ITUPublications

International Telecommunication Union
Standardization Sector

ITU Focus Group Technical Specification

(04/2024)

Focus Group on Testbeds Federations for IMT-2020 and beyond

(FG-TBFxG)

FG-TBFxG-TS-D3.1

Evolution of the testbeds federations reference model

Technical Specification ITU FG-TBFxG-TS-D3.1

Evolution of the testbeds federations reference model

Summary

This Technical Specification covers the following aspects:

- APIs invocations 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 federations reference model APIs defined in Recommendation ITU-T Q.4068 and the extensions listed in FG-TBFxG deliverable D2.2;
- 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.

Keywords

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

Introduction

In complementing the elements in scope of this document, and considering items that extend the work achieved in specification of the testbeds federations reference model defined in Recommendation ITU-T Q.4068, the following aspects are also of relevance:

1) The work done in other SDOs/Fora on various concepts may be considered in describing the functionality of *Inter-Testbed E2E Universal Resource Broker for Testbed Federations* defined in Recommendation ITU-T Q.4068. For instance, components and mechanisms that can be used in "Search/Query Operations" by testbeds users in discovering testbeds of certain capabilities, proximity and other attributes, and other non-functional attributes such as reputation of testbed (based on the scoring of Testbeds).
2) Industry adoption of certain testbeds such as those being built by SDOs/Fora (e.g., IEEE) in instrumenting APIs for testbeds federations into such testbeds to help built testbeds that conform to the testbeds federations reference model ([ITU-T Q.4068]). The value is that the resultant testbeds would be "*federate'able*" with other testbeds that conform to the reference model for testbeds federations defined in [ITU-T Q.4068]. Beyond basic interconnection of testbeds, testbeds federations capabilities based on [ITU-T Q.4068] should be pursued by the industry and research communities.
3) Possibility to run Proof-of-Concepts (PoCs) on IMT-2020 and beyond technologies, test scenarios and on federation capabilities testing in assessing and ensuring testbeds conformance to the reference model defined in [ITU-T Q.4068] should be considered by industry and research communities.

4) There are APIs such as the TM Forum's Business Assurance related APIs [b-TMF-Business-Assurance] and Service Assurance related APIs [b-TMF-Service-Assurance] that could be used in the context of Testbeds Federations and Testbed-as-a-Service as well.

5) There are other types of APIs developed by organizations such as Broadband Forum (BBF) in their PoC Projects and Open-Source Initiatives [b-BBF-OpenSource-APIs], and there is potential that such APIs can be exploited in the context of Testbeds Federations to build testbeds that conform to the reference model ([ITU-T Q.4068]).

6) The study carried out by FG-TBFxG concluded that APIs in IEEE 2302-2021 on IEEE Standard for Intercloud Interoperability and Federation (SIIF) are applicable in implementing APIs for testbeds federations that complement the APIs defined in the reference model defined ([ITU-T Q.4068]).

Note

This is an informative ITU-T publication. Mandatory provisions, such as those found in ITU-T Recommendations, are outside the scope of this publication. This publication should only be referenced bibliographically in ITU-T Recommendations.

Change Log

This document contains Version 1.0 of the ITU-T FG-TBFxG D3.1 Technical Specification "Evolution of the Testbeds Federations Reference Model" approved at FG-TBFxG eighth meeting held in Sophia Antipolis, France from 10 to 12 April 2024.

Acknowledgement

This Technical Specification was prepared under the leadership of Dr.-Ing. Giulio Maggiore (Telecom Italia, Italy), Dr.-Ing. Ranganai Chaparadza (Capgemini Engineering, Germany) and Dr.-Ing. Tayeb Ben Meriem (IPv6 Forum, France), who served as the FG-TBFxG chair, FG-TBFxG vice-chair/WG3 chair and FG-TBFxG vice-chair/WG3 vice-chair respectively.

It is based on the contributions of various authors who participated in the Focus Group activities. FG-TBFxG appreciates Dr. Taesang Choi (ETRI, Korea Rep.of), Dr. Brecht Vermeulen (Ghent University/imec, Belgium) and Dir. MoonKook Park (Mobigen, Korea Rep.of) for their inputs to this Technical Specification.

Dr.-Ing. Ranganai Chaparadza (Capgemini Engineering, Germany), Dr.-Ing. Tayeb Ben Meriem (IPv6 Forum, France) and Dr.-Ing. Muslim Elkotob (Vodafone, Germany) served as the main Editors of this Technical Specification.

