



GROUP SPECIFICATION

Network Functions Virtualisation (NFV) Release 5; Testing; API Conformance Testing Specification

Disclaimer

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

Reference

RGS/NFV-TST010ed521

Keywords

API, conformance, NFV, testing

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from the
[ETSI Search & Browse Standards](#) application.

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

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

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

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

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

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

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2026.
All rights reserved.

Contents

Intellectual Property Rights	5
Foreword.....	5
Modal verbs terminology.....	5
Introduction	5
1 Scope	6
2 References	6
2.1 Normative references	6
2.2 Informative references.....	7
3 Definition of terms, symbols and abbreviations.....	7
3.1 Terms.....	7
3.2 Symbols.....	8
3.3 Abbreviations	8
4 Methodology	8
4.1 General	8
4.1.1 Introduction.....	8
4.1.2 Conformance of the NFV specifications.....	9
4.2 System Under Test (SUT)	9
4.3 Test configurations	9
4.3.1 General.....	9
4.3.2 Config_prod_VE.....	10
4.3.3 Config_prod_VNFM	10
4.3.4 Config_prod_NFVO	11
4.3.5 Config_prod_VNFM_GRANT.....	11
4.3.6 Config_prod_Notif_Endpoint.....	12
4.3.7 Config_prod_NFV-MANO	12
4.4 Void.....	13
4.5 Generic Test Description.....	13
4.5.1 General.....	13
4.5.2 Test Description format	13
4.5.3 Scope of the tests	15
4.5.3.1 General	15
4.5.3.2 General characteristics of the reference points.....	16
4.5.3.3 Basic behaviours of the API producer/consumer and verification steps	17
4.5.3.3.0 Introduction	17
4.5.3.3.1 Producer sends event-triggered notifications based on consumer subscriptions	17
4.5.3.3.2 Void.....	19
4.5.3.3.3 Producer executes the requested API operation	19
4.5.3.3.4 Consumer fetches the files/package info	20
4.5.3.4 Workflow test considerations.....	22
4.5.4 Verification	22
4.5.4.0 Introduction.....	22
4.5.4.1 Common verification aspects	22
4.5.4.2 Verification aspects for individual APIs	23
4.5.4.3 Verification aspects not considered.....	24
5 Void.....	24
6 Void.....	24
7 Void.....	24
Annex A (informative): Known Issues.....	25
Annex B (informative): Workflow Test Descriptions	26

Annex C:	Void	27
Annex D (normative):	Word format presentation of the test suite for the Os-Ma-nfvo Reference Point	28
Annex E (normative):	Word format presentation of the test suite for the Ve-Vnfm Reference Point	29
Annex F (normative):	Word format presentation of the test suite for the Or-Vnfm Reference Point	30
Annex G (normative):	Word format presentation of the test suite for NFV-MANO Management	31
Annex H (normative):	Word format presentation of the test suite for the Or-Or reference point	32
H.1	General	32
H.2	SOL011 NSD Management.....	32
H.3	SOL011 NS Lifecycle Management	32
H.4	SOL011 NS Performance Management	32
H.5	SOL011 NS Fault Management	33
Annex I (normative):	Word format presentation of the test suite for NFV-MANO Policy Management	34
Annex IA (normative):	Word format presentation of the test suite for Telco Cloud data analytics service.....	35
Annex J (informative):	Inter-connected Test Descriptions.....	36
Annex K (informative):	TD Change history.....	38
K.1	Release 5.1.1	38
K.1.1	New TDs for the new testing aspects	38
K.1.2	New TDs that were supposed to be included but missing in the previous versions of the present document	38
K.2	Release 5.2.1	39
K.2.1	New TDs for the new testing aspects	39
K.2.2	New TDs that were supposed to be included but missing in the previous versions of the present document	41
Annex L (normative):	Special Remarks on Test Descriptions of this version of the present document	42
Annex M (informative):	Change history	43
History		44

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the [ETSI IPR online database](#).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

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

Foreword

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

Modal verbs terminology

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

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

Introduction

The interoperability among the functional entities which supports the requirements and functionalities specified in ETSI NFV deliverables is one of the key and most important aspect to develop and deploy the NFV environment. In order to achieve such interoperability, ETSI GR NFV-TST 007 [i.1] specifies the methodologies and test scenario for the testing of interoperability mainly based on the ETSI NFV IFA specification series which specifies the reference point/interface and functional requirements. At the same time, the validation of the NFV specification compliance of each functional entities is also key aspects to be ensured for the interoperability, in particular protocol solution/API level. Therefore, the present document specifies the API conformance testing.

1 Scope

The present document specifies methodologies of conformance test, including Test Descriptions for NFV implementations with interfaces specified in NFV specifications of the following:

- ETSI GS NFV-SOL 002 [2] for the *Ve-Vnfm* reference point;
- ETSI GS NFV-SOL 003 [1] for the *Or-Vnfm* reference point;
- ETSI GS NFV-SOL 005 [3] for the *Os-Ma-Nfvo* reference point;
- ETSI GS NFV-SOL 009 [7] for the management of NFV-MANO;
- ETSI GS NFV-SOL 011 [8] for the *Or-Or* reference point;
- ETSI GS NFV-SOL 012 [11] for the policy management interface; and
- ETSI GS NFV-SOL 025 [12] for the protocol and data models of Telco Cloud data analytics service.

Each ETSI NFV-SOL deliverable specifies a set of interfaces built on the RESTful approach and intended for use over the HTTP protocol. The aim of the present document is to define the methodologies and the procedures with Test Descriptions to test conformance of the exchanged HTTP payloads and the implementation of required actions for one or more of the available interfaces within a reference point.

Since the targets of the testing are functionality (semantic checks) and the HTTP payloads (syntax checks), methodologies, including test suite(s) and/or any technologies from any organizations (in particular Open Source Initiatives) that can improve or help the (automated) test execution are also considered as being in the scope of the present document. See more details of the scope and means of conformance test specified in clause 4.1.2.

NOTE: The full set of functionalities supported by ETSI NFV Release 5 (V5.2.1) can be found in ETSI GS NFV-IFA 010 [i.5].

2 References

2.1 Normative references

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

Referenced documents which are not found to be publicly available in the expected location might be found in the [ETSI docbox](#).

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

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

- [1] [ETSI GS NFV-SOL 003 \(V5.2.1\)](#): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; RESTful protocols specification for the Or-Vnfm Reference Point".
- [2] [ETSI GS NFV-SOL 002 \(V5.2.1\)](#): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; RESTful protocols specification for the Ve-Vnfm Reference Point".
- [3] [ETSI GS NFV-SOL 005 \(V5.2.1\)](#): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point".
- [4] [ETSI GS NFV-TST 002 \(V1.1.1\)](#): "Network Functions Virtualisation (NFV); Testing Methodology; Report on NFV Interoperability Testing Methodology".

- [5] [ETSI GS NFV-SOL 013 \(V5.2.1\)](#): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; Specification of common aspects for RESTful NFV MANO APIs".
- [6] [ETSI GS NFV-SOL 001 \(V5.2.1\)](#): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; NFV descriptors based on TOSCA specification".
- [7] [ETSI GS NFV-SOL 009 \(V5.2.1\)](#): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; RESTful protocols specification for the management of NFV-MANO".
- [8] [ETSI GS NFV-SOL 011 \(V4.5.1\)](#): "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; RESTful protocols specification for the Or-Or Reference Point".
- [9] Void.
- [10] Void.
- [11] [ETSI GS NFV-SOL 012 \(V4.5.1\)](#): "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; RESTful protocols specification for the Policy Management Interface".
- [12] [ETSI GS NFV-SOL 025 \(V5.2.1\)](#): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; Specification of protocol and data model solutions for Telco Cloud data analytics service".

2.2 Informative references

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

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

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

- [i.1] ETSI GR NFV-TST 007 (V3.1.1): "Network Functions Virtualisation (NFV) Release 3; Testing; Guidelines on Interoperability Testing for MANO".
- [i.2] [Robot Framework](#).
- [i.3] [Robot2doc tool](#).
- [i.4] Void.
- [i.5] [ETSI GS NFV-IFA 010 \(V5.2.1\)](#): "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Functional requirements specification".
- [i.6] ETSI Drafting Rules.

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

Test Description (TD): set of information required to define/run the API conformance test and to realize the verdict for the API conformance test

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

API	Application Programming Interface
AUT	API Under Test
EM	Element Manager
FUT	Function Under Test
IUT	Implementation Under Test
LCM	Life Cycle Management
NFVI	NFV Infrastructure
NFVO	NFV Orchestrator
NS	Network Service
NSD	Network Service Descriptor
PNFD	PNF Descriptor
SUT	System Under Test
TD	Test Description
TLS	Transport Layer Security
VE	Virtual Element
VIM	Virtualised Infrastructure Manager
VNF	Virtual Network Function
VNFD	VNF Descriptor
VNFM	VNF Manager

4 Methodology

4.1 General

4.1.1 Introduction

The purpose of general conformance testing is to determine to what extent an implementation of a standard conforms to the requirements of that standard. Concepts from ETSI GS NFV-TST 002 [4] are used in the present document.

The important factors which characterize conformance testing are as follows:

- The System or Implementation Under Test (SUT or IUT) defines the boundaries (open interfaces) for testing.
- The conformance test system is a specialized tool (system) built for the purpose of testing and on which specific test scripts can be run.
- The SUT comes from a single supplier (or, at least, a single product line).
- The tests are executed by a dedicated test system that has full control of the SUT and the ability to observe all communications from the SUT.
- The tests are performed at open standardized interfaces that are not (usually) accessible to a normal user (i.e. they are specified at the protocol level).
- The tests are specified at the detailed protocol level and are not usually based on functionality as experienced by a user.
- The tests verify responses or related request operations from SUT and also verify notifications when appropriate.

4.1.2 Conformance of the NFV specifications

The conformance test including Test Descriptions (see clause 4.5 of the present document) is designed to test NFV implementations if those are following to the NFV specifications listed in clause 1 of the present document. The Test Descriptions tests the conformance of the exchanged HTTP payloads and the implementation of required actions for one or more of the available interfaces within a reference point.

The objectives of the "conformance" are the provisions described in each NFV specifications with typically use of "Verbal forms for the expression of provisions" specified in clause 3.2 of the ETSI Drafting Rules [i.6], i.e. "shall", "shall not" for Requirements, "should", "should not" for Recommendations, "may", "need not" for Permission, "can", "cannot" for Possibility and capability. Thus, all of the above types of listed provisions in the target NFV specifications are subject to conformance test with Test Descriptions of the present document, in terms of functionality (semantic checks) and the HTTP payloads (syntax checks).

NOTE: See also Annex L for special remarks on "conformance target provisions".

4.2 System Under Test (SUT)

The System Under Test (SUT) is identified by an implementation of the Function Under Test (FUT) that produces or consumes the API under test, e.g. in the case of the Or-vnfm reference point, the Function Under Test (FUT) may be either an NFVO implementation or a VNFM implementation.

The function shall be tested in isolation from other functional blocks in an NFV platform, to ensure that the outcomes of the conformance tests are not due to interoperability issues with other components.

4.3 Test configurations

4.3.1 General

In accordance with clause 1, the scope of the present document is to define a testing methodology and test suite for both the conformant protocol exchange (i.e. valid serialization and order of messages) and the initialization or execution of the functionalities for each protocol operation, including the conformant management of internal state.

