

Mahdi, Mahmoud A.; Mossad, Omar S.; ElNainay, Mustafa

Conference Paper

Research and Educational Federation Testbed: A New Architecture Design for the Collaborative Research Cloud

2nd Europe - Middle East - North African Regional Conference of the International Telecommunications Society (ITS): "Leveraging Technologies For Growth", Aswan, Egypt, 18th-21st February, 2019

Provided in Cooperation with:

International Telecommunications Society (ITS)

Suggested Citation: Mahdi, Mahmoud A.; Mossad, Omar S.; ElNainay, Mustafa (2019) : Research and Educational Federation Testbed: A New Architecture Design for the Collaborative Research Cloud, 2nd Europe - Middle East - North African Regional Conference of the International Telecommunications Society (ITS): "Leveraging Technologies For Growth", Aswan, Egypt, 18th-21st February, 2019, International Telecommunications Society (ITS), Calgary

This Version is available at:

<https://hdl.handle.net/10419/201751>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

Research and Educational Federation Testbed: A New Architecture Design for the Collaborative Research Cloud

Mahmoud A. Mahdi¹, Omar S. Mossad², Mustafa EINainay²

Abstract—We present a new architecture design with implementation to handle the unified testbed federation System, created initially for all the faculties and research centers in Egypt. It is based on self-management testbed with synchronize data and shared interfaces. The architecture includes the functions required to enable, manage and control testbeds over multiple sites for research and educational purposes. The main target of this research is to propose an architecture for the federation of Internet experimentation facilities, which extends across a diverse research community. This paper comes out of Collaborative Research Cloud (CRC) project funded by the National Telecommunications Regulatory Authority to create a unified testbed federation system to allow for remote access for existing research equipment in some faculties and research centers in Egypt as a bootstrapping to apply it later to all other research centers in Egypt. The main target of CRC project is to build a system that allocate resources to the authenticated users, keep track of system resources, and be accessed only by the authorized user. This paper presents the new architecture design for the federated testbed which includes the structure design, and implementation techniques of the main modules. The federation architecture is also designed with flexibility for further development and/or modification. The main federation system is divided into manageable processes that are grouped to sub-modules and modules that are built with abstraction.

I. BACKGROUND

The credibility of research results, new standards, and prototype testing; depends on the method of research validation. Several methods exist for the evaluation of research. The first and most popular method is using a simulation program because of its relative ease and low cost. The problem with simulations is that they make many assumptions to simplify calculations, which might compromise the credibility of results. In addition, large-scale simulations require resources or take duration that could go beyond what is feasible on personal computers. The second method is to acquire the hardware which has the advantage of leading to accurate results that reflect reality. But, in addition to the high cost, it has a limitation of reuse over group of researches and the overhead of setup both hardware and software. The last method, is remotely accessible testbeds offer researchers a more realistic tool of validation, while avoiding most of the problems associated with the management of hardware. Thus, it can be used for increasing the trust of the proposed systems in an economically feasible manner. With the massive deployment of equipment and research centers, a unified testbeds center has become a prerequisite for the provision of resources. However the independent provision

and ease of use process to researchers is still limited. In last past years, some of international organizations have been established unified testbeds to manage and take advantage of resources across multiple sites such as ORBIT [1], CORNET [2], NITOS [3], and CRC [4]. They already developed the infrastructure and tools needed for testbed management. However, they still need extension to reach their ultimate promise.

II. INTRODUCTION

A federation is defined as a union comprising a number of partially self-governing regions united by a central government under a common set of objectives. The purpose of federation is to establish the interconnection between two or more independent testbeds for creating a richer testing environment [5], which increase the experimentation multi-lateral benefits rather than the users of the individual testbeds. The portal allows researchers to access through out different testbeds. Therefore, the testbeds become more dynamic and evolve over time based on the requirements of the users. Furthermore, the federation portal supports user interface of testbeds control and management. Federation is considered a useful technique as it allows several independent testbeds to communicate with each other and provide a common interface to the users using minimum resources and costs. A testbed federation can have variety of hardware and software that are located far apart. Researches will be able to access these testbeds as if it was one larger testbed. The main benefit of the federation is to allow users to cross-validate their experiments under different conditions and System under Test (SUT). Additionally, a number of experiments can be performed using the necessary hardware from a group of testbeds without the need to have the entire experiment setup present at a single testbed.