Mr Denis Andreev (FG TBFxG Advisor) and Ms Emmanuelle Labare (FG-TBFxG Assistant) served as the FG-TBFxG Secretariat.

Editor:	Dr.-Ing. Ranganai Chaparadza Capgemini Engineering, Germany	Email: ranganai.chaparadza@capgemini.com
Editor:	Dr.-Ing. Tayeb Ben Meriem IPv6 Forum, France	Email: tay.b52@yahoo.com
Editor:	Dr.-Ing. Muslim Elkotob Vodafone, Germany	Email: muslim.elkotob@vodafone.com

Table of Contents

Technical Specification ITU FG-TBFxG-TS-D3.1

Evolution of the testbeds federations reference model

1 Scope

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

- "APIs Invocations Framework" that clarifies how Generic APIs invoke testbed-specific APIs.
- Workflow Scenario Illustration involving a user accessing and using a testbeds federations ecosystem based on the testbeds federations reference model APIs defined in [ITU-T Q.4068] and the extensions listed in FG-TBFxG deliverable D2.2.
- 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 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 testbeds 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 federations 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 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

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

[IEEE 2302-2021] IEEE 2302-2021, *IEEE Standard for Intercloud Interoperability and Federation (SIIF)*.

[IETF RFC 2865] *Remote Authentication Dial In User Service (RADIUS)*.

[IETF RFC 6733] *Diameter Base Protocol*.

[IETF RFC 8907] *The Terminal Access Controller Access-Control System Plus (TACACS+) Protocol*.

3 Definitions

3.1 Terms defined elsewhere

This Technical Specification uses the following term defined elsewhere:

3.1.1 federateable [ITU-T FG-TBFxG D0.1]: Possess the ability or capability to be federated with other testbed(s) by conforming to the reference model for testbeds federations.

3.2 Terms defined in this Technical Specification

None.

4 Abbreviations and acronyms

This Technical Specification uses the following abbreviations and acronyms:

AMC Autonomic Management and Control
API Application Programming Interface
CSP Communications Service Provider
EC European Commission
ETSI European Telecommunications Standards Institute
FIRE Future Internet Research and Experimentation
FWA Fixed Wireless Access
GANA Generic Autonomic Networking Architecture
GENI Global Environment for Networking Innovations
IEEE Institute of Electrical and Electronics Engineers
IoT Internet of Things
ISV Independent Software Vendor
MEC Multi-Access Edge Computing
ONP Open Networking Platform
O-RAN Open Radio Access Network
PoC Proof of Concept
SDO Standards Development Organization
SME Small to Medium Enterprise
TIP Telecom Infrastructure Project

5 Conventions

None.

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 federations reference model ([ITU-T Q.4068]) and provides the insights on 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 such as the ***Partner Onboarding APIs*** that are defined by TM Forum in [b-TMF-OnboardingMgnt-API-REST], 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 such as the ***Customer OnBoarding API*** (also defined in TM Forum in [b-TMF-CustomerMgnt-API-REST]) could also be considered for adding to APIs of the ***Inter-Testbed E2E Universal Resource Broker for Testbed Federations*** defined in [ITU-T Q.4068].
- APIs defined in [IEEE 2302-2021] 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, the deliverable D2.2 has adopted some APIs from [IEEE 2302-2021] on IEEE Standard for SIIF in adding to the APIs defined in [ITU-T Q.4068].
- APIs that use Digital Twins should be considered for further study, as to 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 is a subject for further study.

Annex A lists some points/aspects for further study to see whether they have already been addressed by the FG-TBFxG deliverables D2.2, D0.2 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.

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]. FG-TBFxG deliverable D2.2 provides RESTful specification of the Generic APIs of the testbeds federations reference model defined in [ITU-T Q.4068].

7 APIs Invocations Framework

The APIs invocations framework for the for testbeds federations 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 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 the FG-TBFxG deliverable D2.2.

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 federations 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 federations 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 Doman entity.

Table 1 – Generic APIs for Testbed as a Service (TaaS) and comments on fundamental invocations conditions

#	API name (FG-TBFxG D2.2)	API description (FG-TBFxG D2.2)	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 during the time of testing that the target is integrated well with the Caller and is functioning well. It may also be invoked during the time when the resource is about to be reserved/booked for use by a User who has been permitted/admitted to use of the testbed.
2	APIb	The API is used for enabling a test manager to interact with the resource, in cases where the resource is either a component under test (CUT) or system under test (SUT), or provides an interface that can be used by a Test System to configure it, such that the test manager may configure the resource as may be required for some test scenario,	The API is invoked by the Caller during the time of testing that the target is integrated well with the Caller and is functioning well. Also, during execution of certain test scenarios that require the resource to take part.

