

Leveraging Online Lab Development: A New Paradigm to Offer Educational Lab Infrastructure as a Cloud Service

Danilo Garbi Zutin, Carinthia University of Applied Sciences

Danilo G. Zutin is currently a Senior Researcher and team member of the Center of Competence in Online Laboratories and Open Learning (CCOL) at the Carinthia University of Applied Sciences (CUAS), Villach, Austria, where he has been engaged in projects for the development of online laboratories, software architectures for online laboratories and online engineering in general. Danilo is author or co-author of more than 30 scientific papers published in international journals, magazines and conferences. Most of these papers are in the field of online engineering, remote and virtual laboratories and issues associated with their dissemination and usage.

Prof. Michael E. Auer, Carinthia University of Applied Sciences

Dr. (mult.) Michael E. Auer is Professor of Electrical Engineering at the Faculty of Engineering and IT of the Carinthia University of Applied Sciences Villach, Austria and has also a teaching position at the University of Klagenfurt. He is a senior member of IEEE and member of ASEE, IGIP, etc., author or co-author of more than 170 publications and leading member of numerous national and international organizations in the field of Online Technologies. His current research is directed to technology enhanced learning and remote working environments especially in engineering. Michael Auer is Founding-President and CEO of the "International Association of Online Engineering" (IAOE) since 2006, a non-governmental organization that promotes the vision of new engineering working environments worldwide. In 2009 he was appointed as member of the Advisory Board of the European Learning Industry Group (ELIG). Furthermore he is chair of the Advisory Board of the International E-Learning Association (IELA) and member of the Board of Consultants of the International Centre for Innovation in Education (ICIE). In September 2010 he was elected as President of the "International Society of Engineering Education" (IGIP, <http://www.igip.org>). Furthermore he is one of the founders and Secretary General of the "Global Online Laboratory Consortium" (GOLC). GOLC is the result of an initiative started in 2009 at MIT to coordinate the work on educational online laboratories worldwide.

Leveraging Online Lab Development: A new Paradigm to Offer Educational Lab Infrastructure as a Cloud Service

Abstract

In this work we propose a new paradigm to deliver laboratory server infrastructure as a service (LIaaS). The main difference if compared with other traditional LaaS (laboratory as a service) paradigms concerns the service consumers. Traditional LaaS usually focuses in creating and enabling services for end users of the online laboratory (students, teachers), but a laboratory server infrastructure as a service (proposed with this work) offers services to be consumed by laboratory owners and developers aiming to facilitate its development and deployment.

Introduction

As pointed out by many authors, the Internet changed the way distance education has been carried out. The enhanced delivery of educational services through technological innovations and socio-economic factors, like the need for part-time study programs and continuing education have completely redefined the physical and temporal boundaries that education and laboratory experimentation have been subjected to¹. This new scenario placed online laboratories as part of a much larger picture with the potential to deliver education at any time and place, not only to students who could not come to the campus, but also as a tool that could potentially enhance laboratory experience of on-campus students as well². Additionally, the effects of the use of online laboratories and their influence on the learning outcomes have been studied by several authors³.

However, developing an online laboratory from scratch can still be a complex task, especially if the developer is a specialist in the laboratory domain and not a software engineer. Bringing a piece of equipment online requires high technical, and administrative efforts to ensure a secure setup and continuous maintenance of the system. For example, one should ensure proper ICT infrastructure, proper system security, intensely collaborate with institution's IT department, among others.

Background

The development of online laboratories has undergone major changes since the first systems were introduced almost 15 years ago. In the beginning, online laboratories were developed as ad hoc solutions, usually designed for a very specific purpose and were not intended to be adapted or generalized, making online laboratories a closed, self-contained system. Fortunately the community soon realised that this strategy did not favor the large-scale use and scalability of online laboratory systems in formal education and research. The initial attempts to address some of these issues began with the development of the first Remote Laboratory Management Systems (RLMS), such as the iLab Shared Architecture⁴ and WebLab Deusto⁵, that took place mainly during the last decade. RLMSs grouped common functionalities around a single framework. Some of the initial functional requirements of these RLMSs were:

- User management, single sign on with institution's authentication systems
- Implementation of laboratory scheduling services

- Support for a scalable federation of online laboratories
- Support single sign-on in a cross-institutional federation of online laboratories
- Support for management of experiment data (data storage and retrieval)
- Integration with Learning Management Systems n (LMS)

With RLMSs it became easier to share online laboratories among institutions that deployed the same RLMS, on the other hand, sharing these systems across different RLMSs was not supported, as the APIs and interfaces used to communicate different components (e.g. laboratory clients and servers, booking services, experiment data retrieval) were not compatible.