III. EXPERIMENT LIFE CYCLE

In order to be able to understand the implications of the federating network experimentation facilities, its better to have a good understanding of all the relevant functions. For this purpose, it presents here the different aspects of the life cycle of the experiments [6]. The following are the mainly experiment lifecycle:

- 1) **Resource discovery:** Find the resources available in all facilities, and obtain the necessary information to match the required specifications.
- 2) **Resource Selection:** Specify all resources required during the experiment and matched the requirements

¹Faculty of Computers and Informatics, Zagazig University ,Egypt

²Faculty of Engineering, Alexandria University, Egypt

such as computing, networking, storage and library software.

- 3) **Resource reservation:** Allocate a time slot in which exclusive access and control over reserved resources is granted.
- 4) **Resource provisioning:**
 - a) Direct: Instantiating specific resources directly through API facility, and the experimenter is responsible for determining individual resources.
 - b) Orchestrated: Instantiating of resources through a functional component, which automatically selects resources that suit the experimenter's requirements.
- 5) **Experiment control:** Control the behavior of resources during the implementation of the experiment.
- 6) **Monitoring:**
 - a) Facility monitoring: Resource tools to oversee the behavior and performance of facilities which allowing system administrators to verify that the utilities perform correctly.
 - b) Infrastructure monitoring: Resource devices that collect data on the behavior and performance of services, technologies, and protocols to obtain measurements.
 - c) Experiment measuring: A set of experimental data created by frameworks or services that experimenter can deploy on their own.
- 7) **Permanent storage:** Store information related to the experiment beyond the lifetime of the experiment, such as experiment description, experiment results, disk images and measurements.
- 8) **Resource release:** Release the reserved resources after expiration or time out of the experiment.

IV. DESIGN CONSIDERATIONS

The user of the testbed federation system is aware of basic operations of a computer and web pages. The user also understands the standard terms used for operation. The system is built accessible for university's staff and students. The web-based unified testbed federation system is designed to work on all operating systems. The system is accessible through any laptop and desktop, that is connected to the back-end server. It is accessible at all times. The system is designed with flexibility for further development and/or modification. The system is divided into manageable processes that are grouped to sub-modules and modules that are built with abstraction.

V. CRC ARCHITECTURE

This section will present a detailed description of the designing of the Unified testbed federation System; which created initially for all the faculties and research centres in Egypt; and gives a detailed of the software architecture and design of main modules.



Fig. 1. Data flow concept between sites

A. Requirements

Generally, federation required a set of services, roles, database, application interface and networking. Based on existing federation portal framework, the main functional requirements should include the following:

- 1) User should trusted among federated authorities.
- 2) Enable access to federated resources.
- 3) Allow execute an experiment across testbeds.
- 4) Provide a uniform user interface for heterogeneous resources.
- 5) Offer a unified way to interact with resources function.

B. Design Description

The following list are main design description for testbed federation system:

- **Site independence:** each site must running based on its own information and resources without need of any external services.
- **Authorization Hierarchy:** the authorization level is used to inherit all permission from parent site. So, each site has a higher privilege to change and update its resources information or change the reservation plan.
- **Resource sharing:** While the site is running the federation system, each site can setup the details of resources and its sharing plan.
- **Sharing permission** The administrator must approve the request of sharing per site to accept the site resources.
- **Resources Synchronization** As well as, site is running independently, the site is responsible for distributing the entries information to other registered sites. And other sites is responsible for collecting all resources information from registered sites. Figure 1 shows the flow of data between sites

C. Integration Layers

The main task of federation is to share resources information with external sites. The federation services is controlled locally by users and operate self-dependant globally with other sites. Figure 2 show the federation service integration with testbed portal. The testbed portal allow users and other staff to view, reserve, and access local resources. The aim of integrating testbed portal with federation services is the allow external users and staff use of resources as well as testbed portal. Therefore, The federation service will allow the sites resources to be flatted over networking. There are three main components of federation testbeds: the federation API services, the Portal site, and System controller. To integrate portal framework with federation system, It will use an API

to link-between portal modules and the Back-End System. The following are the integrated layers ordered from top user view to back system operations.

