

# Recommendation

## **ITU-T Q.4076 (04/2025)**

SERIES Q: Switching and signalling, and associated measurements and tests

Testing specifications – Testing specifications for IMT-2020 and IoT

---

## **Evolution of the testbeds federations reference model**

# ITU-T Q-SERIES RECOMMENDATIONS

## Switching and signalling, and associated measurements and tests

SIGNALLING IN THE INTERNATIONAL MANUAL SERVICE	Q.1-Q.3
INTERNATIONAL AUTOMATIC AND SEMI-AUTOMATIC WORKING	Q.4-Q.59
FUNCTIONS AND INFORMATION FLOWS FOR SERVICES IN THE ISDN	Q.60-Q.99
CLAUSES APPLICABLE TO ITU-T STANDARD SYSTEMS	Q.100-Q.119
SPECIFICATIONS OF SIGNALLING SYSTEMS NO. 4, 5, 6, R1 AND R2	Q.120-Q.499
DIGITAL EXCHANGES	Q.500-Q.599
INTERWORKING OF SIGNALLING SYSTEMS	Q.600-Q.699
SPECIFICATIONS OF SIGNALLING SYSTEM NO. 7	Q.700-Q.799
Q3 INTERFACE	Q.800-Q.849
DIGITAL SUBSCRIBER SIGNALLING SYSTEM NO. 1	Q.850-Q.999
PUBLIC LAND MOBILE NETWORK	Q.1000-Q.1099
INTERWORKING WITH SATELLITE MOBILE SYSTEMS	Q.1100-Q.1199
INTELLIGENT NETWORK	Q.1200-Q.1699
SIGNALLING REQUIREMENTS AND PROTOCOLS FOR IMT-2000	Q.1700-Q.1799
SPECIFICATIONS OF SIGNALLING RELATED TO BEARER INDEPENDENT CALL CONTROL (BICC)	Q.1900-Q.1999
BROADBAND ISDN	Q.2000-Q.2999
SIGNALLING REQUIREMENTS AND PROTOCOLS FOR THE NGN	Q.3000-Q.3709
SIGNALLING REQUIREMENTS AND PROTOCOLS FOR SDN	Q.3710-Q.3899
TESTING SPECIFICATIONS	Q.3900-Q.4099
Testing specifications for next generation networks	Q.3900-Q.3999
Testing specifications for SIP-IMS	Q.4000-Q.4039
Testing specifications for Cloud computing	Q.4040-Q.4059
<b>Testing specifications for IMT-2020 and IoT</b>	<b>Q.4060-Q.4099</b>
PROTOCOLS AND SIGNALLING FOR PEER-TO-PEER COMMUNICATIONS	Q.4100-Q.4139
PROTOCOLS AND SIGNALLING FOR COMPUTING POWER NETWORKS	Q.4140-Q.4159
PROTOCOLS AND SIGNALLING FOR QUANTUM KEY DISTRIBUTION NETWORKS	Q.4160-Q.4179
SIGNALLING REQUIREMENTS AND PROTOCOLS FOR IMT-2020	Q.5000-Q.5049
COMBATING COUNTERFEITING AND STOLEN ICT DEVICES	Q.5050-Q.5069

*For further details, please refer to the list of ITU-T Recommendations.*

# Recommendation ITU-T Q.4076

## Evolution of the testbeds federations reference model

### Summary

Recommendation ITU-T Q.4076 covers the following aspects:

- Application programming interfaces (APIs) invocation framework for testbeds federations reference model defined in Recommendation ITU-T Q.4068;
- Workflow scenario illustration involving a user accessing and using a testbeds federations ecosystem based on the testbeds federation's reference model APIs defined in Recommendation ITU-T Q.4068 and the extensions listed in Recommendation ITU-T Q.4077;
- Security framework that should be applied to the testbeds federations reference model in Recommendation ITU-T Q.4068;
- Instantiations of the testbeds federations reference model, and transformation of existing IMT-2020/5G related testbeds APIs;
- Key performance indicators (KPIs) specific to testbeds federations.

### History \*

Edition	Recommendation	Approval	Study Group	Unique ID
1.0	ITU-T Q.4076	2025-04-13	11	11.1002/1000/16292

### Keywords

5G/IMT-2020, APIs for testbed automation, APIs for testbeds federation, reference model, testbed domain concept, testbeds federations, workflows

---

\* To access the Recommendation, type the URL <https://handle.itu.int/> in the address field of your web browser, followed by the Recommendation's unique ID.

## FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, and information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

## NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

## INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at <https://www.itu.int/ITU-T/ipr/>.

© ITU 2025

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

## Table of Contents

	Page
1 Scope.....	1
2 References.....	1
3 Definitions .....	2
3.1 Terms defined elsewhere .....	2
3.2 Terms defined in this Recommendation.....	2
4 Abbreviations and acronyms .....	2
5 Conventions .....	2
6 Gap Analysis on APIs for testbeds federations .....	2
6.1 Overview .....	2
6.2 Methodologies and tools for specification of APIs .....	3
7 APIs invocation framework.....	3
8 Workflow scenarios illustrations .....	11
8.1 Overview .....	11
8.2 Primary enabling workflow .....	12
8.3 User-oriented workflow scenario illustration.....	15
9 Testbeds federations security framework .....	19
10 Instantiations of the testbeds federations reference model, transformation of existing APIs.....	20
10.1 Instantiations of the testbeds federations reference model.....	20
10.2 Transformation of existing IMT-2020/5G related testbeds APIs.....	22
10.3 Approach to executing proof of concepts (PoCs) on the testbeds federation's reference model.....	22
11 Key Performance Indicators specific to testbeds federations .....	24
11.1 Overview .....	24
12 Further studies .....	25
Annex A – Points/aspects for further studies.....	26
Bibliography.....	28



# Recommendation ITU-T Q.4076

## Evolution of the testbeds federations reference model

### 1 Scope

The scope covers the following items that complement topics covered in [ITU-T Q.4068]:

- "Application programming interfaces (APIs) invocation framework" that clarifies how generic APIs invoke testbed-specific APIs.
- Workflow scenario illustration involving a user accessing and using a testbeds federation ecosystem based on the testbeds federations reference model APIs defined in [ITU-T Q.4068] and the extensions listed in [ITU-T Q.4077].
- Security framework that should be applied to the testbeds federations reference model defined in [ITU-T Q.4068].

A guide on how to perform instantiations of the testbeds federations reference model [ITU-T Q.4068] in creating testbeds that exhibit the capability to federate with other testbeds due to their conformance to the "*testbed domain concept*" prescribed by the reference model defined in [ITU-T Q.4068] and to their implementation of APIs for federation of testbeds. Guiding Illustrations on how instantiations of the testbeds federations reference model [ITU-T Q.4068] should be carried out by the global community. It also serves as a guide to transforming existing testbeds so that they conform to the *testbed domain concept* prescribed by the reference model and to their implementation of APIs for federation of testbeds. It serves as a guide to testbed owners and developers on how to apply the reference model [ITU-T Q.4068] in clean-slate or green field development of testbeds. It also provides testbed developers/implementers with guidance on how transformations or enhancements/evolutions may be pursued by the global community to make existing IMT-2020/5G related testbeds APIs fit or conform to the testbeds federation's reference model [ITU-T Q.4068] and its "APIs invocations framework".

- Important key performance indicators (KPIs) for framing and capturing the performance of federated testbeds as individual testbeds and the testbeds federations themselves. Several KPI categories are covered in relation to federated testbeds, including availability and resilience KPIs, cost/performance related KPIs, and testbed services KPIs. The KPIs, as outlined in this document, capture the essential features and requirements necessary to make federation of testbeds successful from a technical/utility as well as business/commercial point of view. As such, important KPIs for framing and capturing the performance of federated testbeds and testbeds federations themselves are defined in this document. KPIs specific to federation of testbeds help assess the sustainability of testbeds federations ecosystem from the perspective of technical/utility as well as business/commercial point of view.