Table 1 – Generic APIs for Testbed as a Service (TaaS) and comments on fundamental invocations conditions

#	API name (FG-TBFxG D2.2)	API description (FG-TBFxG D2.2)	Complementary comments on fundamental invocations conditions
		and/or pull test results from the resource after a completion of a test.	
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 on the resource.	The API is invoked by the Caller during the time of testing that the target is integrated well with the Caller and is functioning well. Also, 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 on the resource.	The API is invoked by the Caller during the time of testing that the target is integrated well with the Caller and is functioning well. Also, during periodic asynchronous updates on state by the resource.
5	APIe	The API is used by 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 testbed admin intends to view the state of certain resources before deciding to cause connectivity to be established among resources in the testbed and/or establishing connectivity to resources in other testbeds.	The API is invoked by the Caller during the time of testing that the target is integrated well with the Caller and is functioning well. Also, during the time when testbeds and resource connectivity across testbed domains is to be established, or during the time of resource booking/reservation for a User of the testbed admitted making use of the testbed.
6	APIf	The API is used by test manager to pull information about Level-0 and Level-1 Resources that may be required to participate in a certain test scenario a testbed user may want to execute. The APIs is also meant for use by a Test Manager to pull Information about the state of resources (Level-0 and Level-1) from the real-time state repository, especially when state of a resource plays a role during the execution of certain test cases or when the test manager may require to use the state of certain resources in deciding to cause connectivity to be established among resources in the testbed and/or	The API is invoked by the Caller during the time of testing that the target is integrated well with the Caller and is functioning well. 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 (FG-TBFxG D2.2)	API description (FG-TBFxG D2.2)	Complementary comments on fundamental invocations conditions
		establishing connectivity to resources in other testbeds.	
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 during the time of testing that the target is integrated well with the Caller and is functioning well. 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 services request(s) received from prospective testbed user(s). In case 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 for orchestration of 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 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 testbed broker to obtain information about capabilities and state of resources directly under the management and control responsibility of the Level-1 resource.	The API is invoked by the Caller during the time of testing that the target is integrated well with the Caller and is functioning well. Also invoked during the use case scenarios described in the API description.
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 during the time of testing that the target is integrated well with the Caller and is functioning well. Also invoked during the

Table 1 – Generic APIs for Testbed as a Service (TaaS) and comments on fundamental invocations conditions

#	API name (FG-TBFxG D2.2)	API description (FG-TBFxG D2.2)	Complementary comments on fundamental invocations conditions
			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 the Level-1 resource, whenever there are changes that have occurred on the resource.	The API is invoked by the Caller during the time of testing that the target is integrated well with the Caller and is functioning well. 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 the Level-0 resource, whenever there are changes that have occurred on the resource.	The API is invoked by the Caller during the time of testing that the target is integrated well with the Caller and is functioning well. Also invoked during the use case scenarios described in the API description.
12	APIl (or GUI_l)	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 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.	The API/GUI is invoked by the Caller during the time of testing that the target is integrated well with the Caller and is functioning well. Also invoked during the use case scenarios described in the API description.
13	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	The API/GUI is invoked by the Caller during the time of testing that the target is integrated well with the Caller and is functioning well. 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 (FG-TBFxG D2.2)	API description (FG-TBFxG D2.2)	Complementary comments on fundamental invocations conditions
		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.	
14	APIn	The API is used by the testbed resource broker to register itself into the testbed management system, 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 during the time of testing that the target is integrated well with the Caller and is functioning well. Also invoked during the use case scenarios described in the API description.
15	APIo	The API is used by a test manager to register itself into 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 a considered as a resource itself.	The API is invoked by the Caller during the time of testing that the target is integrated well with the Caller and is functioning well. 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	The API is invoked by the Caller during the time of testing that the target is integrated well with the

Table 1 – Generic APIs for Testbed as a Service (TaaS) and comments on fundamental invocations conditions