- 1) **Front-End (Web-Application):** provides the GUI interface to system, to enable create accounts, accessing resources, scheduling, managing requests, and handling all user activity along with testbed site functions. It also include the following:
 - a) **Portal DB:** to stored information related to portal such as user access, resources, reservations and user log.
 - b) **Scheduler:** a system to manage resources' requests time and allocations.
 - c) **Access Token:** an interface to create a user access token, which will used to access resources during reserved session.
- 2) **Middle-ware (Web-API):** a set of APIs to allow federation framework to access the back-end services to enable all federation operations, tasks and inter-connectivity between sites. These operations enabled through peer-to-peer authentication to handle the resources control cross sites such as users reservations. It also include the following:
 - a) **Federation DB:** to stored information related to federation system such as keys, shared resources, and site information.
 - b) **Key Exchanged:** to exchange encrypted peer keys between sites.
 - c) **User authenticated:** to authorize federated user access another sites.
 - d) **Resources API:** To export resources status, and information to other federated site.
- 3) **Back-End (Service):** a back-engine which is responsible for all testbed experimentation functions. The back-end services include the following:
 - a) **Monitoring:** to permits the users to access the real node and interrupter after end their sessions.
 - b) **Disk Imaging:** To enable users to manage their disk image with command such as load, save, and reset nodes.
 - c) **OMF Manager:** The default service of manage the experiment running and execution on the testbeds.

D. Federation API

In order to design this integration, Its need to develop an interconnection services to establish the integration between federation services and testbed. The best practice to integrate between two modules is to design API that connect between two databases. Figure 3 shows the minimum database tables that federation API required to access in order to access/control portal data. The following are the list of the tables with short description:

- 1) **Resources:** stores information about local site resources.
- 2) **Users:** stores information about local users access.

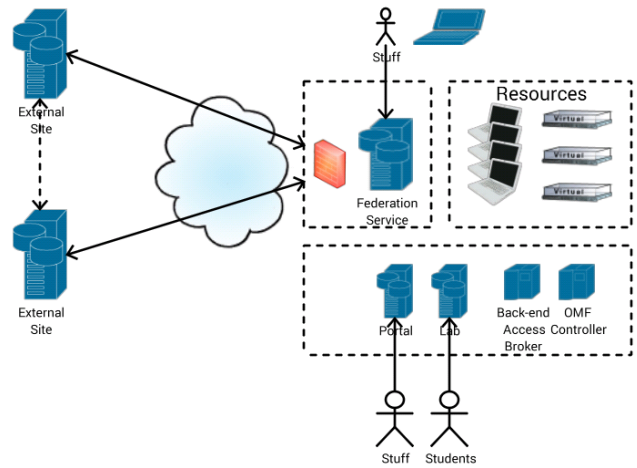


Fig. 2. Federation system architecture design including testbed portal.

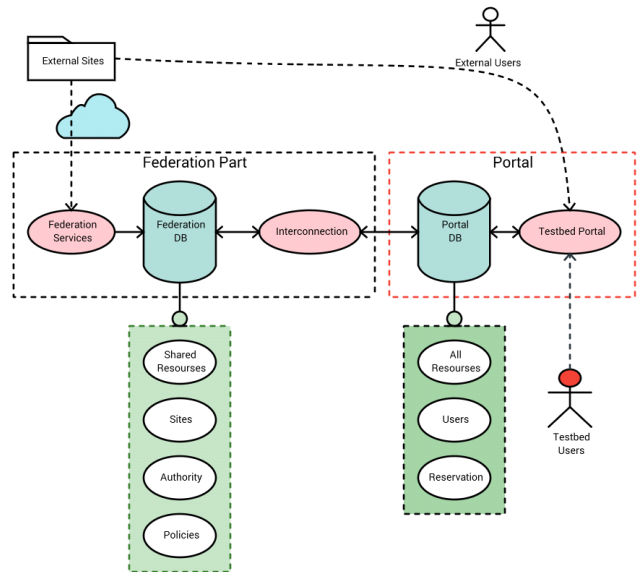


Fig. 3. Federation components design

- 3) **Reservation:** stores information about user reservation time and node access time.

Using the above list of tables, the federation services could read existing resources, reservations. and users to handle the federation services. The following is the possible operations the federation API may handle:

- 1) Check the local user access.
- 2) Check the resources status.
- 3) Check the reservation availability.
- 4) Change the resource status.
- 5) Add/Remove external user reservation.
- 6) Add/Remove local resources.

E. System design

F. Federation System

Based on the design goal, the following block diagram (Figure 4) below shows the principal layers of the federation system and their interactions.