In order to enable the FUT to correctly execute the operations, the FUT shall be tested while being executed in a test environment which provides all the functional elements needed for the correct outcome of the operation.

EXAMPLE: To correctly execute an instantiation, a VNFM requires evaluation in a test environment that provides a VIM and NFVI, plus the NFVO to grant the operation.

The test system shall provide the implementation of an API Consumer and a Notification Endpoint for the API Under Test (AUT). Moreover, the test configuration may include observation interfaces between the Test System and the FUT, or any other functional block which is part of the test environment. The specification of the mentioned observation interfaces is out of the scope of the present document.

Stimuli to the FUT shall be injected by the Test System via the AUT only.

Conformance checks on the status and outcome of the operations triggered by the protocol shall be verified by the Test System by means of:

- read operations issued via the AUT; or
- reception of notifications on the Notification Endpoint exposed by the test system; or
- other test interfaces to support triggers or verifications.

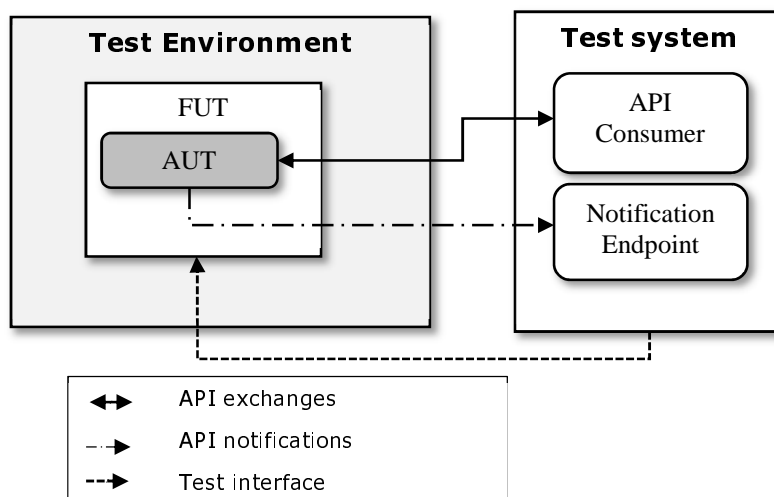


Figure 4.3.1-1: Generic SUT configuration

The test configurations specified in clause 4.3 fulfil the needs of the Test Descriptions specified in annexes D, E, F, G, H, I and J contained in the archive `gs_nfv-tst010v050201p0.zip`, which accompanies the present document, for the different FUTs and AUTs in the scope of the present document.

4.3.2 Config_prod_VE

The configuration `config_prod_VE` shall be implemented to test APIs which are produced by FUTs in a VNF or EM. The test environment of the VNF/EM is the NFVI where the test is executed.

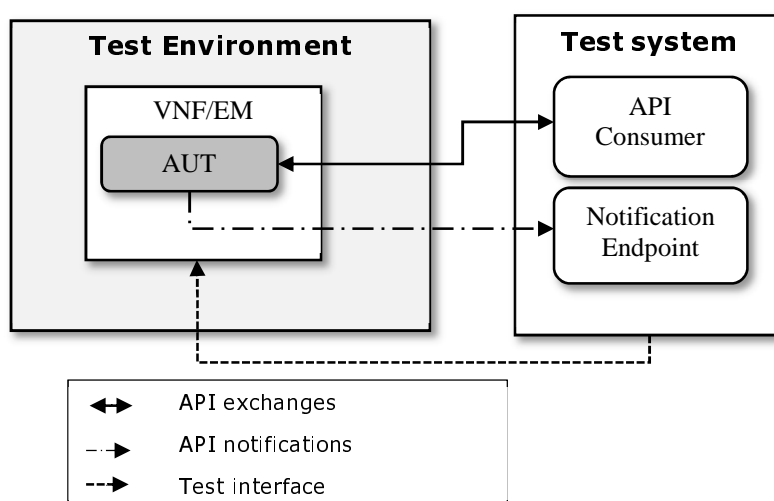


Figure 4.3.2-1: Configuration for tests of APIs with the FUTs as Producer run in a VNF/EM

4.3.3 Config_prod_VNFM

The configuration `config_prod_VNFM` shall be implemented to test APIs produced by FUTs which implement a VNFM. The test environment of the virtual element is the NFVI where the VE is executed.

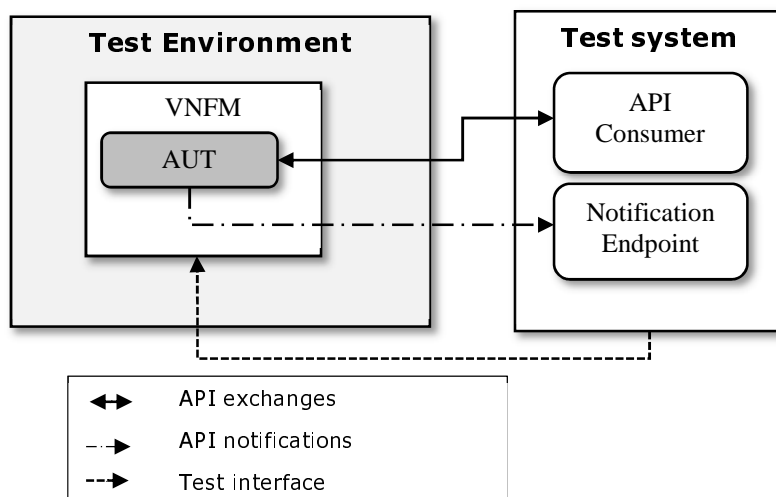


Figure 4.3.3-1: Configuration for tests of APIs with VNFM as Producer

4.3.4 Config_prod_NFVO

The configuration config_prod_NFVO shall be implemented to test APIs produced by FUTs which implement an NFVO. The test environment for the virtual element is an NFV platform that provides VNFM, VIM and NFVI.

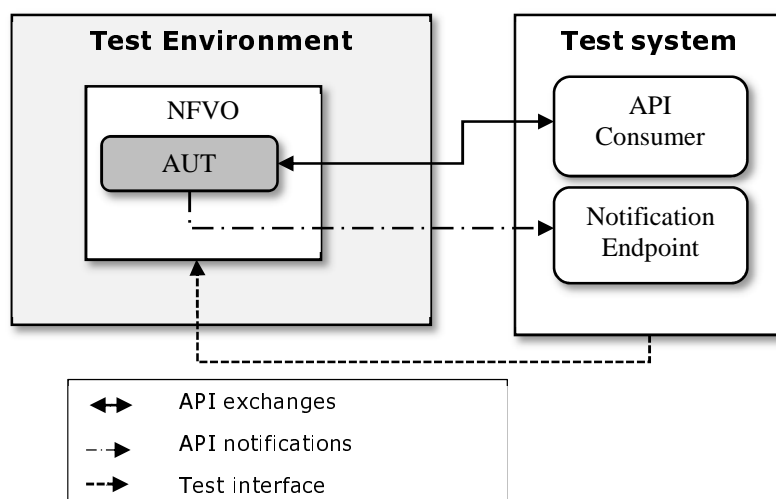


Figure 4.3.4-1: Configuration for tests of APIs with NFVO as Producer

4.3.5 Config_prod_VNFM_GRANT

The configuration config_prod_VNFM_GRANT shall be implemented to test APIs produced by FUTs that implement a VNFM for VNF LCM test cases where an Operation Grant is needed. The test environment of the virtual element consists of the NFVI where the VE is executed and an NFVO component that exposes the VNF Lifecycle Operation Granting API.

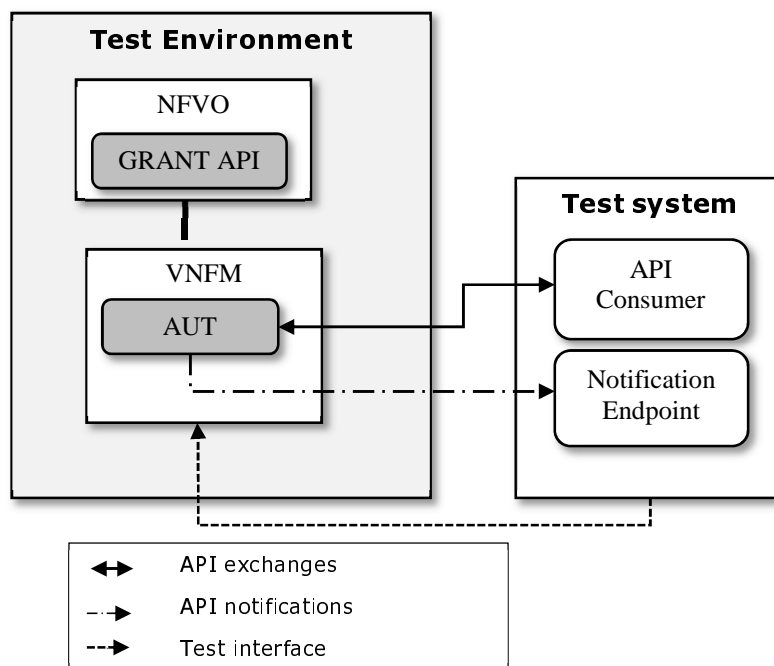


Figure 4.3.5-1: Configuration for tests of APIs with VNFM as Producer, where the VNFLifecycleOperationGranting API is required

4.3.6 Config_prod_Notif_Endpoint

The configuration config_prod_Notif_Endpoint shall be implemented to test any notification endpoint. The test environment for the virtual element is an NFV platform that provides an API Provider.

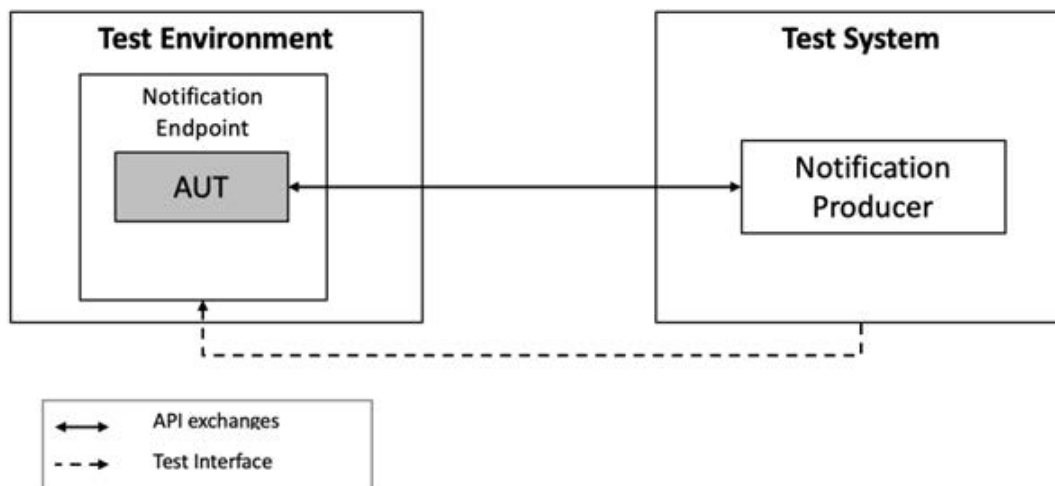


Figure 4.3.6-1: Configuration for tests of APIs where the Test System is an API Provider, and the AUT is a Notification endpoint

4.3.7 Config_prod_NFV-MANO

The configuration config_prod_NFV-MANO shall be implemented to test APIs produced by FUTs which implement NFV-MANO. The test environment for the virtual element is an NFV platform that provides NFVI.