From the laboratory development point of view, the APIs shared between labs developed around one RLMS indeed contributed to simplify online lab development and deployment, assuming one had all necessary skills and was familiar with the RLMS API. Some of the typical tasks that define the profile of an online laboratory developer are:

- Knowledge on different Internet technologies available
- Knowledge on service-oriented architectures and Web services (SOAP, REST, etc)
- Design a client-Server message schema (JSON, XML, etc)
- Provide proper ICT infrastructure at the institution's networks
- Ensure proper system security
- Familiarity with the particular RLMS API

The tasks for lab developers vary and depend strongly on the RLMS, but these requirements usually always involve developing software to communicate with the APIs that expose the systems' functionalities. However, laboratory developers are usually specialists in the laboratory domain, and in most cases do not possess these mentioned necessary skills to develop, deploy and maintain these labs. For software developers, RLMSs did add some extra level of abstraction in the laboratory development process, but this process could be even more abstracted in order to make the development of new labs and their integration with different RLMS APIs as simple as possible.

At the present moment, three different RLMS APIs are supported in the context of this work, namely the original iLab Shared Architecture (ISA) SOAP API from MIT⁴, the ISA JSON API from University of Queensland, Australia⁶ and the API of WebLab Deusto⁵.

The Online Laboratory Infrastructure as a Service Paradigm

The paradigm of online laboratory delivered as a service (LaaS) is not new and has been proposed and implemented by other authors like⁷, that proposes a model to deliver modular remote laboratories based on loosely coupled services. This approach is in line with the work carried out by⁸ in the scope of the Go-Lab Project (www.go-lab-project.eu). In the LaaS (Laboratory as a Service) model, laboratories are delivered as a set of loosely coupled services that expose the functionalities of a piece of equipment. One of the main characteristics of this approach is the decoupling of user interface and laboratory server code. It allows a laboratory server to be consumed by different UIs that can be adapted for different classes and learning objectives.

The approach introduced with this work addresses a different issue, namely the availability of laboratory infrastructure and how to offer this infrastructure as a set of services that can be

used by laboratory owners and developers to develop and deploy an online laboratory independently of its domain. The proposed Laboratory Infrastructure as a Service (LIaaS) unites characteristics of all three most common cloud delivery models, namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

By shifting the core functionalities of online laboratory server to a central location and by allowing its seamless reuse by heterogeneous online laboratories, it is possible to abstract the development of the software necessary to deliver remote experimentation. LIaaS consumers are laboratory developers and lab owners and not the laboratory final user (students, teachers and researchers). The main motivation for the implementation of this model is to abstract the the complexities posed to the lab owners to developed and deploy a new online laboratory. These complexities and how they were addressed are described below.

- **Deployment of proper ICT infrastructure:** We introduce an online laboratory infrastructure as a service (LIaaS) paradigm that shifts part of the deployment of a traditional batched laboratory server to the cloud and exposes its functionalities via a simple set of well defined services.
- **NAT (Network address translation) traversal:** NAT enables several devices in a local network to share a single IP address. Most universities and private networks use NAT to have all their devices connected to the Internet. This technique helped us to overcome the scarceness of public IP addresses, but on the other and it makes more complicated to initiate a connection to a device residing behind a NAT. Using the developed software framework a device can reside anywhere on the Internet.
- **Network Security (e.g. Firewalls):** A simple set of services is provided over HTTP protocol using a RESTfull architectural style, allowing it to bypass firewalls and proxy servers.
- **Integration with RLMSs:** Laboratory developers have to be familiar with the RLMS APIs of their institution in order to deploy the lab for their students. The framework should abstract the implementation of new labs and support well known RLMSs.

Fig. 01 depicts the cloud delivery model for online laboratories proposed in the scope of this work. It delivers several IT-Resources that are essential for the deployment of online laboratories, such as Lab Server provisioning, experiment scheduler, experiment data storage, integration with Remote Laboratory Management Systems (RLMS) and relaying of experiment data to the on-premises laboratory equipment. It additionally implements the APIs of well known remote laboratory management systems that deliver the laboratory client and the services supporting learning analytics and integration with learning management systems.