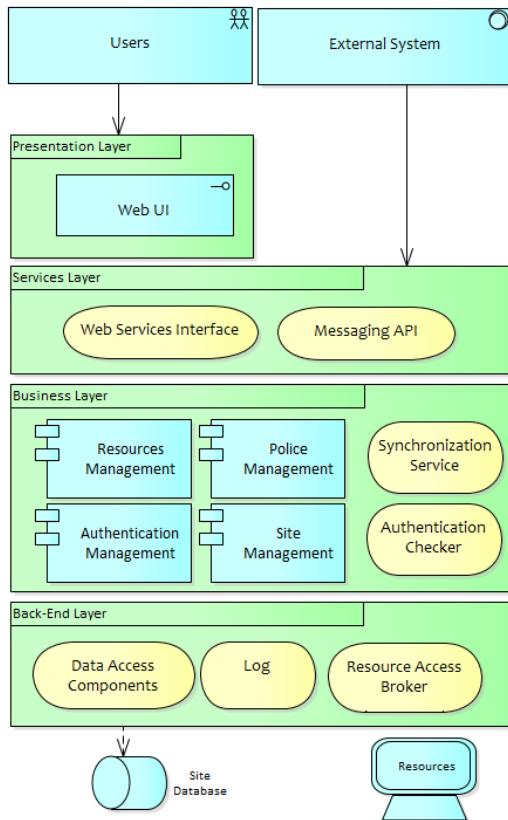


Fig. 4. Federation Architecture Layers Diagram.

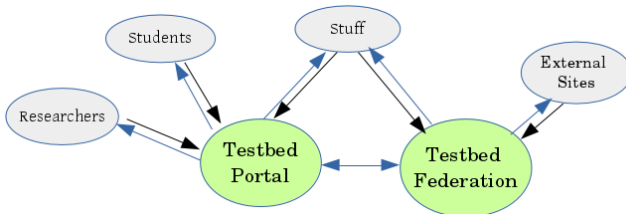


Fig. 5. Context diagram for Federation Service.

1) *Federation Components*: To implement the our federation system, we divide the it to four sub-systems as the following:

- Site Management
 - Site configuration.
 - Site registration.
 - Sites controls.
- Resources Management
 - Add resources.
 - setting sharing policy.
 - control resources.
- Authentication Management: to management key authentication between sites.
- Policy Management: to Manage policy for resources.

The context diagram (Figure 5) shows the main actors interacting with the system.

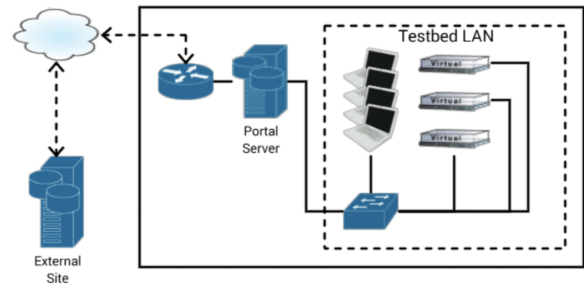


Fig. 6. Network Infrastructure for an Operating Testbed

VI. UNIFIED PORTAL

Along with federation, we have been design a new portal to serve as a single point of access to any testbed that is part of the CRC consortium. Users registered in these testbeds can use their credentials to access the unified portal and perform their experiments on the federated testbeds. The main idea is to use a suitable framework to handle the queries across multiple sites and retrieve the updated information. So, In order to synchronize data between unified portal and other sites, the database synchronization calls occur during the following main user interactions only:

- **User Login**: to grant users access across all testbeds and verify the credentials.
- **List Resources**: to list all the resources currently available in the federation.
- **Reservation Check**: to check if the current reservation is valid and doesn't overlap any existing reservation.
- **Perform Reservation**: to update the tables with the current reservation and mark the resources as busy during that time.
- **Cancel Reservation**: to release the resources during the cancelled reservation and mark them as available.

During each of these calls, system updates the lists with the query results obtained from all the federation databases. In case any query fails to execute, the responsible site will be considered as offline and the rest of the queries resume the execution.

VII. SYSTEM REQUIREMENTS

Next, we focus on the federation between multiple testbeds and how they can communicate with one another. First, a number of requirements must be present for any testbed to be able to federate with the current CRC consortium.

A. Hardware Requirements

- **Public IP**: provides a public access to web and act as identifier for the testbed.
- **Portal Controller Server**: a web-hosting powerful device to manage the testbed and administrate the network.
- **Testbed Nodes**: connected testbed devices and resources.

B. Network Requirements

- **Router:** provides access to the public internet.
- **Switch:** handles the local area network to connect the controller with the testbed resources.
- **Wireless/Ethernet Connections:** connects the controller server with the router.

Fig. 6 describes the testbed connection. A Local area network connects the testbed controller server running the portal with the available resources. Moreover, the controller is connected to a router and assigned a fixed IP Address to communicate with other testbeds and become remotely accessible.