### 2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[[ITU-T Q.4068](#)]

Recommendation ITU-T Q.4068 (2021), *Open application program interfaces (APIs) for interoperable testbed federations*.

<a href="#">[ITU-T Q.4077]</a>	Recommendation ITU-T Q.4077 (2025), <i>Testbed as a Service application program interfaces descriptions and interoperability requirements</i> .
[IEEE 2302]	IEEE 2302-2021, <i>IEEE Standard for Intercloud Interoperability and Federation (SIIF)</i> .
[IETF RFC 2865]	IETF RFC 2865 (2020), <i>Remote Authentication Dial In User Service (RADIUS)</i> .
[IETF RFC 6733]	IETF RFC 6733 (2012), <i>Diameter Base Protocol</i> .
[IETF RFC 8907]	IETF RFC 8907 (2020), <i>The Terminal Access Controller Access-Control System Plus (TACACS+) Protocol</i> .

### 3 Definitions

#### 3.1 Terms defined elsewhere

None.

#### 3.2 Terms defined in this Recommendation

None.

### 4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AMC	Autonomic Management and Control
API	Application Programming Interface
GANA	Generic Autonomic Networking Architecture
GUI	Graphical User Interface
IoT	Internet of Things
PoC	Proof of Concept
Rfp	Reference Point
SDO	Standards Development Organization

### 5 Conventions

In this Recommendation, the application programming interfaces (APIs) and reference points (Rfps) are 'named' by indexing to distinguish them.

### 6 Gap Analysis on APIs for testbeds federations

#### 6.1 Overview

This clause provides a gap analysis to identify the types of APIs that are missing in the testbeds federation's reference model [ITU-T Q.4068] and provides insights into APIs that should be added to the reference model and further characterizes the APIs. The following are the perspectives in this regard:

- There are some APIs that can be considered for adding to the APIs of the *inter-testbed E2E universal resource broker for testbed federations* defined in [ITU-T Q.4068].



- APIs defined in [IEEE 2302] for intercloud interoperability and federation (SIIF) are applicable in implementing APIs for testbeds Federations that complement the APIs defined in the reference model [ITU-T Q.4068]. As such, [ITU-T Q.4077] has mentioned some APIs from [IEEE 2302] on Institute of Electrical and Electronics Engineers (IEEE) standard for SIIF to add to the APIs defined in [ITU-T Q.4068].
- APIs that use digital twins should be considered for further study, regarding how they complement the APIs defined in testbeds federations reference model defined in [ITU-T Q.4068], or how they can be applied to the reference model to extend its APIs.
- APIs and frameworks for business assurance, service assurance, and security assurance in testbeds federations are a subject for further study.

Annex A lists some points/aspects for further study to see whether they have already been addressed by [ITU-T Q.4077], [b-ITU-T QSTR-TFR] and other deliverables as may be necessary, as this could lead to some improvements to the testbeds federation's reference model defined in [ITU-T Q.4068] as may be necessary.

## 6.2 Methodologies and tools for specification of APIs

The generic APIs defined in testbeds federations reference model [ITU-T Q.4068] should be implemented following RESTful APIs specification approaches. The APIs stage-2 detailed RESTful specifications should be achieved by the global community implementing testbeds that conform to the testbeds federations reference model defined in [ITU-T Q.4068]. [ITU-T Q.4077] provides RESTful specification of the generic APIs of the testbeds federations reference model defined in [ITU-T Q.4068].

## 7 APIs invocation framework

The APIs invocation framework for the for testbeds federation's reference model [ITU-T Q.4068], consists of the following aspects:

- a) Workflows that derive from activities performed by some Actors on the interfaces (e.g., APIs) of the testbeds federations reference model defined in [ITU-T Q.4068]. Clause 8 presents two such workflows meant to illustrate some of the main workflows that shall be encountered in the testbeds federation ecosystem that is based on the reference model.
- b) Generic APIs invoking testbeds specific APIs. This aspect is explained below, based on two approaches that should be considered in implementing the APIs invocation framework. And in addition, Table 1 presents the generic APIs for testbed as a service (TaaS) and complementary comments on some fundamental invocation conditions for the individual APIs.
- c) Entities that constitute the testbed domain concept defined in [ITU-T Q.4068], such as the testbed management system, resource broker, resources, and other entities of testbed domain concept, invoking APIs on one another as described by the APIs defined in [ITU-T Q.4068] and the extensions in [ITU-T Q.4077].

Two approaches should be considered in implementing the APIs invocation framework:

1. **Approach-1:** Implementers of testbeds should inherit and extend the generic APIs by further specializing the APIs according to testbed-type-specific details that are based on *information model(s)* and *data model(s)* supported by the specific testbed type as it implements the "testbed domain concept" defined in the testbeds federation's reference model [ITU-T Q.4068]. The effect is that as a "caller" calls, primarily, the generic APIs on the specialized testbed domain (specialized according to the types of technologies and resources that constitute the testbed) , the "caller" achieves the results expected of the calls made on the generic APIs, while the testbed-specific APIs may include other methods the "caller" may call as well to complement the methods supported by the generic APIs.

NOTE – The implementation approach that may be taken by testbed implementers, of directly inheriting and extending the generic APIs by further specializing the APIs according to testbed type specific details that are based on *information model(s)* and *data model(s)* supported by the specific testbed type, could apply primarily to the case of green field or clean slate testbed implementation. Meaning the case when the testbed and some of its entities that constitute the testbed domain concept.

2. **Approach-2:** If the approach taken by the implementers of the specialized testbed domain concept, specialized by the type of testbed for the specific types of technologies and resources that constitute the testbed, is not to directly inherit, extend and implement the generic APIs of the testbeds federation's reference model [ITU-T Q.4068], then the generic APIs' methods would need to be implemented in such a way that they invoke testbeds-specific APIs that then return results expected by the "caller" via the generic API(s) involved by the "caller" on the target testbed domain entity.

**Table 1 – Generic APIs for testbed as a service (TaaS) and comments on fundamental invocations conditions**

#	API name [ITU-T Q.4077]	API description [ITU-T Q.4077]	Complementary comments on fundamental invocations conditions
1	APIa	The API is used for providing descriptive information about the resource, its state and usage in real-time upon the invocation of the API by the testbed management system.	The API is invoked by the caller when performing an integration and functional test of the target. It may also be invoked during resource booking/reservation for users authorized to use the testbed
2	APIb	The API allows a test manager to interact with a resource (either a component under test (CUT), a system under test (SUT), or a configurable interface. It enables the test manager to configure the resource for a test scenario, and/or pull test results after the test completes.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the execution of certain test scenarios that require the participation of the resource
3	APIc	The API is used by level-0 resources to dynamically provide information in real-time about their state in terms of usage and other information such as performance data and workload being sustained by the resource.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during periodic asynchronous updates on state by the resource
4	APId	The API is used by level-1 resources to dynamically provide information in real-time about their state in terms of usage and other information such as performance data and workload being sustained by the resource.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during periodic asynchronous updates on state by the resource
5	APIe	The API is used by the testbed management system to enable testbed admin to pull and view (visualize) information about the state of resources (level-0 and level-1) from the real-time state repository, especially when the	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked when establishing connectivity between

**Table 1 – Generic APIs for testbed as a service (TaaS) and comments  
on fundamental invocations conditions**

#	API name [ITU-T Q.4077]	API description [ITU-T Q.4077]	Complementary comments on fundamental invocations conditions
		testbed admin intends to assess the state of certain resources before deciding to establish connectivity among resources within the testbed and/or with resources in other testbeds.	testbeds and resource across testbed domains, or during resource booking/reservation for a user authorized to use the testbed
6	APIf	The API is used by the test manager to retrieve information about level-0 and level-1 resources that may be required to participate in a certain test scenario a testbed user may intend to execute. It also allows the test manager to retrieve information about the state of resources (level-0 and level-1) from the real-time state repository, especially when resource state plays a role during the execution of certain test cases or when the test manager may require to use the state of certain resources to establish connectivity among resources in the testbed and/or resources in other testbeds.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.
7	APIg	The API is used by a test manager to execute certain test cases that are meant to test the resource during a scenario in which the resource is a system under test (SUT) or a component under test (CUT). The API is also meant for use by a test manager to request the resource to execute a certain behaviour that is required by a test case(s) being executed by the test manager, including configuring some level-0 resource(s) that can only be configured via the level-1 resource.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.
8	APIh	The API is used by the testbed resource broker to gather information about the capabilities of the resource, its state and its availability to serve testbed service request(s) received from prospective testbed user(s). If the resource is an orchestrator of resources (level-1 and/or level-0), then the same API is also used by the testbed resource broker to request the resource to orchestrate instance(s) of certain level-1 resource(s) and/or level-0 resource(s) as slices that are required to fulfil requirements of a testbed service request received from a prospective testbed user, when there is no existing instance (slice) of the required type of	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.