To validate the proposed LIaaS cloud delivery model, a prototype was implemented. The developed software framework was designed to be a tool for laboratory developers. It is not intended to replace any Learning Management System and Remote Laboratory Management System, but rather provides additional services to facilitate the development and deployment of new educational online laboratories and integrate them seamlessly with well known RLMSs. The main requirements for the development of the software framework were proposed in previous works^{9, 10}.

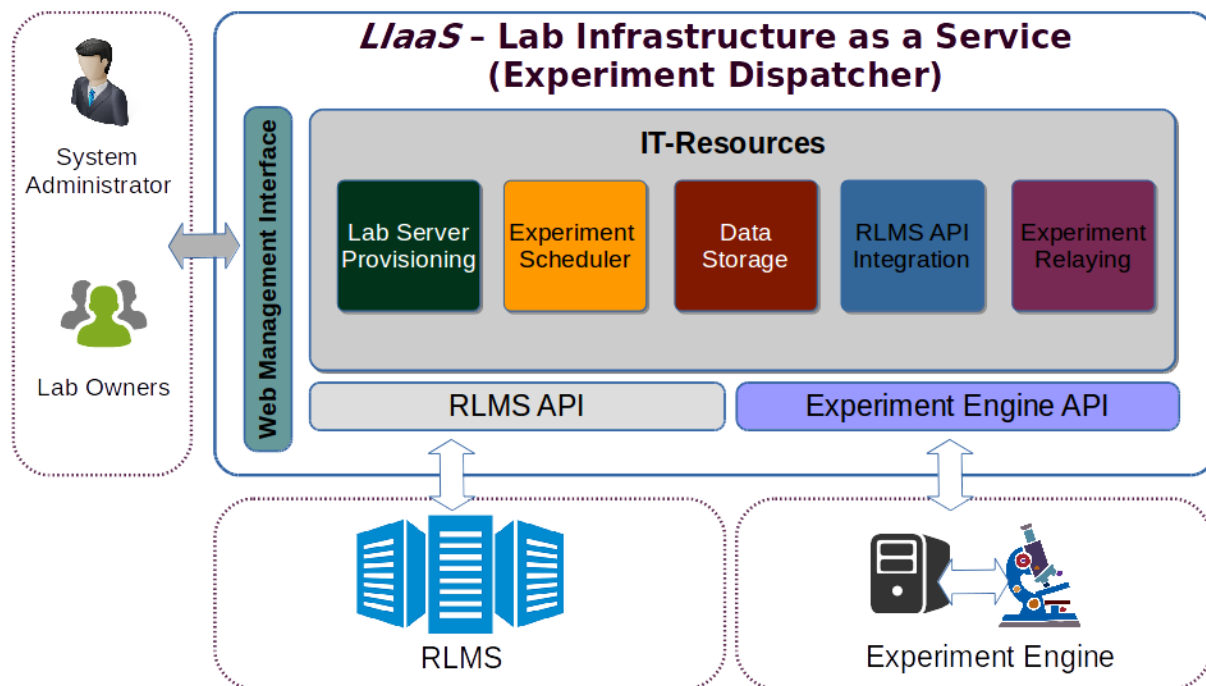


Figure 01. LlaaS software framework

The Experiment Dispatcher

The Experiment Dispatcher is the software framework that implements the proposed cloud model. It abstracts the setup of a laboratory server, allowing the lab owner to deploy several instances of virtual lab servers. The Experiment Dispatcher is not an RLMS. It implements however the API of some well known RLMS like the iLab Shared Architecture and WebLab Deusto that enables it to act as a federated node for these systems by serving up experiment execution requests. For asynchronous online laboratories, it implements also the lab scheduler (experiment queuing mechanism) and allows the lab owner to monitor the status of all experiments she/he owns. The Experiment Dispatcher provides two types of Web services interfaces, one to communicate and receive requests from the supported RLMSs and a second one used by the experiment engines to execute experiments.

The Experiment Engines

They are a laboratory specific part of the architecture. An engine dequeues experiments, executes them and sends the experiment results to the Experiment Dispatcher for later retrieval by the RLMS. The Experiment Dispatcher acts as a rendezvous server that relays messages to the Experiment Engines. The communication with the Dispatcher is always initiated by the Experiment Engine that performs a long polling while it waits for a new experiment request. An engine runs on the lab owner's side (eg. campus) since it must interface with the laboratory hardware. This implementation allows the Experiment Engines to reside anywhere on the Internet, even in a private network (e.g. behind a NAT and protected by a firewall).