C. Hardware Requirements

- **Operating System:** Linux based (Ubuntu, Debian, CentOS, ..). However, the current testbed was only tested using Debian Jessie (8.0)
- **Python:** Python v3.x for the front-end and Python v2.x back-end applications.
- **Python Libraries:** a list of python libraries are required to run the portal application and can be found in the testbed setup document.
- **Django:** Front-end framework (currently version 2.1.1). Used to build the portal from source and connect the front-end interfaces with the database models.
- **MySQL:** A database management system to store the testbed entities and reservations.

D. Application Requirements

- **CRC Portal:** provides a GUI access to the testbed functions and management.
- **CRC Back-end Services:** provides different portal services to control the connected resources.
- **OMF Experiment Control:** provides a common framework to describe, organize, run and retrieve results of an experiment using heterogeneous resources.

Once these requirements are complete, the newly created testbed can take part of the CRC federation. The next step is to submit a request for the CRC administrator with the full details of the testbed. Once confirmed, the public IP of the testbed is used to gain access to the MySQL database. A number of ports must be open for the testbed to communicate with the CRC federation through API calls and channels. We currently only require the default MySQL port (3306) to be open and accessible remotely in order for the unified portal to gain access to the testbed.

VIII. CHALLENGES AND FUTURE WORK

One of the main challenges in our design is to ensure that concurrency is maintained during the reservations. Therefore, we plan to conduct extended testing to make sure that no violations occur. Additionally, experiment control must be able to run in a centralized manner by creating an OMF script that can access all the federation resources as if they were attached to a single testbed. Along with implementation, some of new challenges appeared which may be addressed in futures. The main handling challenges are: Handling the

same user key over multiple sites; which may break the security challenges; Handling multi-site reservation cancellation in case of delay confirmation in some sites or superuser site cancellation; and Handling of multi-site code execution. Finally, the federation with other global consortium will be considered to promote the current testbed and provide more benefits for the researchers in terms of accessing global resources.

IX. IMPLEMENTATION TECHNOLOGY USED

The both portal and federation system is implemented using Django framework (Version 1.2.1, <http://www.djangoproject.com/>), The Front-end applications supported with JavaScript. The Back-end services Programming using Python.

X. CONCLUSIONS

In this paper, we have highlighted the main benefits of the new federation design, and architecture. we provided a brief description of components, layers and technologies used in project. Finally, present a simple design for the unified portal.

ACKNOWLEDGMENT

This research project was supported by National Telecommunication Regulation Authority (NTRA) under CRC: Collaborative Research and Teaching Testbed for Wireless Communications and Networks-Phase II. We thank for Prof. Mustafa ElNainay, who provided insight and expertise that greatly assisted the research.

REFERENCES

- [1] Dipankar Raychaudhuri, Ivan Seskar, Max Ott, Sachin Ganu, Kishore Ramachandran, Haris Kremono, Robert Siracusa, Hang Liu, and Manpreet Singh. Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 3, pages 1664–1669. IEEE, 2005.
- [2] J. Gaedde B. Hilburn T. Bose T. R. Newman, A. He and J. H. Reed. Virginia tech cognitive radio network testbed and open source cognitive radio framework. In *Proceedings of 2009 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops (TridentCom)*, pages 1–3, 2009.
- [3] Katerina Pechlivanidou, Kostas Katsalis, Ioannis Igoumenos, Dimitrios Katsaros, Thanasis Korakis, and Leandros Tassioulas. Nitos testbed: A cloud based wireless experimentation facility. In *Teletraffic Congress (ITC), 2014 26th International*, pages 1–6. IEEE, 2014.
- [4] Samer S Hanna, Arsany Guirguis, Mahmoud A Mahdi, Yaser A El-Nakieb, Mahmoud Alaa Eldin, and Dina M Saber. Crc: collaborative research and teaching testbed for wireless communications and networks. In *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*, pages 73–80. ACM, 2016.
- [5] Paul Müeller and Stefan Fischer. Europes mission in next-generation networking with special emphasis on the german-lab project. In *The GENI Book*, pages 513–544. Springer, 2016.
- [6] Wim Vandenberghe, Brecht Vermeulen, Piet Demeester, Alexander Willner, Symeon Papavassiliou, Anastasios Gavras, Michael Sioutis, Alina Quereilhac, Yahya Al-Hazmi, Felicia Lobillo, et al. Architecture for the heterogeneous federation of future internet experimentation facilities. In *Future Network and Mobile Summit (FutureNetworkSummit), 2013*, pages 1–11. IEEE, 2013.