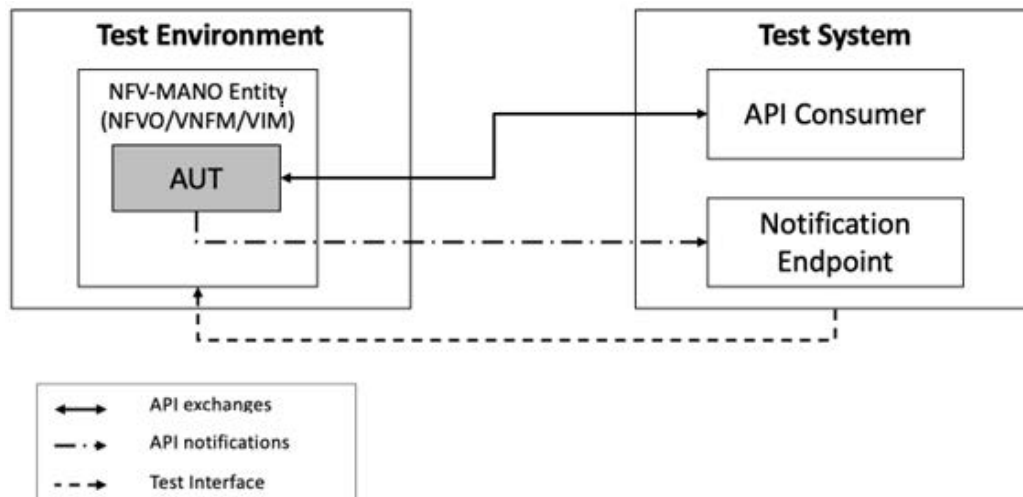


Figure 4.3.7-1: Configuration for tests of APIs with NFV-MANO as Producer

4.4 Void

4.5 Generic Test Description

4.5.1 General

The machine-readable language used for the test case implementations is the Robot Framework syntax language [i.2].

The Robot code for all test cases is available at <https://forge.etsi.org/rep/nfv/api-tests/tree/5.2.1> and in the accompanying annexes, which contain the tabular representations of the Test Descriptions. The primary source for the Test Descriptions (TDs) is the online repository at ETSI Forge.

Compliant NFV MANO API Conformance Test Systems shall implement the Test Descriptions as documented in the ETSI Forge repository and in the accompanying annexes, which follow the requirements set in the following clauses.

Together with the normative Test Descriptions, informative implementation of the tests steps is made available as well in the form of "Keywords in Robot Framework" to facilitate adoption of the present document.

Within the Robot Framework files present in the online repository, the Test Descriptions (which are normative) and test steps implementations (which are informative) can be identified by the heading under which they are grouped, as follows:

- the normative Tests Descriptions are grouped under the "*** Test Cases ***" heading; while
- the informative test steps implementations are grouped under the "*** Keywords ***" heading.

NOTE: The text for the headings needs to conform to the Robot Framework language syntax.

4.5.2 Test Description format

Test Descriptions specify the information required to run an "API conformance test". Test Descriptions are defined per test and table 4.5.2-1 defines the elements of the Test Description.

Table 4.5.2-1: Test Description entries

Elements	Definition
Test ID	The Test ID uniquely identifies the test. Naming convention: the Test ID is the clause number in the present document which includes the Test Descriptions.
Test Objective	The test objective clearly indicates which requirement is intended to be tested in the test. This part eases the understanding of the Test Description behaviour. This also eases the identification of the requirements, which were used as a basis for the test.
Pre-conditions	The pre-conditions field defines the conditions that the state of the SUT shall verify before undergoing the actual Test Description. In the process of the test, if the pre-conditions are not met, it leads to the assignment of an Inconclusive verdict. This field may include identifiers of other tests within the present document, which may be executed to bring the SUT into a state which verifies the mandated pre-conditions.
Reference	In the reference row, the Test Description writer indicates the clauses of the NFV-SOL specification where the tested requirements are expressed. This information is critical, because it justifies the existence and the behaviour of the Test Description. The reference row may refer to several clauses, separated by semi-colon.
Config ID	The identifier of the applicable test configuration. A test configuration defines the components that shall be included in the Test System and the SUT to execute the tests, along with their connections.
Applicability	Only optional functions which are required for this specific Test Description are listed in this field.
Post-conditions	Definition of the events that the SUT is expected to perform or shall not perform, if any, according to the NFV-SOL specifications and following the correct execution of the actions in the expected behaviour below. If the post-conditions concur in the definition of the verdict of the test, a step in the Test Case shall be explicitly provided. Multiple post conditions are separated by a semicolon, which expresses a logical AND relation.
Test Case	A set of steps of the test, carried out by the Test System, to verify conformance of the SUT to the Test Objective. The steps are written in the form of Keywords in the Robot Framework language [i.2]. The verdict of the Test shall be set to 'Pass' if all the applicable steps in the Test Case are successfully carried out and meet expected results criteria, it shall be set to 'Fail' if any of the steps fails. The Inconclusive verdict applies in all other cases.

The Test Descriptions are formatted as tables, as shown in the example in table 4.5.2-2.

Table 4.5.2-2: Test Description entries examples

Test Purpose	
Test ID	A.B.C.D.E.
Test Objective	Example Test Objective.
Pre-conditions	Precondition example.
Reference	Reference 1; reference 2.
Config ID	Configuration_example_id.
Applicability	Applicability statement example.
Post-conditions	Precondition 1; Precondition 2; Precondition 3.
Test Case	
Step Number One Step Number Two Steps are described using the following expressions: <ul style="list-style-type: none"> • GET all [resource] [expression] to query the URI of [resource] to the SUT by GET operation with [expression]. • GET [resource] [expression] to query an individual [resource] to the SUT by the GET operation with [expression]. • Send [HTTP method] Request to [purpose]/[HTTP method] [resource] to send request to SUT by [HTTP method], i.e. POST, PUT, PATCH, DELETE. Same as "Send [HTTP method] Request to [purpose]". • Check HTTP Response Status [expression] to check whether the status code in the response, which corresponds to the previous step request from the SUT, matches [expression]. • Check HTTP Response Header [expression] to check whether the contents of the HTTP header in the response, which corresponds to the previous step request from the SUT, matches [expression]. • Check HTTP Response Body [expression] to check whether the JSON of the body in the payload of the response, which corresponds to the previous step request from SUT, matches [expression]. • Check Postcondition [resource] Exists to check whether [resource] of the SUT is accessible by the GET operation. • Check Postcondition [resource] is [expression] to check whether [resource] matches [expression] by GET operation. • Check [notification name] Http Request Body [expression] Upon [notification name] received, check whether the JSON of the body in the payload of the HTTP post request from SUT matches [expression]. • Trigger [condition] (external action) The test environment prepares [condition] for the particular test to run. • Check [resource] [condition] to check whether the attribute of the [resource] in JSON in the payload of the response from the SUT matches [condition] according to the "Post-Conditions" or "Pre-conditions" by GET operation. 	

4.5.3 Scope of the tests

4.5.3.1 General

In addition to the material described in clause 4.5.1, the behaviours of the SUT shall be tested in terms of:

- expected handling of the messages received from Test System; and
- expected actions inside the SUT.

These two items above shall be checked by examining information retrieved from other operations performed by Test Systems, e.g. GET or appropriate notification messages. These checks of expected handling and internal actions shall be based on material provided in the referenced specifications, including all error cases.

NOTE: See also Annex L for special remarks on the above descriptions.

Further checking includes:

- The expected correct response messages are sent back to the Test system.

The above item shall be checked by analysing the received messages by Test Systems. These checks of response correctness shall be based on material provided in the referenced specifications, including all error cases.

4.5.3.2 General characteristics of the reference points

This clause describes the basic principles of ETSI NFV SOL API specifications, e.g. ETSI GS NFV-SOL 002 [2], ETSI GS NFV-SOL 003 [1] and ETSI GS NFV-SOL 005 [3], which are considered in the API conformance test.

From the API conformance test point of view, the reference points have the following characteristics:

- Asynchronous API and State handling by notifications: applicable for Os-Ma-Nfvo, Or-Vnfm:
 - Due to the varieties of the reasons, e.g. consuming time of process handling, "VNF Lifecycle Management interface" defined in ETSI GS NFV-SOL 003 [1], clause 5 and "NS Lifecycle Management interface" defined in ETSI GS NFV-SOL 005 [3], clause 6 are designed as asynchronous way, i.e. API designed is based on the combination of the specific methods with task resources and state handling by related attributes signalled via notification messages.
 - For example, the operation "Instantiate VNF" is triggered by a POST message with task "Instantiate VNF" sent from NFVO to VNFM. Then the VNFM sends a "provisional" response, "202 accepted", back to the NFVO, which means "accepted request" (ETSI GS NFV-SOL 003 [1], clause 5.4.4). Based on the subscriptions registered in VNFM, the appropriate notification message will be sent to the NFVO, with the state of the processing in the VNFM at that time. After finishing the process of instantiating the VNF in the VNFM, the notification message with the notification endpoint resource which is set to "operationState=COMPLETED" shall be sent (ETSI GS NFV-SOL 003 [1], clause 5.4.18), which means that the VNF has been instantiated. In order to verify these "Asynchronous API" characteristics, the Robot code implements a "workflow test" that includes a series of operations required for "VNF Lifecycle Management interface" and "NS Lifecycle Management interface" operations.
- Linked resource: applicable for Os-Ma-Nfvo, Ve-Vnfm, Or-Vnfm:
 - According to ETSI GS NFV-SOL 003 [1], clause 4.3.3.1, ETSI GS NFV-SOL 005 [3], clause 4.3.3.1 and ETSI GS NFV-SOL 002 [2], clause 4.3.3.1, "link" pointing to the resources which may be fetched by API consumer may be used on the reference point. It is used in order to avoid "big" response messages and to avoid repeating information-request messages. Therefore, the validity of such linked resources is also verified by testing, as same as resources in the response message itself.

NOTE 1: See also Annex L for the special remarks on "Linked resource".

- Request and response data values: applicable for Os-Ma-Nfvo, Ve-Vnfm, Or-Vnfm:
 - According to ETSI GS NFV-SOL 003 [1] and ETSI GS NFV-SOL 002 [2], clause 5.5, a VNF LCM request uses the relevant values, e.g. set in the valid VNFD, i.e. InstantiateVnfRequest, ScaleVnfRequest, ScaleVnfToLevelRequest, ChangeVnfFlavourRequest, TerminateVnfRequest, HealVnfRequest, OperateVnfRequest and ChangeExtVnfConnectivityRequest.
 - According to ETSI GS NFV-SOL 005 [3], clause 6.5, a NS LCM request uses the relevant values, e.g. set in the valid NSD, i.e. InstantiateNsRequest, ScaleNsRequest, UpdateNsRequest, TerminateNsRequest and HealNsRequest.
 - The test system verifies that each attribute of the response body uses relevant values, e.g. those set in the valid VNFD/NSD.

NOTE 2: See also clause 4.5.4.2.

- Grant exchange: applicable for Or-Vnfm:
 - According to ETSI GS NFV-SOL 003 [1], clause 9.1, the VNFM executes a Grant Request before certain LCM operations are performed in the direct mode. The test system verifies that each attribute of the Grant request body uses the relevant values, e.g. set in the valid VNFD.