#	API name (FG-TBFxG D2.2)	API description (FG-TBFxG D2.2)	Complementary comments on fundamental invocations conditions
		upon the invocation of the API by the testbed management system.	Caller and is functioning well. 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 involved 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) to some invocations triggered earlier on the resource by the test manager.	The API is invoked by the Caller during the time of testing that the target is integrated well with the Caller and is functioning well. Also invoked during the use case scenarios described in the API description.
18	APIr	The API is to be used 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.	The API is invoked by the Caller during the time of testing that the target is integrated well with the Caller and is functioning well. Also invoked during the use case scenarios described in the API description.
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 invoked by the Caller during the time of testing that the target is integrated well with the Caller and is functioning well. 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 (FG-TBFxG D2.2)	API description (FG-TBFxG D2.2)	Complementary comments on fundamental invocations conditions
		APIs is using the same uniform resource description model that APIx uses.	
20	APIt	The API is used for providing test results to a test manager(s) that involved 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 during the time of testing that the target is integrated well with the Caller and is functioning well. 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 to 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 during the time of testing that the target is integrated well with the Caller and is functioning well. 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 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 during the time of testing that the target is integrated well with the Caller and is functioning well. Also invoked during the use case scenarios described in the API description.
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	The API is invoked by the Caller during the time of testing that the target is integrated well with the Caller and is functioning well. 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 (FG-TBFxG D2.2)	API description (FG-TBFxG D2.2)	Complementary comments on fundamental invocations conditions
		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.	
24	APIx	The API is used by the testbed resource broker to push updated information (updates) about 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 during the time of testing that the target is integrated well with the Caller and is functioning well. 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 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.	The API/GUI is invoked by the Caller during the time of testing that the target is integrated well with the Caller and is functioning well. 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 (FG-TBFxG D2.2)	API description (FG-TBFxG D2.2)	Complementary comments on fundamental invocations conditions
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 topology information pertaining to their interconnection and federations with other testbeds. 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 during the time of testing that the target is integrated well with the Caller and is functioning well. Also invoked during the use case scenarios described in the API description.

8 Workflow scenarios illustrations

8.1 Overview

This clause provides illustrations on workflow scenarios. The main focus is a workflow scenario illustration involving a user of testbeds 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 FG-TBFxG deliverable D2.2.

8.2 Primary enabling workflow

The primary workflow concerns the preparation of individual testbed domains meant to participate in a federation to be ready to serve users in discovering the testbeds and use them in 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 Rfps involved (identification tags)	Description
Testbed admin	**Step-1**: 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	**Step-2**: Collaboratively work with Testbed admins of other Testbed Domains to establish connectivity between the Testbed Domain with other Testbed Domains. This is determined by policies and SLAs allowing the Testbeds to be interconnected and to participate to federation scenarios of which Testbeds Users can benefit from, 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	RfpT, RfpU, RfpS, RfpR	

Table 2 – The Primary Enabling Workflow

Actor	Step of Activity Execution	API(s) or GUI or Rfps involved (identification tags)	Description
	Rfps (RfpT, RfpU, RfpS, RfpR,) in order, using necessary means like VLANs, VPNs, physical connectivity, etc.		
Testbed admin	**Step-3**: Collaboratively work with Testbed admins of other Testbed Domains that have been interconnected with this Testbed domain in **Step-2**, and with the **Broker Admin** of the **inter-testbed E2E universal resource broker for testbeds federation** to make the interconnected Testbeds get connected to the **inter-testbed E2E universal resource broker for testbeds federation**.	RfpQ, APIw, APIx	
Testbed admin	**Step-4**: Communicates in some way or makes aware to **Broker admin** of the Testbed domain that the Testbed domain is ready to participate in Testbeds Federations and Testbed Services offering to prospective Users		
Broker admin of the Testbed domain	**Step-5**: Collaboratively work with the **Broker Admin** of the **inter-testbed E2E universal resource broker for testbeds federation**, 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 and	APIl (or GUI_l) 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 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

Table 2 – The Primary Enabling Workflow

Actor	Step of Activity Execution	API(s) or GUI or Rfps involved (identification tags)	Description
	ready to participate in federations and be discovered by prospective users through the **inter-testbed E2E universal resource broker for testbeds federation**. And if desirable, collaboratively work with the **Broker Admin** of the **inter E2E brokers global universal resource broker for testbeds federation**, 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 **inter E2E brokers global universal resource broker for testbeds federation**.		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.
Broker admin for the **"inter-testbed E2E universal resource broker for testbeds federation"**	**Step-6**: Perform the activities that ensure that interconnected Testbeds in Step-2 are registered with the Broker and are ready for users to discover them and seek to use them.	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