**Table 1 – Generic APIs for testbed as a service (TaaS) and comments  
on fundamental invocations conditions**

#	API name [ITU-T Q.4077]	API description [ITU-T Q.4077]	Complementary comments on fundamental invocations conditions
		resource(s) that is already available to fulfil the requirements of the newly received request for a prospective testbed user. The same API is also used by the testbed broker to obtain information about capabilities and state of resources directly under the management and control responsibility of the level-1 resource.	
9	APIi	The API is used for providing descriptive information about the resource, its state and usage in real-time upon the invocation of the API by the testbed management system.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.
10	APIj	The API is used for providing updates to descriptive information about the resource, its state and usage in real-time upon the invocation of the API by the level-1 resource, whenever there are changes that have occurred on the resource.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.
11	APIk	The API is used for providing updates to descriptive information about the resource, its state and usage in real-time upon the invocation of the API by the level-0 resource, whenever there are changes that have occurred on the resource.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.
12	APIl (or GUI_l)	The graphical user interface (GUI) is to be used by the broker administrator (broker admin) for performing all necessary broker governance and management activities and operations on the broker. For example, the broker admin installs the policies that govern the operation of the broker in terms of how it registers the testbed domain with the inter-testbed E2E universal resource broker for testbeds federation. It also installs policies that govern the services offered to prospective users of testbeds (including the policies for testbed usage by prospective users). Via the GUI (API), the broker admin can manage the broker to prepare the broker in such a way that the broker can register with the inter-testbed E2E universal resource broker for testbeds federation. In	The API/GUI is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.

**Table 1 – Generic APIs for testbed as a service (TaaS) and comments  
on fundamental invocations conditions**

#	API name [ITU-T Q.4077]	API description [ITU-T Q.4077]	Complementary comments on fundamental invocations conditions
		this case, the broker is ready to provide its services to prospective testbed users.	
13	API <sub>m</sub> (or GUI <sub>m</sub> )	The API is to be used by the testbed administrator (testbed admin) for performing all necessary testbed management activities and operations. Via the GUI (API), the testbed admin can manage the testbed to prepare the testbed in such a way that the testbed can provide testbed services to prospective users and can also participate in testbed federations with other testbeds and the inter-testbed E2E universal resource broker for testbeds federation. The GUI of the testbed management system is used by the testbed administrator in establishing connectivity of the testbed with other testbeds that should be interconnected with this testbed in order to provide federated capabilities and resources to prospective users of federated testbeds.	The API/GUI is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.
14	API <sub>n</sub>	The API is used by the testbed resource broker to register itself with the testbed management system and to provide descriptive state and change of state information in real-time to the testbed management system. The API is also used by the testbed resource broker to obtain descriptive information about all resources and other entities of the testbed domain and their capabilities descriptions (as the various resources and entities not only update the real-time state repository but the testbed management system as well, and the information is kept in sync and consistent between the testbed management system and the real-time state repository).  NOTE – The testbed resource broker may use an API provided by the real-time state repository for directly pulling out information about resources available in the testbed domain and their capabilities descriptions.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.

**Table 1 – Generic APIs for testbed as a service (TaaS) and comments  
on fundamental invocations conditions**

#	API name [ITU-T Q.4077]	API description [ITU-T Q.4077]	Complementary comments on fundamental invocations conditions
15	APIo	The API is used by a test manager to register itself with the testbed management system, provide descriptive state and change of state information in real-time to the testbed management system, because a test manager could be considered as a resource itself.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.
16	APIp	The API is used for providing descriptive information about a test manager, its state and usage in real-time upon the invocation of the API by the testbed management system.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.
17	APIq	The API is used for providing test results to a test manager(s) that involve the resource in a test case as a component under test (CUT) or system under test (SUT). The same API is also used for communicating to a test manager some feedback (e.g., errors or failures during the execution) from some invocations triggered earlier on the resource by the test manager.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.
18	APIr	The API is to be used by a test suite/case designer and test executer (upon the acceptance of its request for testbed service by the testbed resource broker) to connect to the test manager instance assigned to the testbed user to use the test manager to design, compile and run test cases, or to upload and compile some test cases designed offline and execute them. Through the API, the testbed user is able to upload some test cases or test suites if the testbed domain allows that and then compile and/or execute the test cases, or the user is only allowed to design, compile and execute test cases directly on the test manager without uploading test cases/suites from outside.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.

**Table 1 – Generic APIs for testbed as a service (TaaS) and comments  
on fundamental invocations conditions**

#	API name [ITU-T Q.4077]	API description [ITU-T Q.4077]	Complementary comments on fundamental invocations conditions
19	APIs	The API is used by the inter-testbed E2E universal resource broker for testbeds federation to connect to the test manager, e.g., to enable the E2E resource broker to access state information about the specific test manager, or for cases whereby some test results could be shared to the testbed user via the E2E resource broker if not possible that the test manager provides direct access to those kinds of test results directly to the user (test executor), though primarily the test executor should be able to access test results directly from the test manager(s). The API is using the same uniform resource description model that APIx uses.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.
20	APIt	The API is used for providing test results to a test manager(s) that involve the resource in a test case as a component under test (CUT). The same API is also used for communicating to a test manager some feedback (e.g., errors or failures during the execution) to some invocations triggered earlier on the resource by the test manager.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.
21	APIu	The API is used by the testbed management system to keep synchronizing with the testbed resource broker on the state of the broker. The same API is also used by the testbed management system to provide updates on any changes in the descriptive information about all resources and other entities of the testbed domain and their capabilities descriptions (information that is kept in sync and consistent between the testbed management system and the real-time state repository).	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.
22	APIv	The API is used by a level-1 resource to push updated information (updates) about the state of resources under the management and control responsibility of the level-1 resource and their capabilities descriptions, and any changes that may have occurred to the resources and capabilities. The same API is also used for synchronizations between the level-1 resource and the testbed resource broker.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.

**Table 1 – Generic APIs for testbed as a service (TaaS) and comments  
on fundamental invocations conditions**

#	API name [ITU-T Q.4077]	API description [ITU-T Q.4077]	Complementary comments on fundamental invocations conditions
23	APIw	The API is used by the inter-testbed E2E universal resource broker for testbeds federation, after the testbed resource broker has registered itself with it via APIx, to then obtain (pull) descriptive information about all testbed resources available in the testbed domain to serve testbed services requests that may come from the E2E resource broker and their capabilities descriptions. The same API is also used by the E2E resource broker to provide synchronization related descriptive state and change of state information in real-time to the test broker. APIw is using the same uniform resource description model that APIx uses.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.
24	APIx	The API is used by the testbed resource broker to push updated information (updates) about the state of resources of the testbed domain and their capabilities descriptions, and any changes that may have occurred to the resources and capabilities. The same API is also used, complementarily to APIw, for synchronizations between the testbed resource broker and the inter-testbed E2E universal resource broker for testbeds federation. Complementarily to APIs, APIx is used to synchronize information between the test manager and the inter-testbed E2E universal resource broker for testbeds federation. APIx is using the same uniform resource description model that APIw uses.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.
25	APIy (or GUI_y)	The API is to be used by the broker administrator (broker admin) for performing all necessary broker governance and management activities and operations on the broker. For example, the broker admin installs the policies that govern the operation of the broker in terms of admitting (or not admitting) testbed domains in their attempts to discover and register with the broker, as well as policies that govern the services offered to prospective users of testbeds registered with the broker (including the policies for testbeds user	The API/GUI is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.