NOTE 3: The value validation is not considered in the present document.

NOTE 4: The Robot code implements the "Individual Grant". The test for "Grant exchange in the VNF LCM operations" in the workflow test requires a specific test configuration "Config_prod_VNFM_GRANT", specified in clause 4.3, but "Grant exchange" part in the VNF LCM is not considered by the Robot code in the present document.

- VIM registration: applicable for Or-Vnfm:
 - According to ETSI GS NFV-SOL 003 [1] clause 4.4.1.6, the VNFM uses VIM-related information according to ETSI GS NFV-SOL 003 [1], clause C.5, the test system imports the latest VIM registration before testing.

NOTE 5: The test system imports the values for the VIM-related information via test configurations.

4.5.3.3 Basic behaviours of the API producer/consumer and verification steps

4.5.3.3.0 Introduction

Expected behaviours of the API producer specified in the NFV-SOL specifications targeted by the tests (i.e. ETSI GS NFV-SOL 002 [2], ETSI GS NFV-SOL 003 [1] and ETSI GS NFV-SOL 005 [3]) are categorized as follows:

- Producer sends event-triggered notifications based on consumer subscriptions.
- Producer executes the requested API operation.
- Consumer fetches the files/package info.

These three categories are expanded hereafter.

4.5.3.3.1 Producer sends event-triggered notifications based on consumer subscriptions

A typical example for this case is "Fault management". The producer sends a notification to the consumer with mechanisms "filter" and "callback". The Test System checks whether the notification aligns with the conditions of the subscription.

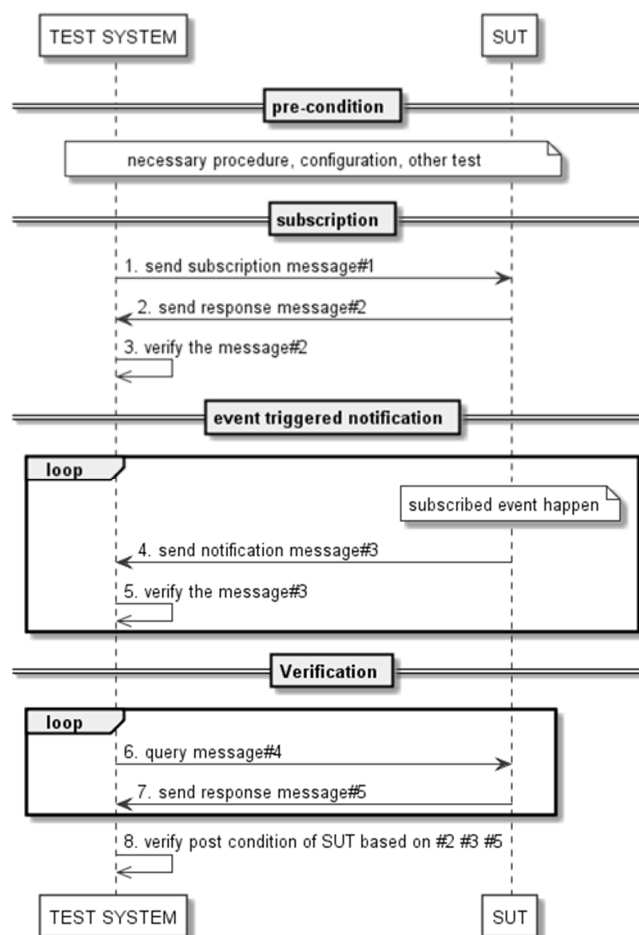


Figure 4.5.3.3.1-1: Verification procedure

The basic method of verification follows the steps illustrated in figure 4.5.3.3.1-1.

Preconditions: Before sending message#1, required features/functionalities are executed, e.g. API authentication, etc. Other tests as a precondition are executed beforehand if required:

- 1) Test System sends subscription message#1. Test System sets the data and attributes of the request body with correct/incorrect cardinalities and values.
- 2) Upon receiving the subscription message#1 from the Test System, the SUT shall handle this subscription according to the requirements defined in the NFV-SOL specifications, and the response message#2 are sent to the Test System.
- 3) Test System verifies the received response message#2, in terms of expected response code, expected HTTP header, HTTP body, cardinality of the responded data, fulfilment of the requirements described in the description field, and verification aspects shown below, etc.:
 - a) Verification aspects: 1-C, 2-C, 2-F, 2-G (see clause 4.5.4).
- 4) When the events that match the subscribed criteria specified in message#1 happen in the SUT, notification message#3 is sent to the Test System.

NOTE 1: How to make a subscribed event in SUT to match the criteria is out of the scope of the present document.

- 5) Test System verifies the received notification message#3, in terms of expected HTTP header, HTTP body, fulfilment of the requirements described in the description field, and verification aspects shown below, etc. steps 4 and 5 can be looped:
 - a) Verification aspects: 1-C, 2-C, 2-F, 2-G (see clause 4.5.4).

- 6) After the expected API messages exchange, the Test System may have some verification aspects that require retrieval of information from the SUT, e.g. linked resources embedded in the response message. In that case, the Test System may invoke a query message to the SUT.
- 7) In response to step 6, the SUT may send additional messages to the Test System. In this case, the verification is identical to step 5.
- 8) In addition to the verification performed in steps 2, 3, 5 and 7, the Test System verifies the expected post-conditions for the SUT.

NOTE 2: See also Annex L for the special remarks on "Received notification validation" on steps 6, 7 and 8.

4.5.3.3.2 Void

4.5.3.3.3 Producer executes the requested API operation

Based on the concept of an "Asynchronous API", the producer starts the operation by receiving a request message from the consumer, and sending some notifications back based on the operational status. The Test System checks whether the procedure is correct, i.e. starts with a request, sends notifications with an appropriate status, completes the operation and ensure the correct status of the resources and the producer.

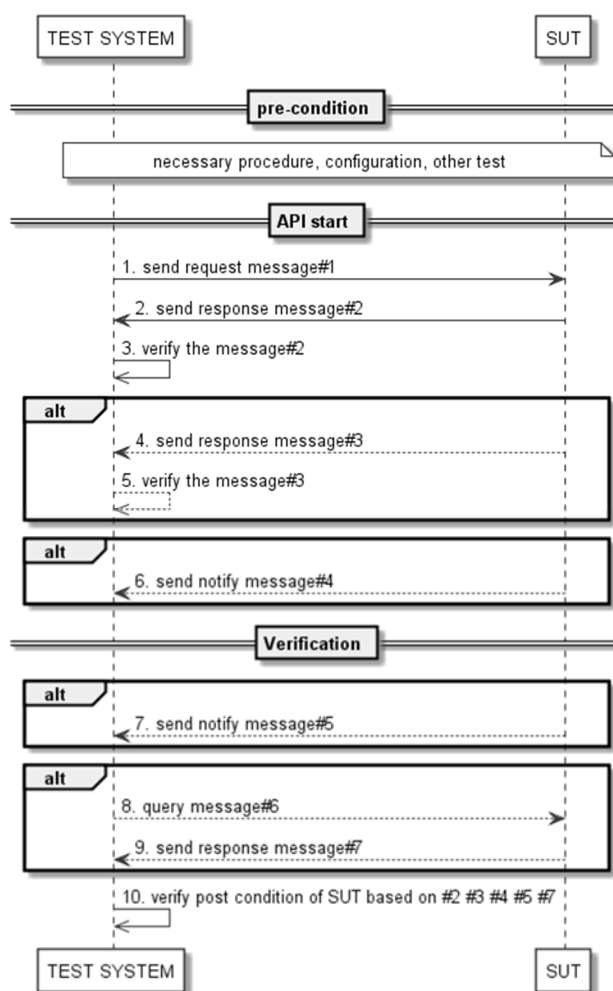


Figure 4.5.3.3.3-1: Verification procedure

The basic method of verification follows the steps illustrated in figure 4.5.3.3.3-1.

Preconditions: Before sending message#1, required features/functionalities are executed, e.g. API authentication, etc.
Other tests as precondition are executed beforehand if required:

- 1) Test System sends request message#1. Test System sets the data and attributes of the request body with correct/incorrect cardinalities and values.
- 2) Upon receiving the request message#1 from the Test System, the response message#2 is sent to the Test System with an appropriate response code and an HTTP body based on the result of the SUT executed procedure.
- 3) Test System verifies the received response message#2, in terms of expected response code, expected HTTP header, HTTP body, cardinality of the responded data, fulfilment of the requirements described in the description field, and verification aspects shown below, etc.:
 - a) Verification aspects: 1-C, 2-A, 2-C, 2-F, 2-G (see clause 4.5.4).

Optionally, the API may have more than 1 message exchange, in particular, for messages sent from the SUT to the Test System. In that case, this step follows step 2.

- 4) Identical to step 2.
- 5) Identical to step 3.
- 6) Optionally, the SUT may send notification messages to the Test System that indicate the status of the processing of the messages in the SUT, e.g. STARTING/PROCESSING, etc., based on the consumer subscriptions.
- 7) After the SUT completes the requested processing in the SUT, the SUT may send a notification message to the Test System, indicating the status of the processing as "COMPLETED".
- 8) After the expected API messages exchange, the Test System may have some verification steps that require retrieval of information from the SUT, e.g. linked resources embedded in the response message. In that case, the Test System may invoke a query message to the SUT.
- 9) The SUT may send additional messages to the Test System. In this case, the verification follows step 3.
- 10) In addition to the verification performed in steps 2, 3, 4, 5 and 7 (optional), the Test System verifies the expected post-conditions for the SUT.

4.5.3.3.4 Consumer fetches the files/package info

A typical example of this case is fetching files for a VNFD, a VNF package, an NSD and a PNFD. The Test System checks if the file or package info is correctly received.

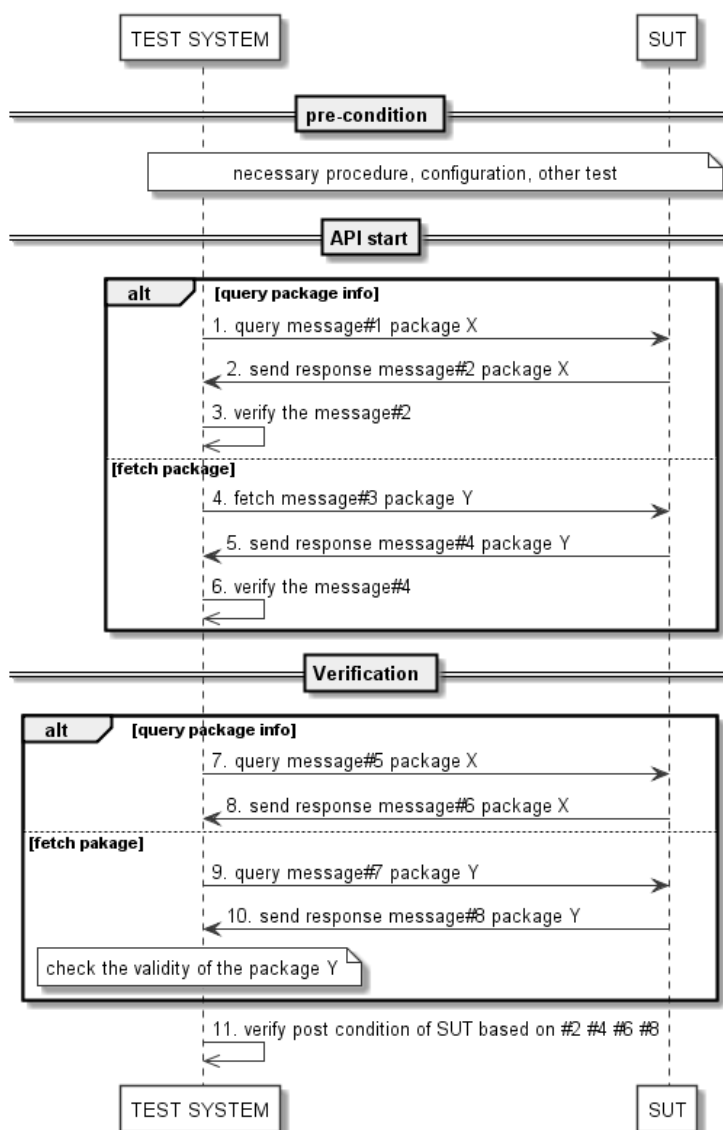


Figure 4.5.3.3.4-1: Verification procedure

The basic method of verification follows the steps illustrated in figure 4.5.3.3.4-1.