Table 2 – The Primary Enabling Workflow

Actor	Step of Activity Execution	API(s) or GUI or Rfps involved (identification tags)	Description
			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 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
Test suite/cases designer and test executer	**Step-1**: Using an appropriate Tool, connect to the APIz of the **inter-testbed E2E universal resource broker for testbeds federation**. 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 capabilities, topology information pertaining to their interconnection and federations with other testbeds, and also discover other	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 topology information pertaining to their interconnection and federations with other testbeds. A prospective testbed user (test suite/cases designer and test executer) can query the broker

Table 3 – User-oriented workflow scenario illustration

Actor	Step of Activity Execution	API(s) or GUI involved (identification tags)	Description
	information such as cost of using a particular testbed or whole federated testbeds, and some KPIs related to Testbeds and their federations.		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.
Test suite/cases designer and test executer	**Step-2**: 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 *Testbed resource brokers* 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 **inter-testbed E2E universal resource broker for testbeds federation**.	APIz	(as described above)
inter-testbed E2E universal resource broker for testbeds federation	**Step-3**: Relays the request to the targeted ***Testbed resource broker(s)*** 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

Table 3 – User-oriented workflow scenario illustration

Actor	Step of Activity Execution	API(s) or GUI involved (identification tags)	Description
			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.
Testbed resource broker(s) of targeted testbed(s) domain(s)	**Step-4**: Perform admission control on the Request and the User launching the request, and sends the results to the User via APIx of **inter-testbed E2E universal resource broker for testbeds federation**.	APIx	The API is used by the testbed resource broker to push updated information (updates) about 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.
Test suite/cases designer and test executer	**Step-5**: 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 *Testbed resource brokers* of the	APIz	(as described above)

Table 3 – User-oriented workflow scenario illustration

Actor	Step of Activity Execution	API(s) or GUI involved (identification tags)	Description
	targeted Testbeds domains.		
Testbed resource broker(s) of targeted testbed(s) domain(s)	**Step-6**: 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 a 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 **inter-testbed E2E universal resource broker for testbeds federation**.	APIx	(as described above)
Test suite/cases designer and test executer			
Test suite/cases designer and test executer	**Step-7**: If Test Session/Service Request(s) launched to specific targeted **Testbed resource broker(s) of targeted testbed(s) domain(s)** 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 Test Manager(s) in the Testbed domain(s) directly at the point when the admitted Test	APIr	The API is to be used 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

Table 3 – User-oriented workflow scenario illustration

Actor	Step of Activity Execution	API(s) or GUI involved (identification tags)	Description
	Session(s) has been configured by the Testbed broker(s), to start working on Test Cases designs and/or executions.		execute test cases directly on the test manager without uploading test cases/suites from outside.
Test suite/cases designer and test executer	**Step-8**: 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 use them in 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 consideration 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, e.g., in the case of [b-IEEE INGR-TestbedsWG-RoadmapChapter], [b-AAA-use in federated-Testbeds] and

other cases in literature. Fault-tolerant Distributed AAA Architectures are possible [b-K-et-al-Fault-Tolerant-Distributed-AAA].

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 higher level for *Actors* on higher level outside 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 higher level for *Actors* on higher level outside 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 with 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]):