**Table 1 – Generic APIs for testbed as a service (TaaS) and comments on fundamental invocations conditions**

#	API name [ITU-T Q.4077]	API description [ITU-T Q.4077]	Complementary comments on fundamental invocations conditions
		registrations). Via the GUI (API), the broker admin can manage the broker to prepare the broker in such a way that the broker can expose the APIz and any GUIs of the broker that can be made available to prospective testbeds users, such that the broker is ready to provide its services to prospective testbed users and to testbeds intending to register with it.	
26	APIz	This API provides the entry point into the system of federated testbeds to prospective users (testbed users) of the system of federated testbeds. It provides 'search and query and find services' that enable the prospective user of testbed service(s), i.e., the test suite/cases designer and test executer to find/discover testbeds that are available to accept new requests within the time of interest to the prospective testbed user as well as their capabilities and topology information pertaining to their interconnection and federations with other testbeds. Then, a prospective testbed user (test suite/cases designer and test executer) can query the broker for testbeds that fulfil certain capabilities and requirements such as end-to-end latency within the scope of the single testbed or across multiple testbeds, before the prospective user can then select testbeds and launch requests for testbed services. And then, the prospective user can contact the APIr of the different testbed domains to create new experiments, new test cases and test suites.	The API is invoked by the caller when performing an integration and functional test of the target. It is also invoked during the use case scenarios described in the API description.

## 8 Workflow scenarios illustrations

### 8.1 Overview

This clause provides illustrations of workflow scenarios. The main focus is a workflow scenario illustration involving a user of the testbed federations ecosystem accessing and using a testbeds federations ecosystem based on the testbeds federations reference model APIs defined in [ITU-T Q.4068] and the extensions defined in [ITU-T Q.4077].

## 8.2 Primary enabling workflow

The primary workflow, which can be found in Table 2, concerns the preparation of individual testbed domains meant to participate in a federation to be ready to serve users in discovering the testbeds and using them in a federation scenario.

Assumption: There are already instances of testbeds domains that conform to the testbeds federations reference model defined in [ITU-T Q.4068] that are already deployed and operational.

**Table 2 – The primary enabling workflow**

Actor	Step of activity execution	API(s) or GUI or reference points (Rfps) involved (identification tags)	Description
Testbed admin	<b>Step-1:</b> Ensures that all entities in a testbed domain are interconnected as required by the integration APIs, and are operational to deliver the functionality expected of the individual entities	APIm (or GUI_m)	The API is to be used by the testbed administrator (testbed admin) for performing all necessary testbed management activities and operations. Via the GUI (API), the testbed admin can manage the testbed to prepare the testbed in such a way that the testbed can provide testbed services to prospective users and can also participate in testbed federations with other testbeds and the inter-testbed E2E universal resource broker for testbeds federation. The GUI of the testbed management system is used by the testbed administrator in establishing connectivity of the testbed with other testbeds that should be interconnected with this testbed in order to provide federated capabilities and resources to prospective users of federated testbeds.
Testbed admin	<b>Step-2:</b> Collaboratively work with testbed admins of other testbed domains to establish connectivity between the testbed domain and other testbed domains. This is determined by policies and SLAs allowing the testbeds to be interconnected and to participate in federation scenarios from which testbeds users can benefit from,	RfpT, RfpU, RfpS, RfpR	

**Table 2 – The primary enabling workflow**

Actor	Step of activity execution	API(s) or GUI or reference points (Rfps) involved (identification tags)	Description
	e.g., for benefits like end-to-end networking for E2E testing scenarios. The testbed admins work together to establish the testbed-to-testbed interconnections on the Rfps (RfpT, RfpU, RfpS, RfpR,) in order, using necessary means like VLANs, VPNs, physical connectivity, etc.		
Testbed admin	<b>Step-3:</b> Collaboratively work with testbed admins of other testbed domains that have been interconnected with this testbed domain in <b>Step-2</b> , and with the <b>broker admin</b> of the <b>inter-testbed E2E universal resource broker for testbeds federation</b> to make the interconnected Testbeds get connected to the <b>inter-testbed E2E universal resource broker for testbeds federation</b> .	RfpQ, APIw, APIx	
Testbed admin	<b>Step-4:</b> Communicate in some way or makes the <b>broker admin</b> of the testbed domain aware that the testbed domain is ready to participate in testbeds federations and testbed services offering to prospective users		
<b>Broker admin</b> of the Testbed domain	<b>Step-5:</b> Collaboratively work with the <b>broker admin</b> of the <b>inter-testbed E2E universal resource broker for testbeds federation</b> , to get the testbed domain registered and policed	APII (or GUI_I) and APIw	The GUI is to be used by the broker administrator (broker admin) for performing all necessary broker governance and management activities and operations on the broker. For example, the broker admin installs the policies that govern

**Table 2 – The primary enabling workflow**

Actor	Step of activity execution	API(s) or GUI or reference points (Rfps) involved (identification tags)	Description
	<p>(by configuration of policies, according to SLAs among the parties) such that the testbed domain becomes visible and ready to participate in federations and be discovered by prospective users through the <b>inter-testbed E2E universal resource broker for testbeds federation</b>.</p> <p>If desirable, collaboratively work with the <b>broker admin</b> of the <b>inter E2E brokers global universal resource broker for testbeds federation</b>, to get the testbed domain registered and policed (by configuration of policies, according to SLAs among the parties) such that the testbed domain becomes visible even on higher scope of visibility and ready to participate in federations and be discovered by prospective users through the <b>inter E2E brokers global universal resource broker for testbeds federation</b>.</p>		<p>the operation of the broker in terms of how it registers the testbed domain to the inter-testbed E2E universal resource broker for testbeds federation. It also installs policies that govern the services offered to prospective users of testbeds (including the policies for testbed usage by prospective users). Via the GUI (API), the broker admin can manage the broker to prepare the broker in such a way that the broker can register with the inter-testbed E2E universal resource broker for testbeds federation. In this case, the broker is ready to provide its services to prospective testbed users.</p>
<b>Broker admin</b> for the " <b>inter-testbed E2E universal resource broker for testbeds federation</b> "	<b>Step-6:</b> Perform the activities that ensure that interconnected testbeds in step-2 are registered with the broker and are ready for	APIy (or GUI_y)	The API is to be used by the broker administrator (broker admin) for performing all necessary broker governance and management activities and operations on the broker. For example, the broker admin

**Table 2 – The primary enabling workflow**

Actor	Step of activity execution	API(s) or GUI or reference points (Rfps) involved (identification tags)	Description
	users to discover them and seek to use them		installs the policies that govern the operation of the broker in terms of admitting (or not admitting) testbed domains in their attempts to discover and register with the broker, as well as policies that govern the services offered to prospective users of testbeds registered with broker (including the policies for testbeds user registrations). Via the GUI (API), the broker admin can manage the broker to prepare the broker in such a way that the broker can expose the APIz and any GUIs of the broker that can be made available to prospective testbeds users, such that the broker is ready to provide its services to prospective testbed users and to testbeds intending to register with it.

### 8.3 User-oriented workflow scenario illustration

The clause describes the user-oriented workflow scenario illustration (Table 3) on how the complete chain of interactions works with a user of testbeds federation from start to finish.