Preconditions: Before sending message#1, required features/functionality are executed, e.g. API authentication, etc. Other tests as a precondition are executed beforehand if required:

NOTE: Even though figure 4.5.3.3.1-1 shows the series of the message exchanges as a sequence flow, the Robot code/Test Descriptions implements the tests on an individual interface basis, i.e. the tests for querying the package info, the tests for fetching the packages.

- 1) Test System sends query message#1 for information about the package X. Test System sets the data and attributes of the request body with correct/incorrect cardinalities and values.
- 2) Upon receiving the query message#1 from the Test System, the SUT shall handle this subscription according to the requirements defined in the NFV-SOL specifications, the response message#2 is sent to the Test System.
- 3) Test System verifies the received response message#2, in terms of expected response code, expected HTTP header, HTTP body, cardinality of the responded data, fulfilment of the requirements described in the description field, and verification aspects shown below, etc.:
 - a) Verification aspects: 1-C, 2-C, 2-F, 2-G (see clause 4.5.4).

- 4) Alternatively to step 1, the Test System sends the fetch message#3 to retrieve the package Y. The Test System sets the data and attributes of the request body with correct/incorrect:
 - a) Verification aspects: 2-F in clause 4.5.4.
- 5) Upon receiving the fetch message#3 from the Test System, the SUT shall handle this request according to the requirements defined in the NFV-SOL specifications. The response message#4 with package Y is sent to the Test System with an appropriate response code and HTTP body based on the result of the SUT executed procedure.
- 6) Identical to step 3, for message#4.
- 7) After the expected API messages exchange, the Test System may have some verification steps that require retrieval of information from the SUT, e.g. linked resources embedded in the response message. In that case, the Test System may invoke a query message to the SUT. And in order to verify the package X information fetched in step 2, the Test System sends query message#5 to the SUT.
- 8) In response to step 7, the SUT may send additional messages to the Test System. In this case, the verification is identical to step 3.
- 9) The Test System may have some verification steps that require retrieval of information from the SUT, e.g. linked resources embedded in the response message. In that case, the Test System may invoke a query message to the SUT. And in order to verify the package Y information fetched in step 5, the Test System sends query message#7 to the SUT.
- 10) In response to step 9, the SUT may send additional messages to the Test System. In this case, the verification is identical to step 10.
- 11) In addition to the verification performed in steps 3 and 6, the Test System verifies the expected post-conditions of the SUT.

4.5.3.4 Workflow test considerations

The asynchronous API concept introduced in ETSI GS NFV-SOL 002 [2], ETSI GS NFV-SOL 003 [1] and ETSI GS NFV-SOL 005 [3] describes a set of lifecycle operations and relations among them. Asynchronous API, though described in the form of a flow as informative in the base specifications, is one of the key and critical design concepts of NFV-SOL specifications and for the deployment of NFV systems. Therefore, the present document includes a set of workflow Test Descriptions to test such characteristics and, in particular, to include checks on the post-conditions of the request for LCM operations.

NOTE 1: As an example, the request for the instantiation of a VNF expects as a post condition the asynchronous notifications of the status change of the VNF throughout the instantiation process. A workflow Test Description includes checks on the actual notifications of such status changes.

NOTE 2: The workflow Test Descriptions for ETSI GS NFV-SOL 002 [2] aspects are not considered in the present document.

The workflow Test Descriptions are listed in Annex B.

4.5.4 Verification

4.5.4.0 Introduction

Since the scope of the API conformance test includes syntax and semantics verifications, clause 4.5.4 defines the methodology of the verification executed by the Test System.

4.5.4.1 Common verification aspects

The common verification aspects checked by the Test System are defined in table 4.5.4.1-1.

Table 4.5.4.1-1: Common verification aspects

Verification aspects	Notes
1-A: URI handling	<p>If the SUT receives an invalid URI by HTTP method with either an invalid path or invalid values for the resources, the SUT shall return the appropriate error response code to the Test System according to the SOL specification.</p> <p>The Test System tests the support of HTTP over TLS version 1.2. See also 1-D: "Error reporting" in this table.</p> <p>See and follow clause 4.1 of ETSI GS NFV-SOL 013 [5].</p>
1-B: Attribute selectors	<p>Expected attributes or sub-resources shall be sent from the SUT according to the attribute selector parameter by a query operation from the Test System. The Test System tests all of the operations with parameters specified in table 5.3.2.1-1 of ETSI GS NFV-SOL 013 [5] if those work correctly.</p> <p>See and follow clause 5.3 of ETSI GS NFV-SOL 013 [5].</p>
1-C: HTTP header fields	<p>Expected HTTP headers shall be sent from the SUT. The Test System tests all of the operations with parameters, ensuring they work as specified in the target NFV-SOL specifications. See note 1.</p> <p>See and follow clause 4.2 of ETSI GS NFV-SOL 013 [5].</p>
1-D: Error reporting	<p>In case of some types of errors occurred in the SUT, the error response with type ProblemDetails data are returned to the Test System. The Test System tests all attributes of the ProblemDetails for error Response Codes explicitly specified in each of the Response body part of the tables for "Details of the request/response on the resources" in the baseline NFV-SOL specifications regarding the error experienced by the SUT.</p> <p>See note 2.</p> <p>See and follow clause 6 of ETSI GS NFV-SOL 013 [5].</p>
1-E: API authorization	<p>For the case of making an API call, OAuth2.0 is required for authentication between the API consumer and the API producer. For the case of the notification, OAuth2.0 or HTTP Basic authentication is used. The Test System tests the OAuth2.0 process, including authorization scope value. From the API producer point of view, if the passed access token is invalid, the API producer should return appropriate errors, e.g. 401 unauthorized, 403 insufficient scope.</p> <p>For conformance purposes, AUTHORIZATION_HEADER = "Authorization" shall be used.</p> <p>See and follow clause 8 of ETSI GS NFV-SOL 013 [5].</p>
NOTE 1: See also Annex L for the special remarks on "1-C: HTTP header fields".	
NOTE 2: See also Annex L for the special remarks on "1-D: Error reporting".	

4.5.4.2 Verification aspects for individual APIs

The verification aspects for individual APIs checked by the Test System are defined in table 4.5.4.2-1.

Table 4.5.4.2-1: Common verification aspects for each API

Verification aspects	Notes
2-A: Asynchronous API	As described in clause 4.5.3, a typical behaviour of the Asynchronous API is that the requested behaviours are not completed in a single response message, but rather delivered as a combination of notifications. See note 1.
2-B: Void	Void.
2-C: HTTP Method support	The Test System verifies all supported HTTP methods, e.g. all types of the specified method with all types of the specified response codes.
2-D: Void	Void.
2-E: Void	Void.
2-F: Response messages validation	The Test System checks the syntax of the HTTP header and body, the cardinalities and the conditions described in the description part and/or the "NOTE" part of the tables for definition of the data types in the NFV-SOL specifications. The "link" attributes of the data model are also subject to validation in terms of the intended values of the resources in the SUT. See note 2.

Verification aspects	Notes
2-G: Validation of the resources in the SUT after API operation	<p>The Test System checks whether the related resources in the SUT are in the expected status after having received the response messages, by checking the notifications from the SUT or by initiating a GET request towards the SUT.</p> <p>The tests make use of NSD and VNFD in order to enhance the verification aspects of the test suite. For example, when a VNF is instantiated, the values in the vnflInstance should reflect the content of the VNFD.</p> <p>The verified parameters are taken from tables D.2-1 and D.3-1 in ETSI GS NFV-SOL 001 [6]. During a test, the focus is on verifying parameters in the responses from the SUT. Where a response includes one or more parameters from ETSI GS NFV-SOL 001 [6], a validation is performed. A validation checks for equality, such that the parameter in the response = parameter from VNFD.</p> <p>This type of validation is done on the following reference points and APIs:</p> <p>Os-Ma-nfvo Reference Point:</p> <ul style="list-style-type: none"> • NSD Management API • NSD Lifecycle Management API • VNF Package Management API <p>Or-Vnfm Reference Point:</p> <ul style="list-style-type: none"> • VNF Lifecycle Management API • VNF Lifecycle Operation Granting API • VNF Package Management API <p>Ve-Vnfm Reference Point:</p> <ul style="list-style-type: none"> • VNF Lifecycle Management API
2-H: Void	Void.
NOTE 1: The Robot code implements "workflow test", which includes a series of operations required for the VNF Lifecycle Management interface and NS Lifecycle Management interface. Special considerations on "workflow Test Descriptions" are described in clause 4.5.3.4.	
NOTE 2: See also Annex L for the special remarks on "2-F: Response messages validation".	

4.5.4.3 Verification aspects not considered

This version of the present document and test code is not considering the aspects for verification listed in table 4.5.4.3-1.

Table 4.5.4.3-1: Exception of the verification aspects for API conformance test

Verification aspects not considered by this version of the present document and test code	Notes
Multiple times notification test	Multiple times notification test based on the conditions subscribed.

5 Void

6 Void

7 Void

Annex A (informative): Known Issues

Table A-1 lists Test Descriptions that have known issues.

Table A-1: Known issues

Clause IDs	Issue Description
None	None

Table A-2 lists the missing Test Descriptions.

Table A-2: Missing Test Descriptions

Clause IDs	Description
None	None

Annex B (informative): Workflow Test Descriptions

Table B-1 contains the exhaustive list of Test Descriptions that are workflows, as defined in clause 4.5.3.4.

