

ITU Focus Group Technical Specification

(04/2024)

Focus Group on Testbeds Federations for IMT-2020
and beyond

(FG-TBFxG)

FG-TBFxG-TS-D0.1

Federated testbeds taxonomy



Technical Specification ITU FG-TBFxG-TS-D0.1

Federated testbeds taxonomy

Summary

This Technical Specification defines taxonomy for the federated testbeds. It contains all the definitions of the terms used in the context of federated testbeds. This Technical Specification is the reference document for the glossary of the terms common to all the aspects linked to testbeds federation.

Keywords

API, automation, federation, test case, testbed, use case.

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 D0.1 Technical Specification "Federated testbeds taxonomy" 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), who served as the FG-TBFxG chair.

It is based on the contributions of various authors who participated in the Focus Group activities. FG-TBFxG appreciates Dr. Hisham Kholidy (State University of New York Polytechnic Institute, College of Engineering, US) for his inputs to this Technical Specification.

Dr.-Ing. Ranganai Chaparadza (Capgemini Engineering, Germany), Dr.-Ing. Tayeb Ben Meriem (IPv6Forum, France), Dr.-Ing. Muslim Elkotob (Vodafone, Germany), Dr. Shao Sicong (Arizona University, US) and Ing. Cédric Crettaz (Mandat International, Switzerland) 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

Editor: Dr. Shao Sicong,
Arizona University, US

Email: sicongshao@arizona.edu

Editor: Ing. Cédric Crettaz
Mandat International, Switzerland

Email: ccretgaz@mandint.org

© ITU 2025

Some rights reserved. This publication is available under the Creative Commons Attribution-Non Commercial-Share Alike 3.0 IGO licence (CC BY-NC-SA 3.0 IGO; <https://creativecommons.org/licenses/by-nc-sa/3.0/igo>).

For any uses of this publication that are not included in this licence, please seek permission from ITU by contacting TSBmail@itu.int.

Table of Contents

	Page
1 Scope.....	1
2 References.....	1
3 Definitions	1
3.1 Terms defined elsewhere	1
3.2 Terms defined in this Technical Specification	2
4 Abbreviations and acronyms	3
5 Conventions	4
6 Other perspectives on taxonomy for testbeds federation.....	4
6.1 The use of federated learning in certain testbeds domains.....	4
6.2 Scalability of federated learning use in federated testbed domain	4
6.3 Data Partitioning in FL	5
6.4 Federated learning techniques	5
Bibliography.....	6

Technical Specification ITU FG-TBFxG-TS-D0.1

Federated testbeds taxonomy

1 Scope

This Technical Specification contains all the terms and their definitions used in the context of testbeds federation. It provides the references to the ITU-T Recommendations and other references defining existing terms related to testbeds federation. The sources of the definitions are published ITU-T Recommendations and other standards published by other SDOs. This Technical Specification presents the taxonomy for federated testbeds.

2 References

- [[ITU-T Q.4068](#)] Recommendation ITU-T Q.4068 (2021), *Open application program interfaces (APIs) for interoperable testbed federations*.
- [[ITU-T X.1812](#)] Recommendation ITU-T X.1812 (2022), *Security framework based on trust relationships for the IMT-2020 ecosystem*.
- [[ITU-T Y.2720](#)] Recommendation ITU-T Y.2720 (2009), *NGN identity management framework*.
- [IEC 80001-1:2021] Standard IEC 80001-1:2021, *Application of risk management for IT-networks incorporating medical devices – Part 1: Safety, effectiveness and security in the implementation and use of connected medical devices or connected health software*.
- [ISO 12651-2:2014] Standard ISO 12651-2:2014, *Electronic document management. Vocabulary. Part 2: Workflow management*.

3 Definitions

3.1 Terms defined elsewhere

This Technical Specification uses the following terms defined elsewhere:

- 3.1.1 testbed** [ITU-T Q.4068]: Platform to realise scientific tests within new technologies on an environment fully controlled by experimenters.
- 3.1.2 federation** [ITU-T Y.2720]: Establishing a relationship between two or more entities or an association comprising any number of service providers and identity providers.
- 3.1.3 stakeholder** [ITU-T X.1812]: Individual, organization, or group of people that has an interest in, or might be affected by the system being contemplated, developed, or deployed.
- 3.1.4 process** [IEC 80001-1:2021]: Set of interrelated or interacting activities which transforms inputs into outputs.
- 3.1.5 workflow management** [ISO 12651-2:2014]: Automation of a process (3.28), in whole or part, during which electronic documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules (3.36).

NOTE – The automation of a process is defined within a process definition, which identifies the various process activities, procedural rules, and associated control data used to manage a workflow. A loose distinction is sometimes drawn between production workflow, in which most of the procedural rules are defined in advance, and ad hoc workflow, in which the procedural rules may be modified or created during the operation of the process.

3.2 Terms defined in this Technical Specification

This Technical Specification defines the following terms:

3.2.1 testbeds federation: The interconnection of testbeds is a way that the testbeds are seamlessly viewed as a distributed platform that presents more aggregated capabilities from the diverse testbeds, such that a user (test executor) can execute tests on the platform without having to approach each testbed individually to request for testbed service, while each testbed can still offer testbed services even as a standalone entity.

3.2.2 testbed as a service: Pertains to the provision and exposure of a testbed by a testbed's owner to prospective users of the testbed (consumers of the testbed service) based on a policy framework that governs the usage of the testbed by the prospective users.

NOTE 1 – The policy framework may be based on SLAs, charging model and time frame and testbed availability by which a user or users would want to use the testbed to run tests, analyse tests results and utilize the results.

NOTE 2 – Cloud-based implementations may be an option.

3.2.3 testbed domain: Set of test resources or facilities that can be used as a standalone and self-contained structure owned by a stakeholder and can be made to interconnect with other test resources (may be owned by a separate stakeholder) in a federation ecosystem such that it can be used together with other resources/facilities of the federation.

3.2.4 business scenarios: Context in which supplier and consumer interact to fulfil an agreement on how a consumer consumes a product or service provided by supplier.

3.2.5 federation ecosystem: A community of entities or players that need to be involved in the testbeds federation, and includes developers, users/consumers of testbed services, and suppliers of testbeds.

3.2.6 workflow: An ordered interaction sequence between entities that perform tasks required to achieve an objective. The objective is considered achieved after each entity involved has successfully executed its tasks in the order in which each participant entity must execute a task or tasks that may depend on tasks completed or initiated by some entity or entities.

3.2.7 resource broker: A functional entity used to maintain real-time knowledge/information about resources of its associated testbed and provides information about availability of resources and capabilities of the testbed to potential users, while being responsible for admission control of user requests.

3.2.8 test manager: A functional entity for use in defining, editing, compiling, scheduling and executing test cases and test scenarios.

3.2.9 testbed management system: A functional entity for use in the management and governance of use of assets that constitute a testbed.

NOTE – Such a system normally exposes a GUI to the user (called testbed administrator).

3.2.10 interoperability of testbeds: The ability of different testing environments or platforms to seamlessly work together, share resources, exchange data, and communicate effectively.

NOTE – This facilitates the validation, verification, and testing of new technologies, ensuring that they can function effectively in heterogeneous and complex operational settings.

3.2.11 semantic technologies: Technologies that play a crucial role in enhancing the capabilities of federated testbeds by enabling interoperability and facilitating data exchange between different testing environments.

NOTE – These technologies, which include ontologies, semantic annotation, knowledge graphs, semantic web services, and natural language processing, provide a common framework for representing and exchanging data in a standardized and meaningful way.

3.2.12 federated learning (FL) technique: A cutting-edge machine learning technique that enables developers to train AI models with supercomputer and hypercomputer on decentralized data without the need to centralize or share that data.

NOTE 1 – The benefits of FL over traditional machine learning approaches are numerous. FL assists in ensuring privacy, data diversity, real-time continual learning, hardware efficiency, and privacy preservation. Furthermore, it enables access to heterogeneous data, which is particularly useful when traditional machine learning may be impeded by network unavailability in edge devices.

NOTE 2 – Federated learning techniques that are used to implement the federation of the testbed.

NOTE 3 – Supercomputer and hypercomputer are very relevant for federated learning.

3.2.13 information leakage: Information leakage occurs when confidential or sensitive data is unintentionally or maliciously exposed, inside or outside an organization, due to inadequate security measures, personnel negligence, or other vulnerabilities.

NOTE – Countermeasures against information leakage include employing encryption technologies, implementing access control measures, regularly auditing for vulnerabilities, and comprehensive security training and awareness programs for employees. In the context of machine learning, information leakage, also known as data leakage or target leakage, causes the predictive scores to overestimate the model's utility at prediction time in a production environment.

3.2.14 administrative entity: A body or structure that is considered as having an owner who governs the body or structure in terms of its interactions with other bodies or structures.

3.2.15 testbed autonomy: The capacity of a testbed to facilitate the development, evaluation, and testing of autonomous technologies in such a way that the testing may be performed using the testbed, while the testbed itself is considered as owned by an autonomous administrative entity.

NOTE – Testbed autonomy encompasses a broad range of applications such as robotics and vehicle computing, etc.

3.2.16 testbed scalability: Ability of a testbed to be extended in terms of resources it can offer or test scenarios it can support.

NOTE – It is one of the key evaluation challenges that has to be considered when creating a realistic environment.

3.2.17 federatable: Possess the ability or capability to be federated with other testbed(s) by conforming to the reference model for testbeds federations.

4 Abbreviations and acronyms

This Technical Specification uses the following abbreviations and acronyms:

API	Application Programming Interface
FL	Federated Learning
GUI	Graphical User Interface
ML	Machine Learning
SLA	Service Level Agreement

5 Conventions

None.

6 Other perspectives on taxonomy for testbeds federation

Testbeds federation improves the interoperability amongst unique testbeds by sharing a common set of testbed APIs and tools to enable the collection of data from any of the participating testbeds. The

testbeds federation's ease of use and ability to provide access to open data creates direct benefits for connected testbeds, such as platform visibility and grown user base, which in turn help attract new types of users that may not be directly interested in the underlying facilities, but specifically in the applied semantic technologies. The expanded federated platform visibility gives the ability to develop synergies between industrial developers and academic experimenters [b-1] [b-2].

Traditionally, testbed facilities for network and systems research have been independent facilities, each owned and operated by a single administrative entity and designed for independent use [b-1] [b-2]. However, this isolated installation model conflicts with researchers' growing needs for experiments on a larger scale and with a greater diversity of network technologies. Federated models were recently introduced to retain the current testbed autonomy, strengths, scalability, and data privacy. These testbeds include federated models that work together to provide their resources within a common framework. Among the challenges of these federated testbeds are to establish trust while maintaining the autonomy of each federation setting, and to operate in a flexible and coordinated manner while avoiding central points of failure and cross-site dependencies. The federated testbeds are classified based on various basic building blocks namely: communication architecture, federation scale, privacy mechanism, data partitioning, and machine learning models that can be used in federated learning (FL) [b-2] [b-3].

NOTE – There are examples of diagrammatic (graphical) representations that tie together taxonomy on federated testbeds, with inclusion in federated learning [b-2] [b-3].

6.1 The use of federated learning in certain testbeds domains

Federated testbeds may use either a traditional centralized or decentralized learning model. The centralized model runs in the cloud, gathering information from all connected devices and sending back a FL model. In this centralized model, each device trains its local model on its private data. These updated local models are then sent to a central aggregation server [b-3] to compute the global FL model. The centralized design is always easy to implement but it suffers from the single point of failure issue. The decentralized design is preferred in some aspects because concentrating information on a single server may pose a risk.

The decentralized approach creates a global model by learning from multiple decentralized edge clients. In this model, data is distributed among many different nodes in a network, rather than being stored in a centralized location and network nodes communicate and share model parameters with each other without any server.

6.2 Scalability of federated learning use in federated testbed domain

FL can be divided into two types according to the size of the federation [b-3]: cross-silo and cross-device. The difference depends on the number of parties involved and the data stored by each party. In the cross-silo model, a relatively small number of trusted customers participate in training the overall model. In contrast, in the multi-device model, a large number of devices participate in training the overall model. However, these devices do not have large datasets to train a given model nor do they have the significant computing power that would allow them to train complex models.

6.3 Data partitioning in FL

FL can be classified into two groups based on different distribution patterns of sample space and feature space: horizontal FL and vertical FL. Horizontal FL is the case where the data distribution among participating training nodes shares the same feature space but differs depending on which data sample belongs to which training node. In contrast, in vertical FL, data on training nodes share the same sample space but differ in their feature space [b-1] [b-2] [b-3].

6.4 Federated learning techniques

FL is a sub-field of machine learning (ML) that can be also classified according to ML methods used to train the FL model while ensuring that their data remains decentralized. These methods include decision tree (DT), neural network (NN), random forest (RF), support vector machine (SVM), and clustering techniques such as the top-rated deep neural network (DNN) and k-Nearest Neighbor [b-1] [b-2] [b-3].

Bibliography

- [b-1] H. A. Kholidy, A. Karam, J. H. Reed and Y. Elazzazi (2022), *An Experimental 5G Testbed for Secure Network Slicing Evaluation*, 2022 IEEE Future Networks World Forum (FNWF), Montreal, QC, Canada, pp. 131-138, doi: 10.1109/FNWF55208.2022.00032.
 - [b-2] H. A. Kholidy and R. Kamaludeen (2022), *An Innovative Hashgraph-based Federated Learning Approach for Multi Domain 5G Network Protection*, 2022 IEEE Future Networks World Forum (FNWF), Montreal, QC, Canada, pp. 139-146, doi: 10.1109/FNWF55208.2022.00033.
 - [b-3] Brecko, Alexander & Kajáti, Erik & Koziorek, Jiri & Zolotová, Iveta (2022), *Federated Learning for Edge Computing: A Survey*, Applied Sciences, 12. 9124. 10.3390/app12189124.
-