**Table 3 – User-oriented workflow scenario illustration**

Actor	Step of activity execution	API(s) or GUI involved (identification tags)	Description
<b>Test suite/cases designer and test executer</b>	<b>Step-1:</b> Using an appropriate tool, connect to the APIz of the <b>inter-testbed E2E universal resource broker for testbeds federation</b> . Then execute 'search and query and find services' to find/discover testbeds that are available to accept new requests within the time of interest to the prospective testbed user as well as their	APIz	This API provides the entry point into the system of federated testbeds to prospective users (testbed users) of the system of federated testbeds. It provides 'search and query and find services' that enable the prospective user of testbed service(s), i.e., the test suite/cases designer and test executer to find/discover testbeds that are available to

**Table 3 – User-oriented workflow scenario illustration**

Actor	Step of activity execution	API(s) or GUI involved (identification tags)	Description
	capabilities, topology information pertaining to their interconnection and federations with other testbeds, and also discover other information such as the cost of using a particular testbed or whole federated testbeds, and some KPIs related to testbeds and their federations.		accept new requests within the time of interest to the prospective testbed user as well as their capabilities topology information pertaining to their interconnection and federations with other testbeds. A prospective testbed user (test suite/cases designer and test executor) can query the broker for testbeds that fulfil certain capabilities and requirements such as end-to-end latency within the scope of the single testbed or across multiple testbeds, before the prospective user can then select testbeds and launch requests for testbed services. Then, the prospective user can contact the APIr of the different testbed domains to create new experiments, new test cases and test suites.
<b>Test suite/cases designer and test executor</b>	<b>Step-2:</b> Select testbed or testbeds of interest to launch requests for testbed services, and request (via APIz) for user accounts and profiles for this user (e.g., AAA user accounts and profiles) to be created on the targeted testbed domains by the <i>testbed resource brokers</i> of the targeted testbeds domains (if no user account and profile already exist), after having paid for the testbed as a service (TaaS) via the APIz of the <b>inter-testbed E2E universal resource broker for testbeds federation</b> .	APIz	(as described above)

**Table 3 – User-oriented workflow scenario illustration**

<b>Actor</b>	<b>Step of activity execution</b>	<b>API(s) or GUI involved (identification tags)</b>	<b>Description</b>
<b>inter-testbed E2E universal resource broker for testbeds federation</b>	<b>Step-3:</b> Relays the request to the targeted <i>testbed resource broker(s)</i> via APIw	APIw	The API is used by the inter-testbed E2E universal resource broker for testbeds federation, after the testbed resource broker has registered itself with it via APIx, to then obtain (pull) descriptive information about all testbed resources available in the testbed domain to serve testbed services requests that may come from the E2E resource broker and their capabilities descriptions. The same API is also used by the E2E resource broker to provide synchronization related descriptive state and change of state information in real-time to the test broker. APIw uses the same uniform resource description model that APIx uses.
<b>Testbed resource broker(s) of targeted testbed(s) domain(s)</b>	<b>Step-4:</b> Perform admission control on the request and the user launching the request, and sends the results to the user via APIx of <b>inter-testbed E2E universal resource broker for testbeds federation</b>	APIx	The API is used by the testbed resource broker to push updated information (updates) about the state of resources of the testbed domain and their capabilities descriptions, and any changes that may have occurred to the resources and capabilities. The same API is also used, complementarily to APIw, for synchronizations between the testbed resource broker and the inter-testbed E2E universal resource broker for testbeds federation. Complementarily to APIs, APIx is used to synchronize information between the test manager and the inter-testbed E2E universal resource broker for testbeds federation. APIx uses the same uniform resource description model that APIw uses.

**Table 3 – User-oriented workflow scenario illustration**

Actor	Step of activity execution	API(s) or GUI involved (identification tags)	Description
<b>Test suite/cases designer and test executer</b>	<b>Step-5:</b> Launch a detailed test session/service request (via APIz) for the intended tests to be executed by the user, the time at which tests would be conducted, test manager(s) required, test executions duration and the resources required in the particular targeted testbed domain, to the individual <i>testbed resource brokers</i> of the targeted testbeds domains.	APIz	(as described above)
<b>Testbed resource broker(s) of targeted testbed(s) domain(s)</b>	<b>Step-6:</b> Perform admission control on the detailed test session/service request and the user launching the request, perform the booking of the required testbed resources at the intended time of test session start-time, provide an access ID and means that the user would need to use to contact and access the test manager(s) required directly at the point the test session/service is configured to start, and sends the respond to the user via APIx of <b>inter-testbed E2E universal resource broker for testbeds federation</b>	APIx	(as described above)
<b>Test suite/cases designer and test executer</b>	<b>Step-7:</b> If test session/service request(s) launched to specific targeted <b>testbed resource broker(s) of targeted testbed(s) domain(s)</b> was/were accepted in whole across the multiple testbed domains (in the case that the user intends to use multiple testbeds individually or as federation), then access the	APIr	The API is to be used by a test suite/cases designer and test executor (upon the acceptance of its request for testbed service by the testbed resource broker) to connect to the test manager instance assigned to the testbed user to use the test manager to design, compile and run test cases, or to upload and compile some test cases designed offline and execute them. Through the API, the



**Table 3 – User-oriented workflow scenario illustration**

Actor	Step of activity execution	API(s) or GUI involved (identification tags)	Description
	test manager(s) in the testbed domain(s) directly at the point when the admitted test session(s) has been configured by the testbed broker(s), to start working on test cases designs and/or executions		testbed user is able to upload some test cases or test suites if the testbed domain allows that and then compile and/or execute the test cases, or the user is only allowed to design, compile and execute test cases directly on the test manager without uploading test cases/suites from outside.
<b>Test suite/cases designer and test executer</b>	<b>Step-8:</b> Obtain and export to own space (if allowed by policies or SLAs), the results of the tests from the test manager(s) before test session duration expires (or request a session duration extension via the APIz)	APIr	(as described above)

NOTE – There may be two kinds of possible scenarios regarding testbeds federations:

- a) Testbed admins triggered federations involving the preparation of individual testbed domains meant to participate in a federation to be ready to serve users in discovering the testbeds and using them in a federation scenario – i.e., this happens before users start discovering testbeds and federations. This is the primary scenario.
- b) Secondary scenario for further study: user triggered federation scenario by which user requests issued to the *inter-testbed E2E universal resource broker for testbed federations* could be the ones that then trigger the federation processes to be initiated by the testbed domains.

## 9 Testbeds federations security framework

The testbeds federations reference model defined in [ITU-T Q.4068] does not include a "security framework" that should be applied in the resultant ecosystem of testbeds federations deployments and usage by users based on the reference model. This clause contains considerations of a security framework that can be applied for the testbeds federations reference model defined in [ITU-T Q.4068].

NOTE 1 – Other proposals or further elaborations on the security framework put forward in this clause should help address the question of security in testbeds federations.

The proposed security framework is characterized as consisting of the following aspects:

1. Use of authentication, authorisation and accounting (AAA) protocols and architectures and infrastructures for each testbed domain and the whole testbeds federations architecture framework and ecosystem up to the *inter-testbed E2E universal resource broker for testbed federations* and further up to the *inter E2E brokers global universal resource broker for testbeds federation*. AAA frameworks and protocols are standardized by IETF and there are various IETF RFCs on AAA architecture and protocols. AAA protocols include TACACS+ [IETF RFC 8907], RADIUS [IETF RFC 2865], DIAMETER [IETF RFC 6733], etc. There

are many cases where AAA frameworks and protocols have been applied to testbeds and multiuser access and use of testbeds.