Table B-1: List of workflow Test Descriptions

Test ID	Test Title
5.3.2.18.1	NS Instance Creation
5.3.2.18.2	NS Instance Creation with DISABLED Network Service Descriptor
5.3.2.19.1	NS Instance Deletion
5.3.2.20.1	Heal Flow of NS lifecycle management operations
5.3.2.21.1	Instantiate Flow of NS lifecycle management operations
5.3.2.22.1	Scale Flow of NS lifecycle management operations
5.3.2.23.1	Terminate Flow of NS lifecycle management operations
5.3.2.24.1	Update Flow of NS lifecycle management operations
7.3.1.19.1	Cancel a VNF LCM Operation - STARTING
7.3.1.19.2	Cancel a VNF LCM Operation - PROCESSING - ROLLING_BACK
7.3.1.20.1	Change external connectivity of VNF Workflow
7.3.1.21.1	Change VNF Flavour Workflow
7.3.1.22.1	Create VNF Instance Resource
7.3.1.23.1	Delete VNF Instance Resource
7.3.1.24.1	Fail a VNF LCM Operation Workflow
7.3.1.25.1	Heal a VNF Instance
7.3.1.26.1	VNF Instantiation
7.3.1.26.2	VNF Instantiation with attribute instantiationLevelId
7.3.1.26.3	VNF Instantiation with attribute targetScaleLevelInfo
7.3.1.27.1	Modify info of a VNF Instance
7.3.1.28.1	Operate a VNF Instance
7.3.1.29.1	Retry VNF LCM Operation - Successful
7.3.1.29.2	Retry VNF LCM Operation - Unsuccessful
7.3.1.30.1	Rollback a VNF LCM Operation - Successful
7.3.1.30.2	Rollback VNF LCM Operation - Unsuccessful
7.3.1.31.1	VNF Instance Scale To Level
7.3.1.32.1	VNF Instance Scale Out
7.3.1.33.1	Terminate a VNF Instance

Annex C:
Void

Annex D (normative): Word format presentation of the test suite for the Os-Ma-nfvo Reference Point

The Microsoft® Word-format presentation of the high-level tests for the Os-Ma-nfvo is generated from the Robot Framework test suite using the robot2doc tool available at [i.3]. The format is provided to assist readers who are not familiar with the Robot Framework syntax language.

The Microsoft® Word format presentation of the high-level tests for the Os-Ma-nfvo is the file Tests-SOL005-OsMaNfvo.docx contained in archive gs_nfv-tst010v050201p0.zip, which accompanies the present document.

The normative Test Descriptions and the informative test steps implementations are available in the Robot Framework at <https://forge.etsi.org/rep/nfv/api-tests/tree/5.2.1/SOL005>.

Some Test Descriptions that were present in previous versions of the present document are indicated as "void". This is because the Test Descriptions are not applicable to version 5.2.1 of ETSI GS NFV-SOL 005 [3].

Annex E (normative): Word format presentation of the test suite for the Ve-Vnfm Reference Point

The Microsoft® Word format presentation of the high-level tests for the Ve-Vnfm is generated from the Robot Framework test suite using the robot2doc tool available at [i.3]. The format is provided to assist readers who are not familiar with the Robot Framework syntax language.

The Microsoft® Word format presentation of the high-level tests for the Ve-Vnfm is the file Tests-SOL002-VeVnfm.docx contained in archive gs_nfv-tst010v050201p0.zip, which accompanies the present document.

The normative Test Descriptions and the informative test steps implementations are available in the Robot Framework at <https://forge.etsi.org/rep/nfv/api-tests/tree/5.2.1/SOL002>.

Some Test Descriptions that were present in previous versions of the present document are indicated as "void". This is because the Test Descriptions are not applicable to version 5.2.1 of ETSI GS NFV-SOL 002 [2].

Annex F (normative): Word format presentation of the test suite for the Or-Vnfm Reference Point

The Microsoft® Word format presentation of the high-level tests for the Or-Vnfm is generated from the Robot Framework test suite using the robot2doc tool available at [i.3]. The format is provided to assist readers who are not familiar with the Robot Framework syntax language.

The Microsoft® Word format presentation of the high-level tests for the Or-Vnfm is the file Tests-SOL003-OrVnfm.docx contained in archive gs_nfv-tst010v050201p0.zip, which accompanies the present document.

The normative Test Descriptions and the informative test steps implementations are available in the Robot Framework at <https://forge.etsi.org/rep/nfv/api-tests/tree/5.2.1/SOL003>.

Some Test Descriptions that were present in previous versions of the present document are indicated as "void". This is because the Test Descriptions are not applicable to version 5.2.1 of ETSI GS NFV-SOL 003 [1].

Annex G (normative): Word format presentation of the test suite for NFV-MANO Management

The Microsoft® Word format presentation of the high-level tests for the NFV-MANO Management is generated from the Robot Framework test suite using the robot2doc tool available at [i.3]. The format is provided to assist readers who are not familiar with the Robot Framework syntax language.

The Microsoft® Word format presentation of the high-level tests for the NFV-MANO Management is the file Tests-SOL009-NFV-MANO-mgmt.docx contained in archive gs_nfv-tst010v050201p0.zip, which accompanies the present document.

The normative Test Descriptions and the informative test steps implementations are available in the Robot Framework at <https://forge.etsi.org/rep/nfv/api-tests/tree/5.2.1/SOL009>.

Some Test Descriptions that were present in previous versions of the present document are indicated as "void". This is because the Test Descriptions are not applicable to version 5.2.1 of ETSI GS NFV-SOL 009 [7].

Annex H (normative): Word format presentation of the test suite for the Or-Or reference point

H.1 General

The Microsoft® Word format presentation of the high-level tests for the Or-Or is generated from the Robot Framework test suite using the robot2doc tool available at [i.3]. The format is provided to assist readers who are not familiar with the Robot Framework syntax language.

The Microsoft® Word format presentation of the high-level tests for the Or-Or is the file Tests-SOL011-OrOr.docx contained in archive gs_nfv-tst010v050201p0.zip, which accompanies the present document.

The normative Test Descriptions and the informative test steps implementations are available in the Robot Framework at <https://forge.etsi.org/rep/nfv/api-tests/tree/5.2.1/SOL011>.

In addition to the ETSI GS NFV-SOL 011 [8] tests listed in the electronic attachment, the ETSI GS NFV-SOL 011 [8] tests make use of applicable ETSI GS NFV-SOL 005 [3] tests. The following ETSI GS NFV-SOL 011 [8] interfaces reference the related SOL005 interfaces.

NOTE: The tests for ETSI GS NFV-SOL 011 [8] are for version 4.5.1 and not version 5.2.1.

H.2 SOL011 NSD Management

The interface follows the provisions specified in clause 5 of ETSI GS NFV-SOL 005 [3] for the NSD management interface, except that the producer is NFVO-N and the consumer is NFVO-C.

The following SOL005 NSD Management conformance tests are applicable only to ETSI GS NFV-SOL 011 [8]:

Resource	Applicable Test IDs
NS Descriptors	5.3.1.1.1-10, 5.3.1.1.1-15-17
Individual NS Descriptors	5.3.1.2.1-2

H.3 SOL011 NS Lifecycle Management

The interface follows the provisions specified in clause 6 of ETSI GS NFV-SOL 005 [3] for the NS lifecycle management interface, except that the producer is NFVO-N and the consumer is NFVO-C.

All of the SOL005 NS Lifecycle Management tests are applicable to ETSI GS NFV-SOL 011 [8], **except for** the following ones:

Resource	Not Applicable Test IDs
Update NS task	5.3.2.5.1-6, 5.3.2.24.1

H.4 SOL011 NS Performance Management

The interface follows the provisions specified in clause 7 of ETSI GS NFV-SOL 005 [3] for the performance management interface, except that the producer is NFVO-N and the consumer is NFVO-C.

All of the ETSI GS NFV-SOL 005 [3] NS Performance Management tests are applicable to ETSI GS NFV-SOL 011 [8].

H.5 SOL011 NS Fault Management

The interface follows the provisions specified in clause 8 of ETSI GS NFV-SOL 005 [3] for the fault management interface, except that the producer is NFVO-N and the consumer is NFVO-C.

All of the ETSI GS NFV-SOL 005 [3] NS Fault Management tests are applicable to ETSI GS NFV-SOL 011 [8].

Annex I (normative): Word format presentation of the test suite for NFV-MANO Policy Management

The Microsoft® Word format presentation of the high-level tests for the NFV-MANO Policy Management is generated from the Robot Framework test suite using the robot2doc tool available at [i.3]. The format is provided to assist the readers who are not familiar with the Robot Framework syntax language.

The Microsoft® Word format presentation of the high-level tests for the NFV-MANO Policy Management is the file Tests-SOL012-NFV-MANO-Policy-mgmt.docx contained in archive gs_nfv-tst010v050201p0.zip, which accompanies the present document.

The normative Test Descriptions and the informative test steps implementations are available in the Robot Framework at <https://forge.etsi.org/rep/nfv/api-tests/tree/5.2.1/SOL012>.

Some Test Descriptions that were present in previous versions of the present document are indicated as "void". This is because the Test Descriptions are not applicable to version 4.5.1 of ETSI GS NFV-SOL 012 [11].

NOTE: The tests for ETSI GS NFV-SOL 012 [11] are for version 4.5.1 and not version 5.2.1.

Annex IA (normative): Word format presentation of the test suite for Telco Cloud data analytics service

The Microsoft® Word format presentation of the high-level tests for Telco Cloud data analytics service is generated from the Robot Framework test suite using the robot2doc tool available at [i.3]. The format is provided to assist readers who are not familiar with the Robot Framework syntax language.

The Microsoft® Word format presentation of the high-level tests for Telco Cloud Data Analytics Service is the file Tests-SOL025-TelcoCloudDataAnalytics.docx contained in archive gs_nfv-tst010v050201p0.zip, which accompanies the present document.

The normative Test Descriptions and the informative test steps implementations are available in the Robot Framework at <https://forge.etsi.org/rep/nfv/api-tests/tree/5.2.1/SOL025>.

Annex J (informative): Inter-connected Test Descriptions

To run the multiple tests implemented by Robot Framework in a connected manner within the test system, the present document/work includes a set of interconnected Test Descriptions. Table J-1 contains the exhaustive list of Test Descriptions that are interconnected Test Descriptions.

NOTE 1: In this version of the present document, each of the interconnected test descriptions includes the numbers of the test descriptions which are defined in Annex F, i.e. the tests for the Or-Vnfm, applying to ETSI GS NFV-SOL 003 [1].

NOTE 2: The inter-connected test descriptions listed in table J-1 are not just connecting individual test descriptions which are defined in Annex F, i.e. the tests for the Or-Vnfm, applying to ETSI GS NFV-SOL 003 [1], but have some Robot code implementation level modifications in order to inter-connect. See more details in Robot code repository described in the file Tests-InterconnectedTestCases.docx contained in archive gs_nfv-tst010v050201p0.zip, which accompanies the present document.

