggered by performance metrics
IFS_VNF_IND	NFVO supports collecting VNF indicators
IFS_VNF_PM	NFVO supports collecting VNF/virtual resource performance metrics
IFS_VNF_FM	NFVO supports collecting VNF/virtual resource faults and alarms
IFS_VNF_SCALE	NFVO supports manual VNF scale out/in by adding/removing VNFC instances
IFS_VNF_SCALE_IND	NFVO supports automated VNF scale out/in triggered by VNF Indicators
IFS_VNF_SCALE_PM	NFVO supports automated VNF scale out/in triggered by performance metrics
IFS_NS_SCALE_TO_LEVEL	NFVO supports manual NS scale to level
IFS_NS_SCALE_TO_LEVEL_IND	NFVO supports automated NS scale to level triggered by VNF Indicators
IFS_NS_SCALE_TO_LEVEL_PM	NFVO supports automated NS scale to level triggered by performance metrics
IFS_VNF_SCALE_TO_LEVEL	NFVO supports manual VNF scale to level
IFS_VNF_SCALE_TO_LEVEL_IND	NFVO supports automated VNF scale to level triggered by VNF Indicators
IFS_VNF_SCALE_TO_LEVEL_PM	NFVO supports automated VNF scale to level triggered by performance metrics
IFS_VNF_OPERATE	NFVO supports NS update with start/stop VNF
IFS_VNF_DF	NFVO supports NS update with change VNF DF
IFS_VNF_CONFIG	NFVO supports NS Update with VNF config modif
IFS_FM_SUB	NFVO supports NS fault management subscription operations
IFS_FM_GET	NFVO exposes NS fault management alarm reports
IFS_PM_JOB	NFVO supports NS performance monitoring jobs operations
IFS_PM_THR	NFVO supports NS performance monitoring thresholds operations

A.2 NFV VNFM IFS

Table A-2: NFV VNFM IFS

IFS VNFM	
IFS_VNF	VNFM supports VNFs
IFS_CNF	VNFM supports CNFs
IFS_VNF_SCALE	VNFM supports VNF scale out/in by adding/removing VNFC instances
IFS_VNF_IND	VNFM supports collecting VNF indicators
IFS_VNF_PM	VNFM supports collecting VNF/virtual resource performance metrics
IFS_VNF_FM	VNFM supports collecting VNF/virtual resource faults and alarms
IFS_VNF_SCALE_TO_LEVEL	VNFM supports VNF scale to level
IFS_VNF_SCALE_TO_LEVEL_IND	VNFM supports automated VNF scale to level triggered by VNF Indicators
IFS_VNF_SCALE_TO_LEVEL_PM	VNFM supports automated VNF scale to level triggered by performance metrics
IFS_VNF_OPERATE	VNFM supports VNF start/stop operations
IFS_VNF_DF	VNFM supports change VNF deployment flavor
IFS_VNF_CONFIG	VNFM supports VNF configuration modification
IFS_FM_SUB	VNFM supports VNF fault management subscription operations
IFS_FM_GET	VNFM exposes VNF fault management alarm reports queries
IFS_PM_JOB	VNFM supports VNF performance monitoring jobs operations
IFS_PM_THR	VNFM supports VNF performance monitoring thresholds operations

A.3 NFV VNF IFS

Table A-3: NFV VNF IFS

IFS VNF/CNF	
IFS_VNF_SCALE	VNF can scale out/in by adding/removing VNFC instances
IFS_VNF_IND	VNF exposes VNF Indicators towards VNFM
IFS_VNF_CONFIG	VNF supports VNF configuration modification
IFS_VNF_OPERATE	VNF can be started/stopped by VNFM/NFVO

A.4 NFV NFVI/VIM IFS

Table A-4: NFV NFVI-VIM IFS

IFS NFVI/VIM	
IFS_VNF	VIM supports VM-based VNFs
IFS_CNF	VIM supports Containerized Network Functions (e.g. on top of Kubernetes)
IFS_VNF_PM	VIM exposes virtualised resource performance metrics
IFS_VNF_FM	VIM exposes virtualised resource faults and alarms

Annex B – MEC Interoperability Feature Statements

B.1 Entities

Table B.1-1: Entities

Item	Which entity do you support?	Status	Support
1	MEC App	Available	Optional
2	MEC Platform	Available	Optional
3	NFV Platform (NFVI + VIM)	Available	Optional
4	MANO	Available	Optional

B.2 MEC App

Table B.2-1: MEC App Features

Item	Feature	ID	Status	Support
1	App Descriptor	IFS_MEC_APP_APPD	Available	Mandatory
2	MEC Service API consumer	IFS_MEC_APP_CONS	Available	Optional
3	MEC Service API producer	IFS_MEC_APP_PROD	Available	Optional
4	Packaged as VNF	IFS_MEC_APP_VNF	Available	Optional
5	Able to discover services through Service Enablement API over Mp1	IFS_MEC_APP_DISCOVER	Available	Optional
6	Able to request traffic rules support	IFS_MEC_APP_TRAFFIC	Available	Optional
7	Able to request DNS rules support	IFS_MEC_APP_DNS	Available	Optional
8	Support of MEC-013 Location API	IFS_MEC_APP_LOC	Available	Optional

B.3 MEC Platform

Table B.3-1: MEC Platform Features

Item	Feature	ID	Status	Support
1	Implements Service Enablement API	IFS_MEC_PLAT_SRV	Available	Optional
2	Implements Traffic Rules feature of Application Enablement API	IFS_MEC_PLAT_TRAFFIC	Available	Optional
3	Implements DNS Rules feature of Application Enablement API	IFS_MEC_PLAT_DNS	Available	Optional

4	Implement MEC-013 Location service	IFS_MEC_PLAT_LOC	Available	Optional

History

Document history		
V1.0.0	23/22/2021	Publication