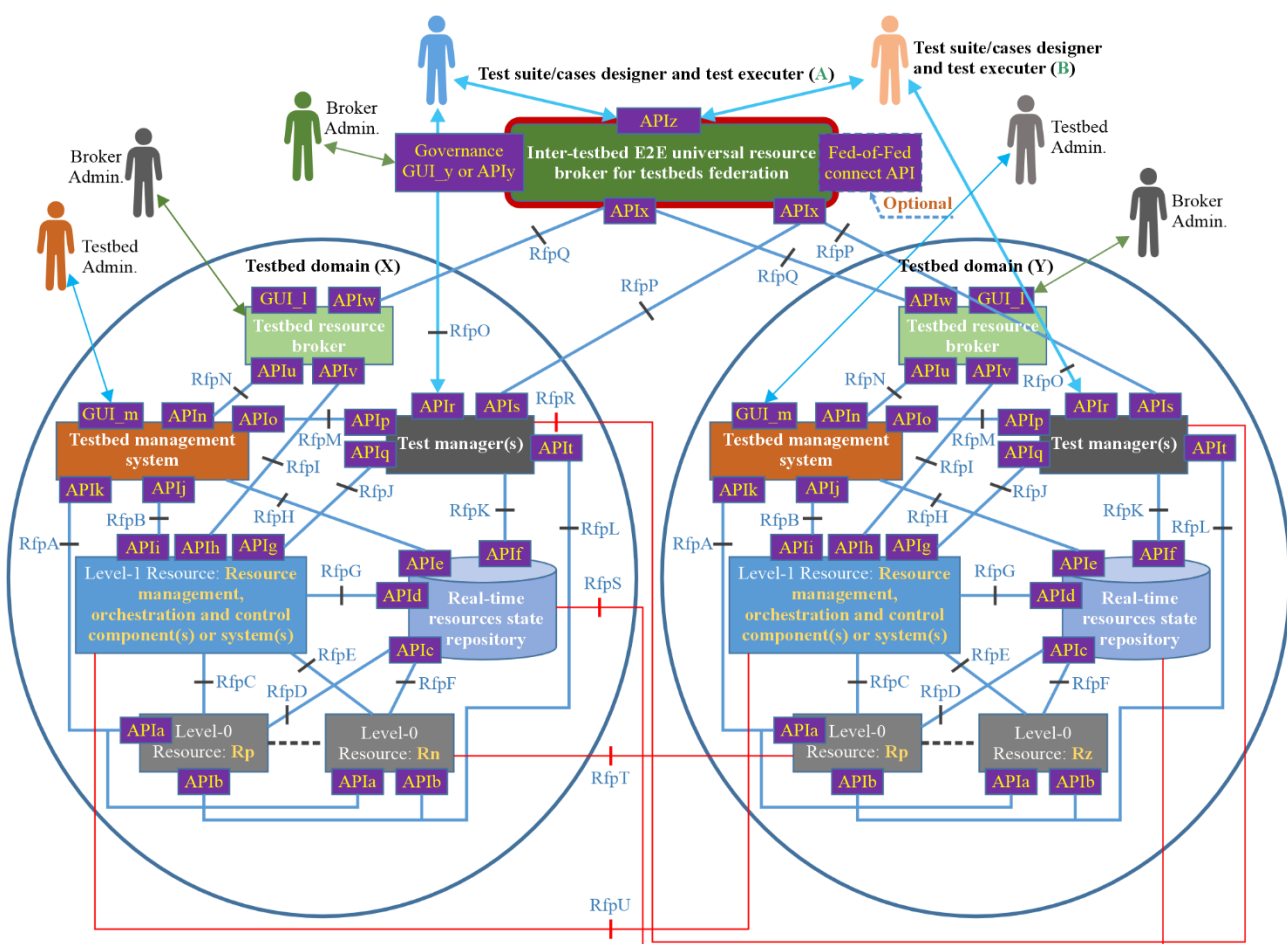
2. AAA framework and infrastructure should be provisioned for each individual testbed domain to cover *user profiles for actors* who access individual entities of the testbed domain, e.g., *level-0 and level-1 resources, testbed resources broker, test manager(s), testbed management system, etc.*
3. AAA framework and infrastructure on a higher level for *actors* on a higher level outside the scope of an individual *testbed domain*, at the *inter-testbed E2E universal resource broker for testbed federations*. This is because AAA frameworks and protocols support the hierarchical deployments of AAA services. Federated AAA services are also possible to deploy to cover distributed sites and user interfaces.
4. AAA framework and infrastructure on a higher level for *actors* on a higher level outside the scope of an individual *inter-testbed E2E universal resource broker for testbed federations*, at the *inter E2E brokers global universal resource broker for testbeds federation*. This is also because AAA frameworks and protocols support the hierarchical deployments of AAA services. Federated AAA services are also possible to deploy to cover distributed sites and user interfaces.
5. Use of encryption protocols on all communications among entities in the ecosystem.
6. Firewalls and security zoning techniques should be applied within and outside the testbed domain on interfaces where firewalls and security zoning should be applied within the whole ecosystem.
7. Zero trust principles (ZTP) and framework (s) should be considered where possible to apply them for certain benefits.

NOTE 2 – Dynamic management of user accounts and profiles concerning users of testbeds, in terms of creation and deletion, needs to take into account the fact that the users come and go after using testbeds federations.

## **10      Instantiations of the testbeds federations reference model, transformation of existing APIs**

### **10.1    Instantiations of the testbeds federations reference model**

The procedures to be followed when performing instantiations of the testbeds federations reference model defined in [ITU-T Q.4068] are as described below, based on the testbeds federations reference model (see Figure 1 of [ITU-T Q.4068]):



**Key:** There may be established horizontal and/or hierarchical/vertical federations of assets/components or resources at specific levels across federated testbeds

Q.4076(25)

**Figure 1 – Generic federated testbed model [ITU-T Q.4068]**

Suggested procedures on performing instantiations of the reference model are as follows:

1. Start with implementing the testbed domain concept using a bottom-up approach of implementing level-0 and level-1 resources and manually testing them in terms of their functionality and integration.
2. Deploy all the other entities (testbed management system, real-time resources state repository, test managers, and test resource broker) of the testbed domain concept being instantiated as a concrete specific testbed type that is specialized according to the types of technologies and resources that constitute the testbed. Also, activate and test their interfaces that can be used by human actors.
3. Implementing the generic APIs of the entities of the testbed domain testbed while taking into consideration testbed type specific details that are based on *information model(s)* and *data model(s)* supported by the specific testbed type as specialized according to the types of technologies and resources that constitute the testbed.
4. Implement the "APIs invocations framework" prescribed in this document.
5. Integrate the level-0 and level-1 resources with the testbed management system.
6. Integrate all the testbed domain reference points (Rfps) and APIs that implement them, before implementing Rfps and APIs for integrating with the *inter-testbed E2E universal resource broker for testbed federations*.
7. Interconnect the testbed domain with another testbed domain(s) and testing the connectivity among testbed domains, and testing all the Rfps concerning the connectivity.

8. Activate the federation behaviours of the testbed domain and test the federation behaviours as it is integrated with the *inter-testbed E2E universal resource broker for testbed federations*.
9. Consider further integrations and testing necessary to make testbeds domains visible at the level of the *inter E2E brokers global universal resource broker for testbeds federation*.

NOTE – These procedures above form the basic guiding illustrations on how instantiations of the testbeds federation's reference model [ITU-T Q.4068] should be carried out by the global community. This serves as a guide on how to perform instantiations of the testbeds federation's reference model [ITU-T Q.4068] in creating testbeds that exhibit the capability to federate with other testbeds, thanks to their conformance to the testbed domain concept prescribed by the reference model and to their implementation of APIs for federation of testbeds.

## 10.2 Transformation of existing IMT-2020/5G related testbeds APIs

The procedures to be followed when performing transformation of existing IMT-2020/5G related testbeds APIs based on the testbeds federations reference model defined in [ITU-T Q.4068] are as described below, based on Figure 1.

Suggested procedures on performing transformation of existing IMT-2020/5G related testbeds APIs based on the reference model and its generic APIs are as follows:

1. Adding/deploying entities of the testbed domain concept that may be missing in the existing testbed to make the testbed conform to the testbed domain concept and its related APIs.
2. Implementing and executing similar procedures as prescribed on performing instantiations of the reference model.