Table J-1: List of inter-connected Test Descriptions

Test ID of interconnected test Descriptions	Test Title	Test Descriptions included in the interconnected Test Descriptions (with reference to the Test ID of the individual Test Descriptions)
11.3.1.1.1	Create a new VNF instance resource and query information about multiple and individual VNF instance	GET Individual VNF Package (7.3.3.2.1) -> POST Create a new vnflInstance (7.3.1.1.1) -> GET Information about multiple VNF instances (7.3.1.1.2) -> GET Information about an individual VNF Instance (7.3.1.2.2)
11.3.1.2.1	Modify an individual VNF instance resource	GET Individual VNF Package (7.3.3.2.1) -> POST Create a new vnflInstance (7.3.1.1.1) -> PATCH Individual VNFInstance (7.3.1.2.4)
11.3.1.3.1	Delete an individual VNF instance	GET Individual VNF Package (7.3.3.2.1) -> POST Create a new vnflInstance (7.3.1.1.1) -> DELETE Individual VNFInstance (7.3.1.2.7)
11.3.1.4.1	Instantiate a VNF instance	GET Individual VNF Package (7.3.3.2.1) -> POST Create a new vnflInstance (7.3.1.1.1) -> VNF Instantiation (7.3.1.26.1)
11.3.1.5.1	Terminate a VNF instance	GET Individual VNF Package (7.3.3.2.1) -> POST Create a new vnflInstance (7.3.1.1.1) -> VNF Instantiation (7.3.1.26.1) -> Terminate a VNFInstance (7.3.1.33.1)
11.3.1.6.1	Retrieve VNF instance Operation Occurrences	GET Individual VNF Package (7.3.3.2.1) -> POST Create a new vnflInstance (7.3.1.1.1) -> VNF Instantiation (7.3.1.26.1) -> GET status information about multiple VNF LCM Operation OCC (7.3.1.11.2) -> Get Individual VNF LCM Operation occurrence (7.3.1.12.2)
11.3.1.7.1	Create a new subscription and query information about multiple and individual subscription resources	GET Individual VNF Package (7.3.3.2.1) -> POST Create a new subscription (7.3.1.17.1) -> GET Subscriptions (7.3.1.17.4) -> GET Individual Subscription (7.3.1.18.2)
11.3.1.8.1	Delete individual subscription	GET Individual VNF Package (7.3.3.2.1) -> POST Create a new subscription (7.3.1.17.1) -> GET Subscriptions (7.3.1.17.4) -> GET Individual Subscription (7.3.1.18.2) -> DELETE an individual subscription (7.3.1.18.5)
11.3.1.9.1	Scale Out a VNF instance	GET Individual VNF Package (7.3.3.2.1) -> POST Create a new vnflInstance (7.3.1.1.1) -> VNF Instantiation (7.3.1.26.1) -> VNF Instance Scale Out (7.3.1.32.1)
11.3.1.10.1	Scale In a VNF instance	GET Individual VNF Package (7.3.3.2.1) -> POST Create a new vnflInstance (7.3.1.1.1) -> VNF Instantiation (7.3.1.26.1) -> GET information about multiple VNF instances (7.3.1.1.2) -> VNF Instance Scale In

Test ID of interconnected test Descriptions	Test Title	Test Descriptions included in the interconnected Test Descriptions (with reference to the Test ID of the individual Test Descriptions)
11.3.1.11.1	Heal a VNF instance	GET Individual VNF Package (7.3.3.2.1) -> POST Create a new vnflInstance (7.3.1.1.1) -> VNF Instantiation (7.3.1.26.1) -> GET Information about an individual VNF Instance (7.3.1.2.2) -> VNF Instance Heal
11.3.1.12.1	VNF instance Change External Connectivity	GET Individual VNF Package (7.3.3.2.1) -> POST Create a new vnflInstance (7.3.1.1.1) -> VNF Instantiation (7.3.1.26.1) -> GET Information about an individual VNF Instances (7.3.1.1.2) -> VNF Change External Connectivity
11.3.1.13.1	Retry a VNF lifecycle operation if the operation is in FAILED_TEMP state	GET Individual VNF Package (7.3.3.2.1) -> POST Create a new vnflInstance (7.3.1.1.1) -> VNF Instantiation (7.3.1.26.1) -> GET status information about multiple VNF LCM Operation OCC (7.3.1.11.2) -> Check operation resource state is FAILED_TEMP -> Post Retry operation task (7.3.1.13.1)
11.3.1.14.1	Rollback on a VNF lifecycle instantiation operation	GET Individual VNF package (7.3.3.2.1) -> POST Create a new vnflInstance (7.3.1.1.1) -> VNF Instantiation (7.3.1.26.1) -> GET status information about multiple VNF LCM Operation OCC (7.3.1.11.2) -> Post Rollback operation task (7.3.1.14.1)
11.3.1.15.1	Create a new subscription and verify notification is generated	GET Individual VNF Package (7.3.3.2.1) -> POST Create a new subscription (7.3.1.1.1) -> GET Subscriptions (7.3.1.17.4) -> GET Individual Subscription (7.3.1.18.2) -> VNF LCM Operation Occurrence Notification (7.3.1.36.1) -> Test the Notification Endpoint - Successful (7.3.1.36.4)
11.3.1.16.1	Retry a VNF Scale Out operation that fails	GET Individual VNF Package (7.3.3.2.1) -> POST Create a new vnflInstance (7.3.1.1.1) -> VNF Instantiation (7.3.1.26.1) -> VNF Instance Scale Out - will fail -> Check VNF LCM operation is failed -> Post Retry operation task(7.3.1.13.1)
11.3.1.17.1	Retry a VNF Change External Connectivity operation that fails	GET Individual VNF Package (7.3.3.2.1) ->POST Create a new vnflInstance (7.3.1.1.1) ->VNF Instantiation (7.3.1.26.1) ->VNF Change External Connectivity - will fail ->Check VNF LCM operation is failed ->Post Retry operation task (7.3.1.13.1)
11.3.1.18.1	Rollback on a VNF Scale Out operation	GET Individual VNF Package (7.3.3.2.1) ->POST Create a new vnflInstance (7.3.1.1.1) ->VNF Instantiation (7.3.1.26.1) ->VNF Instance Scale out - will fail ->Check VNF LCM operation is failed ->Post Rollback operation task (7.3.1.14.1)

Annex K (informative): TD Change history

K.1 Release 5.1.1

K.1.1 New TDs for the new testing aspects

Table K.1.1-1 contains the list of Test Descriptions that are new in terms of new testing aspects in this version of present document.

Table K.1.1-1: List of Test Descriptions that are new in terms of new testing aspects

Test ID	Test Title
5.3.6.7.1	POST Individual CCMs NFVI Capacity Information - Method not implemented
5.3.6.7.2	GET Information about an Individual CCMs NFVI Capacity Information
5.3.6.7.3	GET Information about an Individual CCMs NFVI Capacity Information using filter query parameter
5.3.6.7.4	GET Information about an Individual CCMs NFVI Capacity Information - Not Found
5.3.6.7.5	PUT Individual CCMs NFVI Capacity Information - Method not implemented
5.3.6.7.6	PATCH Individual CCMs NFVI Capacity Information
5.3.6.7.7	DELETE Individual CCMs NFVI Capacity Information - Method Not implemented
5.3.6.7.8	GET Information about an Individual CCMs NFVI Capacity Information with permitted authorization scope
5.3.6.7.9	GET Information about an Individual CCMs NFVI Capacity Information with not permitted authorization scope
6.3.5.3.11	POST Instantiate a vnflInstance with non permitted authorization scope
7.3.1.3.9	Post Instantiate Individual VNFInstance - Unauthorized
7.3.1.4.10	POST Scale a vnflInstance Scale Vertical - Compute
7.3.1.4.11	POST Scale a vnflInstance Scale Vertical - Storage
7.3.1.4.12	POST Scale a vnflInstance Scale Vertical - Oscontainer
7.3.1.10.8	POST Change external VNF connectivity - Conflict
8.3.1.10.9a	POST new subscription - Unprocessable Content
11.3.1.11.1	Heal a VNF instance
11.3.1.12.1	VNF instance Change External Connectivity
11.3.1.13.1	Test that POST method initiates retrying a VNF lifecycle operation if the operation is in FAILED_TEMP state
11.3.1.14.1	Test that POST method initiates rollback on a VNF lifecycle operation

K.1.2 New TDs that were supposed to be included but missing in the previous versions of the present document

Table K.1.2-1 contains the list of Test Descriptions that were supposed to be included but missing in the previous versions of the present document.

Table K.1.2-1: List of Test Descriptions that were supposed to be included but missing in the previous versions of the present document

Test ID	Test Title
6.3.4.8.4	Test the Notification Endpoint - Successful
6.3.4.8.5	PUT Notification endpoint - Method Not Implemented
6.3.4.8.6	PATCH Notification endpoint - Method Not Implemented
6.3.4.8.7	DELETE Notification endpoint - Method Not Implemented
6.3.5.1.16	GET information about multiple VNF instances - Bad Request Response too big
6.3.5.1.17	GET information about multiple VNF instances to get Paged Response
6.3.5.1.17a	GET information about multiple VNF instances as a Paged Response with nextpage_opaque_marker parameter
6.3.5.2.11	Get Information about an individual VNF Instance - Not Found
6.3.5.11.17	GET information about multiple VNF instances - Bad Request Response too big
6.3.5.12.8	Get status information about individual VNF instances - Not Found
6.3.5.16.10	POST Cancel operation task
6.3.5.21.4	Test the Notification Endpoint - Successful
6.3.5.21.5	PUT Notification endpoint - Method Not Implemented
6.3.5.21.6	PATCH Notification endpoint - Method Not Implemented
6.3.5.21.7	DELETE Notification endpoint - Method Not Implemented
7.3.1.1.19	POST Create a new vnflInstance - UNPROCESSABLE CONTENT
7.3.1.36.4	Test the Notification Endpoint - Successful
7.3.1.36.5	PUT Notification endpoint - Method Not Implemented
7.3.1.36.6	PATCH Notification endpoint - Method Not Implemented
7.3.1.36.7	DELETE Notification endpoint - Method Not Implemented
7.3.3.12.3	Test the Notification Endpoint - Successful
7.3.3.12.4	PUT Notification endpoint - Method Not Implemented
7.3.3.12.5	PATCH Notification endpoint - Method Not Implemented
7.3.3.12.6	DELETE Notification endpoint - Method Not Implemented
7.3.4.10.3	Test the Notification Endpoint - Successful
7.3.4.10.4	PUT Notification endpoint - Method Not Implemented
7.3.4.10.5	PATCH Notification endpoint - Method Not Implemented
7.3.4.10.6	DELETE Notification endpoint - Method Not Implemented
7.3.5.7.4	Test the Notification Endpoint - Successful
7.3.5.7.5	PUT Notification endpoint - Method Not Implemented
7.3.5.7.6	PATCH Notification endpoint - Method Not Implemented
7.3.5.7.7	DELETE Notification endpoint - Method Not Implemented
7.3.6.8.3	Test the Notification Endpoint - Successful
7.3.6.8.4	PUT Notification endpoint - Method Not Implemented
7.3.6.8.5	PATCH Notification endpoint - Method Not Implemented
7.3.6.8.6	DELETE Notification endpoint - Method Not Implemented
7.3.7.4.2	Test the Notification Endpoint - Successful
7.3.7.4.3	PUT Notification endpoint - Method Not Implemented
7.3.7.4.4	PATCH Notification endpoint - Method Not Implemented
7.3.7.4.5	DELETE Notification endpoint - Method Not Implemented

K.2 Release 5.2.1

K.2.1 New TDs for the new testing aspects

Table K.2.1-1 contains the list of Test Descriptions that are new in terms of new testing aspects in this version of present document.