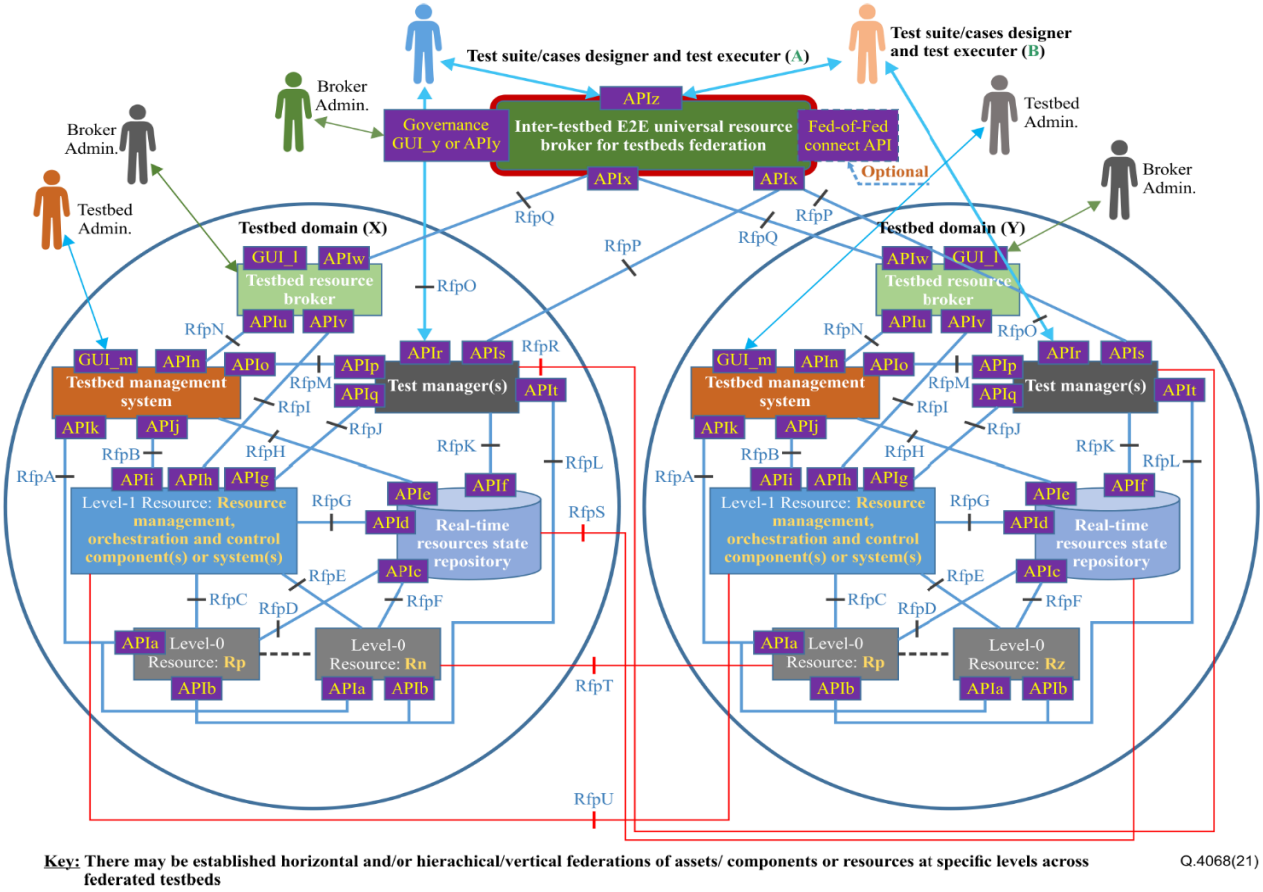

Figure 1 – Generic federated testbed model [ITU-T Q.4068]

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

1) Start with implementing the testbed domain concept by way of taking a bottom-up approach of implementing Level-0 and Level-1 Resources and manually testing them in terms of their functionality and integration.

2) Deploying 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, activating and testing 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) Implementing the "APIs Invocations Framework" prescribed in this document.

5) Integrating the Level-0 and Level-1 Resources with the Testbed Management System.

6) Integrating 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) Interconnecting the Testbed Domain with another Testbed Domain(s) and testing the connectivity among Testbed Domains, and testing all the Rfps concerning the connectivity.

8) Activating the federation behaviours of the Testbed Domain and testing the federation behaviours as it is integrated with the *Inter-Testbed E2E Universal Resource Broker for Testbed Federations*.

9) Considerations for integrations and testing pertaining making testbeds domains visible higher up to the *Inter E2E brokers global universal resource broker for testbeds federation* level.

NOTE – These procedures above form the basic guiding illustrations on how instantiations of the testbeds federations reference model ([ITU-T Q.4068]) should be carried out by the global community. This serves 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, 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 it 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 federations 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 FG-TBFxG deliverable D1.1. For example, an insight on how to a scope for a PoC on the use case #7 (UC07) listed in the FG-TBFxG deliverable D1.1 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 of relevance 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, 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 federations 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.