NOTE – These procedures above form a guide on how transformations or enhancements/evolutions may be pursued by the global community to make existing IMT-2020/5G related testbeds APIs fit/conform to the reference model and its APIs invocations framework. This serves as a guide to transforming existing testbeds so that they conform to the testbed domain concept prescribed by the reference model and as guide to their implementation of APIs for federation of testbeds.

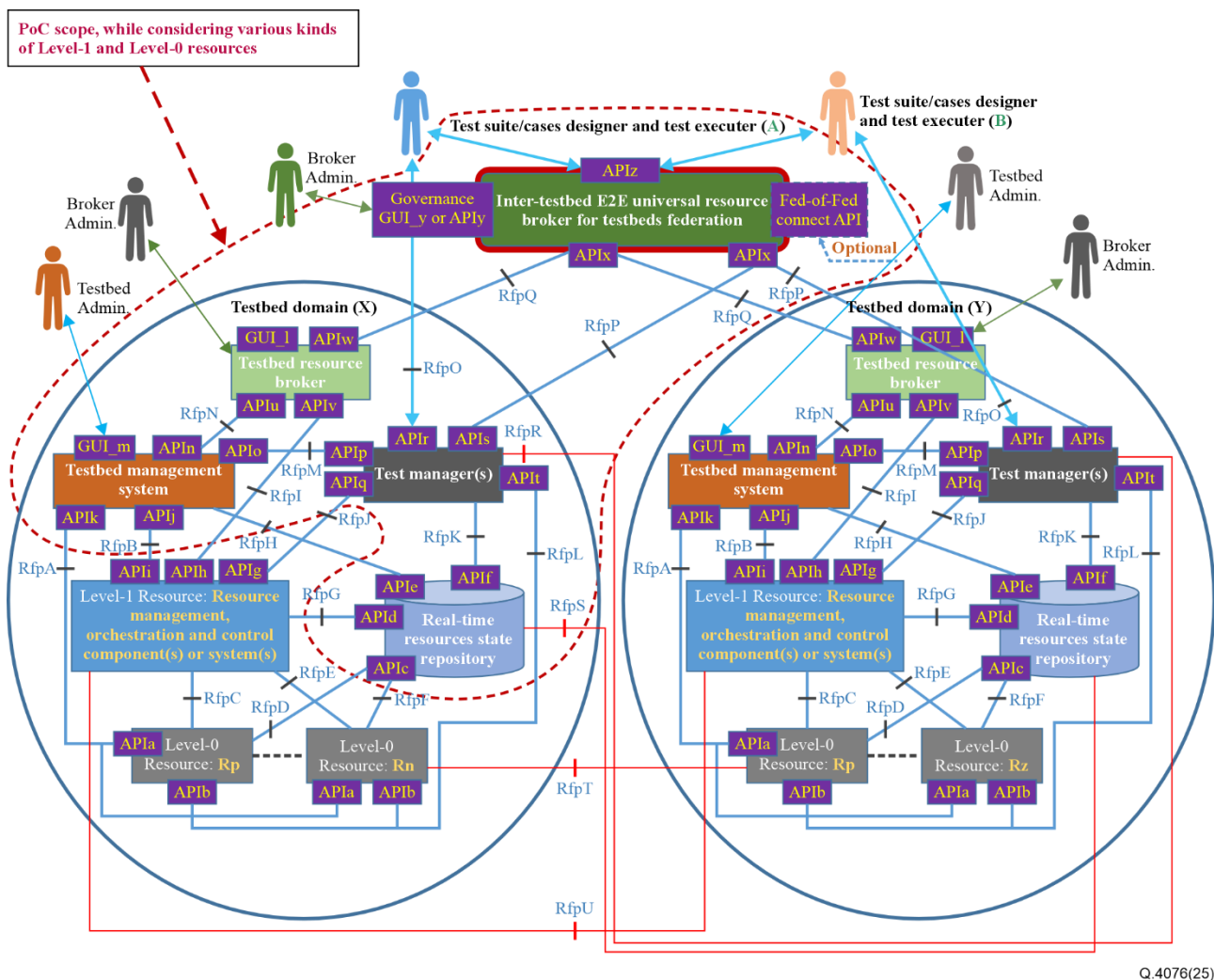
## 10.3 Approach to executing proof of concepts (PoCs) on the testbeds federation's reference model

Time is ripe for the global community to pursue programs/projects on executing proof-of-concepts (PoCs) on the reference model for testbeds federations [ITU-T Q.4068] and on use cases defined in [b-ITU-T QSTR-UCFTBS]. For example, an insight into how to a scope for a PoC on the use case #7 (UC07) listed in [b-ITU-T QSTR-UCFTBS] could be structured as illustrated in Figure 3.

NOTE – Figure 2 and Figure 3 are based on the diagram defined in [ITU-T Q.4068] (Figure 1 of [ITU-T Q.4068]) with some complementary information relevant to PoC. As defined in [ITU-T Q.4068], there may be established horizontal and/or hierarchical/vertical federations of assets/components or resources at specific levels across federated testbeds.

In order to start making use of the reference model for testbeds federations defined in [ITU-T Q.4068], the global community (industry, standards development organizations (SDOs) / Fora, and research communities) should consider launching of the proof-of-concept (PoC) initiatives. The results from such PoCs would be useful for the further study and/or for the global ecosystem on testbeds federations for IMT-2020 and beyond. Such PoCs can also serve as good platforms or instruments for offering trainings to communities on the use of the testbeds federation's reference model [ITU-T Q.4068].

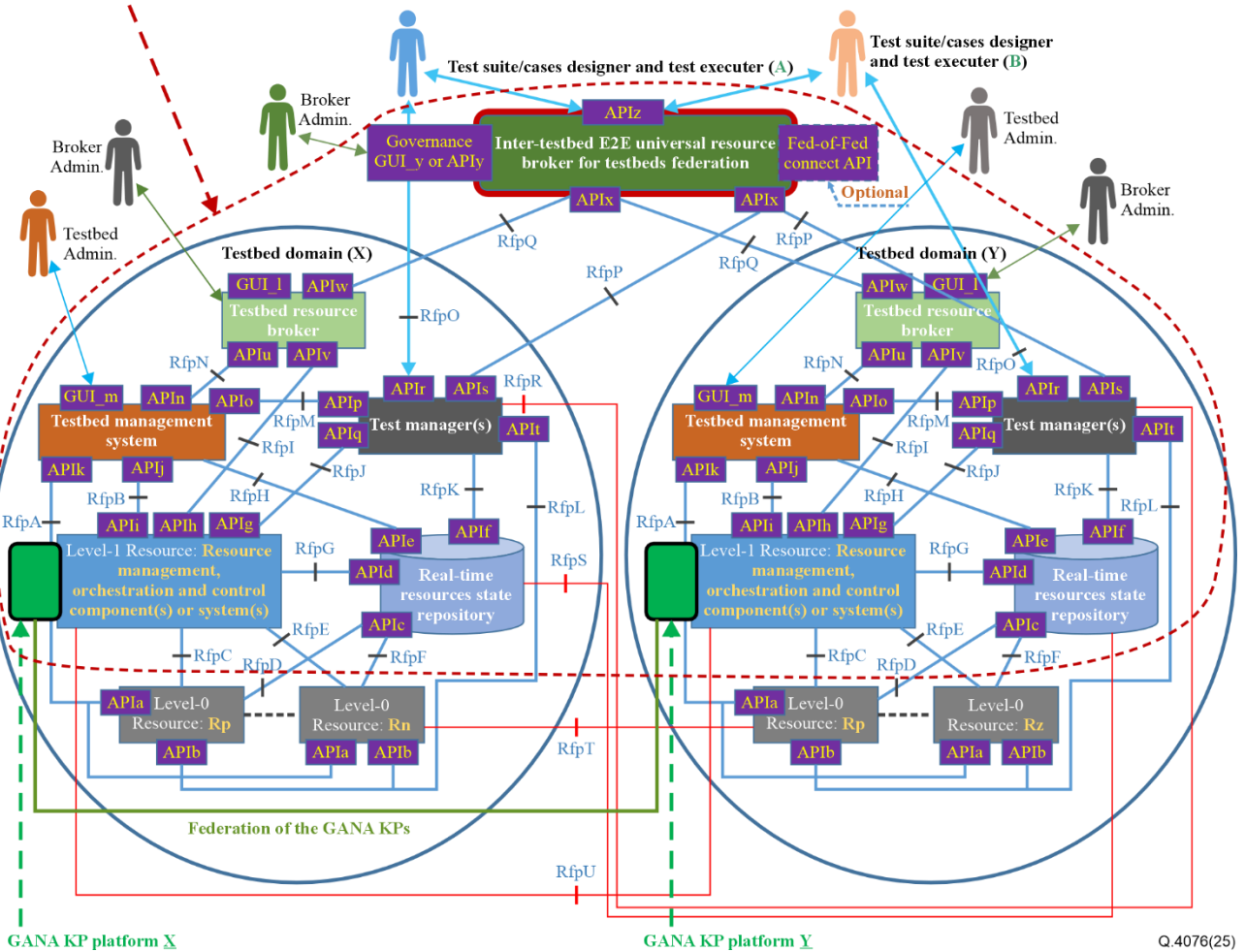
Figure 2 presents insights of the scope of a PoC that can provide valuable results to the ecosystem on testbeds federations.