Table K.2.1-1: List of Test Descriptions that are new in terms of new testing aspects

Test ID	Test Title
5.3.2.1.18	SOL005 POST Create a new NsInstance with invalid resource
5.3.2.2.9	SOL005 GET Information about an individual NS Instance with invalid resource
5.3.2.3.9	SOL005 POST Instantiate a nsInstance with invalid resource
5.3.2.6.8	SOL005 POST Heal a NSInstance with invalid resource
5.3.2.8.2	SOL005 GET status information about multiple NS LCM occurrences with invalid resource

Test ID	Test Title
5.3.2.9.7	SOL005 PATCH status information about Individual NS LCM occurrences with not permitted authorization scope
5.3.2.9.8	SOL005 Get status information about Individual NS LCM occurrence with invalid resource
5.3.2.16.8	SOL005 GET Information about an individual subscription with invalid resource
6.3.5.1.18	SOL002 POST Create a new vnflInstance - Unauthorized
6.3.5.2.12	SOL002 Get Information about an individual VNF Instance - Unauthorized
6.3.5.3.12	SOL002 POST Instantiate a vnflInstance - Not Found
6.3.5.4.10	SOL002 Post Scale a vnflInstance - Unauthorized
6.3.5.5.10	SOL002 POST Scale a vnflInstance to level with Attribute instantiationLevelId - Unauthorized
6.3.5.6.10	SOL002 POST Change deployment flavour of a vnflInstance - Unauthorized
6.3.5.7.9	SOL002 POST Terminate a vnflInstance Not Found
6.3.5.7.10	SOL002 POST Terminate a vnflInstance - Unauthorized
6.3.5.8.10	SOL002 POST Heal a vnflInstance - Unauthorized
6.3.5.9.10	SOL002 POST Operate a vnflInstance - Unauthorized
6.3.5.10.9	SOL002 POST Change external VNF connectivity - VNF instance not found
6.3.5.10.10	SOL002 POST Change external VNF connectivity - Unauthorized
6.3.5.11.18	SOL002 GET status information about multiple VNF LCM operation occurrences - Unauthorized
6.3.5.12.9	SOL002 Get Individual VNF LCM Operation occurrences - Unauthorized
6.3.5.13.10	SOL002 POST Retry operation task - Unauthorized
6.3.5.14.10	SOL002 POST Rollback operation task - Unauthorized
6.3.5.15.10	SOL002 POST Fail operation task - Unauthorized
6.3.5.16.11	SOL002 Post Cancel operation task - Unauthorized
6.3.5.17.19	SOL002 POST Create a new subscription - Unauthorized
6.3.5.18.8	SOL002 GET Individual Subscription - Unauthorized
6.3.5.18.9	SOL002 GET Individual Subscription - Not Found
6.3.5.22.9	SOL002 POST Change current VNF Package Task - VNF instance not found
6.3.5.22.10	SOL002 POST Change current VNF Package Task - Unauthorized
6.3.5.23.11	SOL002 POST Create VNF Snapshot Task - Unauthorized
6.3.5.24.10	SOL002 POST Revert to VNF Snapshot Task - Unauthorized
6.3.5.25.17	SOL002 POST Create a new VNF Snapshot - Unauthorized
6.3.5.26.10	SOL002 GET Information about an individual VNF Snapshot - Unauthorized
6.3.5.27.8	SOL002 POST Select VNF Deployable Module - Unauthorized
7.3.1.1.20	SOL003 POST Create a new vnflInstance - Unauthorized
7.3.1.1.21	SOL003 GET information about VNF instances with invalid resource endpoint
7.3.1.2.12	SOL003 Get Information about an individual VNF Instance - Unauthorized
7.3.1.3.10	SOL003 POST Instantiate a vnflInstance - Not Found
7.3.1.4.13	SOL003 Post Scale a vnflInstance - Unauthorized
7.3.1.5.11	SOL003 Post Scale a vnflInstance to level - Unauthorized
7.3.1.6.10	SOL003 POST Change deployment flavour of a vnflInstance - Unauthorized
7.3.1.7.9	SOL003 POST Terminate a vnflInstance Not Found
7.3.1.7.10	SOL003 POST Terminate a vnflInstance - Unauthorized
7.3.1.8.10	SOL003 POST Heal a vnflInstance - Unauthorized
7.3.1.9.10	SOL003 POST Operate a vnflInstance - Unauthorized
7.3.1.10.9	SOL003 POST Change external VNF connectivity - VNF instance not found
7.3.1.10.10	SOL003 POST Change external VNF connectivity - Unauthorized
7.3.1.10.11	SOL003 POST Change external VNF connectivity
7.3.1.11.17	SOL003 GET status information about multiple VNF LCM operation occurrences - Unauthorized
7.3.1.12.9	SOL003 Get Individual VNF LCM Operation occurrences - Unauthorized
7.3.1.13.10	SOL003 POST Retry operation task - Unauthorized
7.3.1.13.21	SOL003 GET VNF Subscription with invalid resource endpoint
7.3.1.14.10	SOL003 POST Rollback operation task - Unauthorized
7.3.1.15.10	SOL003 POST Fail operation task - Unauthorized
7.3.1.16.10	SOL003 Post Cancel operation task - Unauthorized
7.3.1.17.20	SOL003 POST Create a new subscription - Unauthorized
7.3.1.17.21	SOL003 GET VNF Subscription with invalid resource endpoint
7.3.1.18.9	SOL003 GET Individual Subscription - Unauthorized
7.3.1.37.9	SOL003 POST Change current VNF Package Task - VNF instance not found
7.3.1.37.10	SOL003 POST Change current VNF Package Task - Unauthorized
7.3.1.38.11	SOL003 POST Create VNF Snapshot Task - Unauthorized
7.3.1.39.10	SOL003 POST Revert to VNF Snapshot Task - Unauthorized
7.3.1.40.18	SOL003 POST Create a new VNF Snapshot - Unauthorized

Test ID	Test Title
7.3.1.41.12	SOL003 GET VNF State Snapshot - Unauthorized
7.3.1.42.12	SOL003 GET VNF State Snapshot - Unauthorized
7.3.1.43.8	SOL003 POST Select VNF Deployable Module - Unauthorized
8.3.1.10.1	SOL009 POST Change consumed NFV MANO interface state task
8.3.1.10.2	SOL009 POST Change consumed NFV MANO interface state task - Conflict
8.3.1.10.3	SOL009 GET Change consumed NFV MANO interface state task - Method not implemented
8.3.1.10.4	SOL009 PUT Change consumed NFV MANO interface state task - Conflict - Method not implemented
8.3.1.10.5	SOL009 PATCH Change consumed NFV MANO interface state task - Conflict - Method not implemented
8.3.1.10.6	SOL009 DELETE Change consumed NFV MANO interface state task - Conflict - Method not implemented
8.3.1.10.7	SOL009 POST Change consumed NFV MANO interface state task with permitted authorization scope
8.3.1.10.8	SOL009 POST Change consumed NFV MANO interface state task with permitted authorization scope
11.3.1.15.1	Interconnected Create a new subscription and verify notification is generated
11.3.1.16.1	Interconnected Retry a VNF Scale Out operation that fails
11.3.1.17.1	Interconnected Retry a VNF Change External Connectivity operation that fails
11.3.1.18.1	Interconnected Rollback on a VNF Scale Out operation
12.3.1.1.1-17	SOL025 TDs for Analytics
12.3.1.2.1-10	SOL025 TDs for Individual Telco-Cloud data Analytics
12.3.1.3.1-9	SOL025 TDs for Individual Telco-Cloud data Analytics report
12.3.1.4.1-15	SOL025 TDs for Subscriptions
12.3.1.5.1-9	SOL025 TDs for Individual Subscription
12.3.1.6.1	SOL025 Data Analytics Change Notification
12.3.1.7.1-10	SOL025 TDs for NotificationEndpoint
12.3.1.8.1-10	SOL025 TDs for ApiVersion

K.2.2 New TDs that were supposed to be included but missing in the previous versions of the present document

Table K.2.2-1 contains the list of Test Descriptions that were supposed to be included but missing in the previous versions of the present document.

Table K.2.2-1: List of Test Descriptions that were supposed to be included but missing in the previous versions of the present document

Test ID	Test Title
11.3.1.10.1	Scale In a VNF instance. NOTE: GET information about multiple VNF instances (7.3.1.1.2) were missing in the previous version of the present document.

Annex L (normative): Special Remarks on Test Descriptions of this version of the present document

Table L-1 contains the list of special remarks on Test Descriptions of this version of the present document.

Table L-1: List of special remarks on Test Descriptions of the present document

Clause number	Aspects	Notes
4.1.2	Conformance target provisions	This version of present document and Robot code supports "shall"/"shall not" provisions specified in the target NFV-SOL specifications. In addition, "should"/"should not", "may"/"need not", "can"/"cannot" are supported only for JSON schema and data types validation.
4.5.3.1	Expected action test in SUT	This is supported in this version of present document and Robot code in the case of: <ul style="list-style-type: none"> • creation of new resources, and related post condition checks to verify resource has been created; • modifications of existing resources (e.g. through PATCH operations), with partial support; • testing the generation of the expected notification, where provisioned by NFV-SOL specifications; • validating intermediate notifications or states for workflow tests and interconnected tests.
4.5.3.2	Linked resource validation	This test is not supported in this version of present document and Robot code.
4.5.3.3.1	Received notification validation	Steps 6, 7 and 8 are not supported in this version of the present document and Robot code.
4.5.3.3.3	Received notification validation	Steps 8, 9 and 10 are not supported in this version of the present document and Robot code.
4.5.4.1	1-C HTTP header	The validation only for Content-type header for responses with HTTP response body and Location header when specified by the target NFV-SOL specifications are supported.
4.5.4.1	1-D Error reporting	The test system does not support the test for the exhaust error Response Codes specified in ETSI GS NFV-SOL 013 [5] for the "4xx/5xx" part of each Response body filed in the tables for "Details of the request/response on the resources". The tests for the error code 401/404 those are not explicitly specified in each of the Response body part of the tables for "Details of the request/response on the resources" in the baseline NFV-SOL specifications, are supported with only Test ID 6.3.5.11 and tests for VNFLifecycleManagement-API of ETSI GS NFV-SOL 003 [1]. The test for specific error code 412 on ETSI GS NFV-SOL 025 [12] is not supported due to ETSI GS NFV-SOL 025 [12] specification reason.
4.5.4.1	1-E API authentication	ETSI GS NFV-SOL 013 [5], clause 8 specifications are supported with these exceptions: <ul style="list-style-type: none"> • only type OAUTH2_CLIENT_CREDENTIALS supported, • 8.3.4 Authorizing the sending of notifications is not tested.
4.5.4.2	2-F Response message validation	This version of the present document and Robot code supports: <ul style="list-style-type: none"> • JSON schema validation is supported. • Conditional check for the requirements of the data type, i.e. requirements described in the description part of the data type table in the NFV-SOL specifications is not fully performed: <ul style="list-style-type: none"> – Validation on presence/absence of and attribute based on presence/absence of other attributes in the same data type. • HTTP Header presence and syntax is supported. • HTTP body presence and syntax through JSON schema validation is supported.

Annex M (informative): Change history

Date	Version	Information about changes
October 2025	V5.1.2	Initial draft with implementation of contributions, approved during SOL#338: NFVSOL(25)000297_TST010ed521_Target_specification_update approved during SOL#339: NFVSOL(25)000296r1 TST010ed521_Clarification_on_conformance
December 2025	V5.1.3	Draft with implementation of contributions, approved during SOL#345: NFVSOL(25)000371r1 TST010 Reference update approved during SOL#350: NFVSOL(25)000450r3 TST010_support_clarification
January 2026	V5.1.4	Draft with implementation of contributions, approved during SOL#352: NFVSOL(26)000016r2 TST010ed521 Annex J Update approved during SOL#352: NFVSOL(26)000017r1 TST010ed521 Annex K Update
February 2026	V5.1.5	Draft with implementation of contributions, approved during SOL#354: NFVSOL(26)000040 TST010ed521_Finalization approved during SOL#355: NFVSOL(26)000047 TST010ed521_Fix_Annex_I-2_numbering

History

Version	Date	Status
V5.1.1	April 2025	Publication
V5.2.1	April 2026	Publication