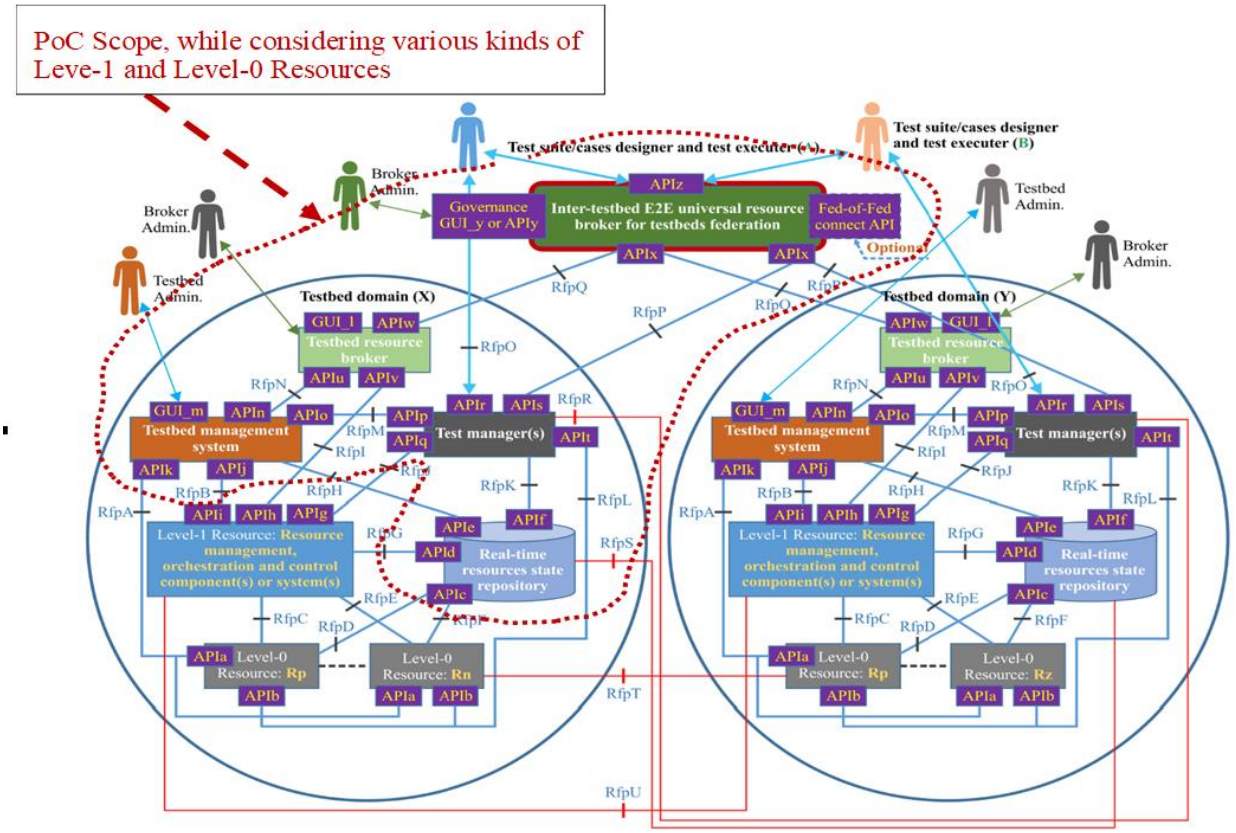

Figure 2 – General scope for use in defining PoC projects on testbeds federations

Figure 3 illustrates the scope and approach for running a PoC on the use case #07 (UC07) listed in FG-TBFxG deliverable D1.1, on use of testbeds federations for testing federated Autonomic Management and Control (AMC) by federated ETSI GANA Knowledge Planes (KPs) Platforms for Autonomic/autonomous 5G and beyond networks.