**Figure 2 – General scope for use in defining PoC projects on testbeds federations**

Figure 3 illustrates the scope and approach for running a PoC on the use case #07 (UC07) listed in [b-ITU-T QSTR-UCFTBS], on use of testbeds federations for testing federated autonomous management and control (AMC) by the federated European Telecommunications Standards Institute (ETSI) generic autonomous networking architecture (GANA) knowledge planes (KPs) platforms for autonomous/autonomous 5G and beyond networks.

PoC scope, while considering various kinds of Level-1 and Level-0 resources



**Figure 3 – Testing federated autonomic management and control (AMC) by federated ETSI GANA knowledge planes (KPs) platforms for autonomic/autonomous 5G and beyond networks**

## 11 Key Performance Indicators specific to testbeds federations

### 11.1 Overview

There are various types of key performance indicators (KPIs) that are specific to testbeds federations conforming to the testbeds federation's reference model defined in [ITU-T Q.4068]. Relevant KPIs include the following:

- Availability of a testbed and/or a testbeds federation;
- Cost model of providing a Testbed as a Service;
- Usability of a testbeds federation;
- Resilience of a testbeds federation;
- Performance of federated testbeds;
- Testbed as a Services specific KPIs.

NOTE – KPIs can be used in benchmarking testbed as a service (TaaS) and should also be used in policing of the admission control operations performed by testbed resource broker of a testbed domain to define and use the policies for admission control of user requests coming from testbed users.

## **12 Further studies**

The roadmaps on trends and requirements for standards for testbeds federations continue to be studied and documented by the industry and research communities. That means such roadmaps on trends and requirements for standards for testbeds federations should be further studied in order to create new standardization work items addressing the emerging roadmaps and needs.

## Annex A

### Points/aspects for further studies

(This annex forms an integral part of this Recommendation.)

This annex lists some points/aspects for further studies in order to see whether they have already been addressed by [ITU-T Q.4077], [b-ITU-T QSTR-TFR] and other deliverables as may be necessary, as this could lead to some improvements on the testbeds federations reference model defined in [ITU-T Q.4068] as may be necessary.

The aspects for further studies and potentially further discussions as may be necessary:

1. Consider and checking if there may be missing interfaces/APIs in the reference architecture (testbeds federations reference model defined in [ITU-T Q.4068]) relevant for federation:
  - a. There is a question of the mentioning APIx two times in the architecture: for communication with the testbed resource broker and test manager (s) is fine or not. The description does not seem to address the communication with the test manager(s).
  - b. Reference points RfpS (communication between *Real-time resources state repository* of two testbed domains) and RfpR (communication between test manager(s) of different testbed domains) do not have an API. There is a question of whether it might be important for federation.
2. There is a question of where it is missing the description of functionality of APIr.
  - a. [ITU-T Q.4068] defines APIr as follows: "*The API is meant for use by a test suite/cases designer and test executer (upon the acceptance of its request for testbed service by the testbed resource broker) to connect to the test manager instance assigned to the testbed user to use the test manager to design, compile and run test cases, or to upload and compile some Test Cases designed offline and execute them. Through the API, the testbed user is able to upload some test cases or test suites if the testbed domain allows that and then compile and/or execute the test cases, or the user is only allowed to design, compile and execute test cases directly on the test manager without uploading test cases/suites from outside.*"  
It does not mention how to get a test manager instance assigned or created. There is a question of whether it is sufficient that APIw is only defined as for getting resource descriptions.
  - b. There is a need to confirm whether the APIr as defined in [ITU-T Q.4068] is aligned with the way it is defined in [ITU-T Q.4077].
3. The reference architecture defined in [ITU-T Q.4068] has no APIs/interfaces listed for identity providers/authentication/authorisation modules, while clauses 9 to 13 of [ITU-T Q.4068] mention these aspects.  
NOTE 1 – A proposed security framework for the testbeds federations reference model defined in [ITU-T Q.4068] is defined in this document, and so further study should consider that the security framework should evolve to address various security and even trust related aspects in testbeds federations.
4. On the interfaces relevant for federation: APIr, APIs, APIw, APIx, APIy/GUI\_y, APIz:
  - a. APIr: seems an important interface that needs more exact definition and API.  
NOTE 2 – Deliverable D2.2 may already have addressed this point.
  - b. APIs (and alternative for APIx): need to have list of functionality/requirements and API description (and there is a question of whether another interface than APIx on the *Inter-Testbed E2E Universal Resource Broker for Testbed Federations* defined in [ITU-T Q.4068] is required).



NOTE 3 – This needs to be checked and considered.

- c. APIw/APIx: there is a question whether uniform resource description model may be needed. A question of whether APIw can also be used (called) directly by a user GUI or tool (similarly to APIr).
  - d. APIz: is already addressed in deliverable D2.2, there is a question regarding APIw/APIx and resource description model.
5. Regarding the question of priorities there is a possibility in implementation efforts, the following are key to be considered (this subject could be for further consideration):
- a. APIr, APIw/APIx, APIz.
  - b. Identity provider/authentication/authorisation (and relevant APIs).
- NOTE 4 – proposed security framework for the testbeds federations reference model defined in [ITU-T Q.4068] is defined in this document, and so further study should consider that the security framework should evolve to address various security and even trust related aspects in testbeds federations.
6. The interfaces inside a testbed domain seem have the following implications that may need to be checked or considered further if necessary:
- a. There is a question whether the following interfaces are relevant inside a testbed domain only: APIa, APIb, APIc, APId, APIe, APIf, APIg, APIh, APIi, APIj, APIk, GUI\_l , GUI\_m (APIm), APIn, APIo, APIp, APIq, APIr, APIu, APIv.
7. There is question whether the interfaces that are important from a federation point of view are the ones: APIr, APIs, APIw, APIx, APIy/GUI\_y, APIz.

## Bibliography

- [b-ITU-T QSTR.FTT] ITU-T Technical Report QSTR.FTT, *Federated testbeds taxonomy*.
- [b-ITU-T QSTR-TFR] ITU-T Technical Report QSTR-TFR, *Testbeds Federation roadmap*.
- [b-ITU-T QSTR-UCFTBS] ITU-T Technical Report QSTR-UCFTBS, *Use cases for federated testbeds and business scenarios*.



## SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
<b>Series Q</b>	<b>Switching and signalling, and associated measurements and tests</b>
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems