ITUPublications

International Telecommunication Union
Standardization Sector

ITU-T Technical Report

(02/2025)

QSTR.FTT

Federated testbeds taxonomy

Technical Report ITU-T QSTR.FTT
Federated testbeds taxonomy

Summary

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

Keywords

API, automation, federation, testbed, test case, 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 Technical Report QSTR.FTT (ex. Q.FTT) "Federated testbeds taxonomy" approved at ITU-T SG11 meeting held in Geneva from 19 to 28 February 2025.

Editor:	Martin Brand A1 Telekom Austria Austria	Email: martin.brand@a1.at
Editor:	Sébastien Ziegler Mandat International Switzerland	Email: sziegler@mandint.org
Editor:	Cédric Crettaz, Mandat International Switzerland	Email: ccrettaz@mandint.org

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 Report ... 2

4 Abbreviations and acronyms ... 3

5 Conventions ... 3

6 Other perspectives on taxonomy for testbeds federation ... 3

6.1 The use of federated learning in certain testbeds domains ... 4

6.2 Scalability of federated learning use in the federated testbed domain ... 4

6.3 Data partitioning in federated learning ... 4

6.4 Federated learning techniques ... 5

Bibliography ... 6

Technical Report ITU-T QSTR.FTT

Federated testbeds taxonomy

1 Scope

This Technical Report contains all the terms and their definitions used in the context of testbeds federation. It provides 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 Report defines 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 Report 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 [b-ISO/PAS 19450]: 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 Report

This Technical Report defines the following terms:

3.2.1 testbeds federation: The method of integrating multiple testbeds into a unified platform that enhances their collective capabilities, creating a seamless and efficient testing environment.

NOTE – This integration allows users to execute tests without needing to interact with individual testbeds separately, simplifying access and improving usability. Simultaneously, each testbed retains its independence, ensuring it can continue to operate as a standalone entity while contributing to the broader, more powerful distributed platform.

3.2.2 testbed as a service (TaaS): The offering of a testbed by its owner to users provided according to a defined policy framework, allowing the testbed to be used for running tests, analysing results and utilizing the findings while adhering to the established policies.

NOTE 1 – This framework may include service level agreements (SLAs), pricing models, usage time frames, and testbed availability.

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

3.2.3 testbed domain: A collection of test resources or facilities that function independently under a single stakeholder's ownership, which can operate as standalone entities or integrated with other test resources (potentially owned by different stakeholders) within a federated ecosystem, enabling collaborative testing and resource sharing while maintaining individual ownership and control.

3.2.4 business scenarios: The context in which a supplier and consumer interact to fulfil an agreement on how 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, including 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.

NOTE – 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 that manages real-time information about a testbed's resources, including their availability and capabilities, and provides this information to potential users while also handling admission control for user requests, ensuring efficient resource allocation and utilization.

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 used in the management and governance of the assets that constitute a testbed.

NOTE – Such a system normally exposes a graphical user interface (GUI) to the user (called the 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: An advanced machine learning approach that allows artificial intelligence (AI) models to be trained on decentralized data using supercomputers and hypercomputers, without requiring data centralization or sharing.

NOTE – FL offers several advantages over traditional machine learning, including enhanced privacy, data diversity, real-time continual learning, and improved hardware efficiency. It also enables access to heterogeneous data, making it especially useful in scenarios where network connectivity is limited, such as on edge devices.

3.2.13 information leakage: The unintended or unauthorized exposure of confidential or sensitive data, either within or outside an organization, due to security weaknesses, human error, or malicious intent.

NOTE – Preventive measures include encryption, access controls, regular security audits, and employee training. In machine learning, information leakage – also called data or target leakage – can lead to overly optimistic predictive scores, reducing model reliability in real-world applications.

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

3.2.15 testbed autonomy: The ability of a testbed to support the development, evaluation, and testing of autonomous technologies while being managed by an independent administrative entity.

NOTE – Testbed autonomy applies to various fields, including robotics, vehicle computing, and other autonomous systems, enabling controlled and efficient testing environments.

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 federateable: The ability of a testbed to integrate with other testbeds by adhering to a reference model for testbed federations, enabling seamless interoperability and collaboration.

4 Abbreviations and acronyms

This Technical Report uses the following abbreviations and acronyms:

API	Application Programming Interface
DT	Decision Tree (DT)
GUI	Graphical User Interface
ML	Machine Learning
RF	Random Forest
SLA	Service Level Agreement
SVM	Support Vector Machine

5 Conventions

None.

6 Other perspectives on taxonomy for testbeds federation

Testbeds federation improves the interoperability among unique testbeds by sharing a common set of testbed application programming interfaces (APIs) and tools to enable the collection of data from any of the participating testbeds. Testbed federation's ease of use and ability to provide access to open data creates direct benefits for connected testbeds, such as the platform visibility and a growing user base which in turn helps attract new sorts 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.

Traditionally, testbed facilities for network and systems research have been independent facilities, with each facility being owned and operated by a single administrative entity and designed for independent use. However, this isolated installation model conflicts with researchers' growing need 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).

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

6.1 The use of federated learning in certain testbeds domains

Federated testbed 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 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 the federated testbed domain

FL can be divided into two types according to the size of the federation: 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 federated learning

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.

6.4 Federated learning techniques

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

Bibliography

[b-ISO/PAS 19450] Publicly Available Specification ISO/PAS 19450:2015, *Automation systems and integration – Object-process methodology*.