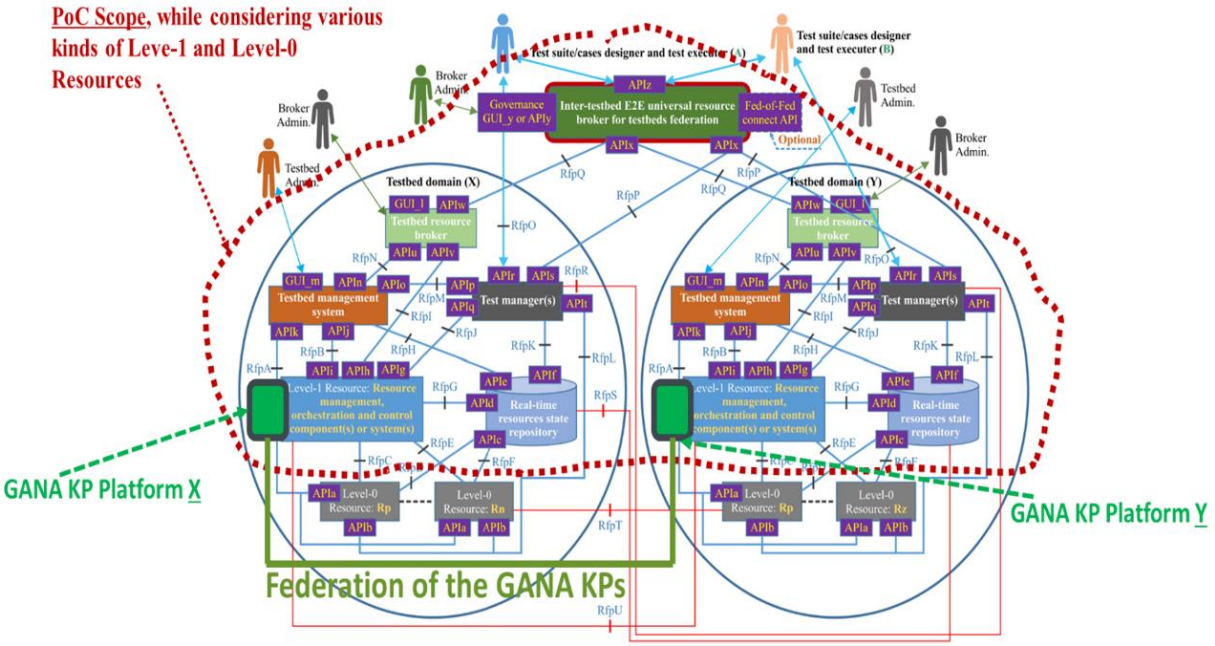

Figure 3 – Testing federated Autonomic Management and Control (AMC) by federated ETSI GANA Knowledge Planes (KPs) Platforms for Autonomic/autonomous 5G and beyond networks

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 that conform to the testbeds federations 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. An example of a roadmap to be considered is the IEEE INGR Future Networks Testbeds Working Group (WG) Roadmap Chapter Editions series of 2021, 2022, 2023, 2024 [b-IEEE INGR-Testbeds-Roadmap] and future editions of this roadmap.

Annex A

Points/aspects for further studies

This annex lists some points/aspects for further studies in order to see whether they have already been addressed by the FG-TBFxG deliverables D2.2, D0.2 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 the FG-TBFxG deliverable D2.2.
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, APIt, 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 D0.1 FG-TBFxG] FG-TBFxG Technical Specification D0.1 (2025), *Federated testbeds taxonomy*.

[b-IEEE INGR-Testbeds-Roadmap] https://futurenetworks.ieee.org/roadmap
https://futurenetworks.ieee.org/roadmap/testbed-working-group

[b-TBFxG-I-028R1] IEEE, Presentation that provides Information on IEEE Std 2302-2021 that could be considered and referenced for potential use in Testbed Federation APIs.

[b-AAA-use in federated-Testbeds] Fulvio Risso, et al, *Open Federated Testbed: Technical Overview and Live Demo*: https://sdn.ieee.org/images/files/pdf/FederatedTestbeds/november2016_risso_-_open-federated-testbed.pdf

[b-BBF-OpenSource-APIs] Open Broadband Software; Open Broadband – Broadband Access Abstraction (OB-BAA)
https://www.broadband-forum.org/open-broadband/open-broadband-software
https://www.broadband-forum.org/open-broadband/open-broadband-software/open-broadband-broadband-access-abstraction-ob-baa

[b-K-et-al-Fault-Tolerant-Distributed-AAA] K. Huhtanen and A. Kolehmainen (2022), *Fault-tolerant Distributed AAA Architecture Supporting Connectivity Disruption*, 2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN), Barcelona, Spain, pp. 405-407, doi: 10.1109/ICUFN55119.2022.9829606.

[b-TMF-business-API] FIWARE TM Forum Business API Ecosystem.
https://fiwaretmfbizecosystem.docs.apiary.io/#

[b-TMF-Business-Assurance] TM Forum Business Assurance Toolkit.
https://www.tmforum.org/toolkits/business-assurance-toolkit/

[b-TMF-CustomerMgnt-API-REST] TMF629 Customer Management API REST Specification R19.0.1.

[b-TMF-OnboardingMgnt-API-REST] TMF650 Onboarding Management API REST Specification R16.0.1.

[b-TMF-OSS-BSS] TM Forum OSS/BSS.
https://www.tmforum.org/resources/toolkit/agile-ossbss-toolkit/

[b-TMF-Service-Assurance] TMF656 Service Problem API REST Specification R18.5.1
https://www.tmforum.org/resources/specification/tmf656-service-problem-api-rest-specification-r18-5-0/
https://inform.tmforum.org/features-and-opinion/my-api-story-service-assurance