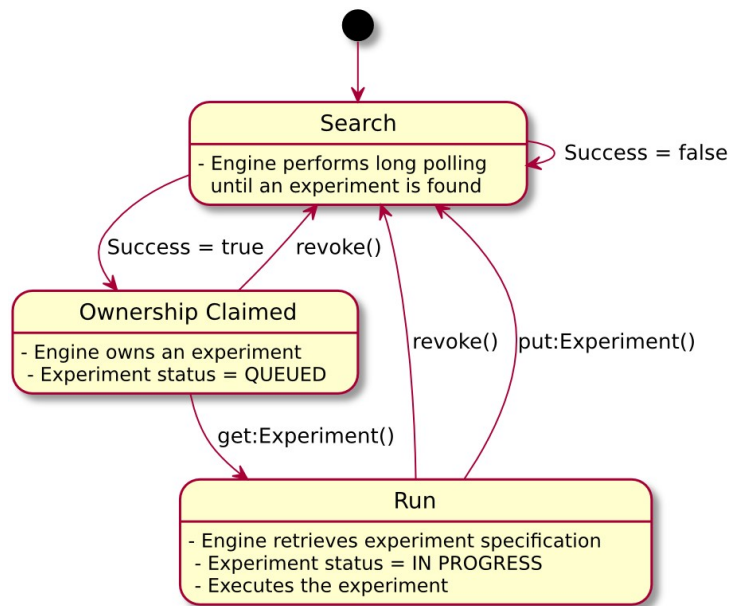


Figure 02. The Experiment Engine's state machine behavior

The Experiment Engine behavior can be described by the state machine depicted in Fig. 02. It has three finite states. It is possible because it executes asynchronous (or batched) experiments. These experiments can have their entire execution course specified before it starts. The task of performing a batched experiment can be summarized in submitting an experiment specification, executing the experiment, retrieving, analysing and displaying the results to the user. The parameters cannot be changed after the experiment was submitted and the results cannot be observed in real time. Changing one parameters requires the submission of the complete batch. In this sense, batches are completely independent from each other, what makes it a stateless model. The finite states are described as follows.

- **SEARCH:** This state can also be interpreted as the Idle state. In this state the Experiment Engine performs long polling on the Experiment Dispatcher to check on the availability of a new experiment. If an new experiment exists, the engine goes to the next state (CLAIM).
- **CLAIM:** In this state the Experiment Engine claims ownership over the experiment found during the last execution of the SEARCH state. This causes the experiment to be locked, meaning that other engines subscribing for the same online laboratory cannot claim ownership over this particular experiment. From this state the engine can either release the experiment by revoking the ownership and therefore return to the SEARCH state or run it.
- **RUN:** In this state the experiment is executed and the results are generated. If the experiment is canceled, the engine can revoke the ownership and return to the SEARCH state. If the experiment is successfully executed, the results are sent to Experiment Dispatcher and it returns to the SEARCH state.

Conclusion

The proposed framework to deliver remote laboratory infrastructure as a service was not designed to be an RLMS, but rather to serve up experiment requests from RLMS users.

therefore, the direct beneficiaries of the proposed Framework will be the lab server developers. Laboratory developers are specialists in the laboratory domain and are most likely not software engineers. Therefore, adapting an existing lab or developing a new one from scratch in compliance with an RLMS might require advanced software development skills.

The technical model introduced in this work has served as the basis for a project proposal submitted to the Austrian Ministry of Science and Research in the scope of the Sparkling Science research grant Programme (BMWFW, 2015). The proposal was accepted in July 2014 and the project was officially launched in October 2014 and will end by October 2016. By the time of writing, six online laboratories are being deployed with the Experiment dispatcher and several experiment engines have been developed in different platforms to serve as templates for lab owners. Some of these engines were developed by users who were not specialists in online laboratory development, but the laboratories were successfully implemented.

As previously mentioned, the developed frameworks supports only asynchronous laboratories. Future developments will include support for interactive asynchronous laboratories as well, however, this requires support for a real-time or soft real time message relaying mechanism between the RLMS and an Experiment Engine passing through the Experiment Dispatcher taking into account the same original requirements (ability to surpass NAT, firewalls, proxy servers). This new requirements makes relaying not the most suitable method due to its inefficiency, as pointed out in ¹¹. Delivering real time or soft real time message over an overlay network is not a new concept, in fact, this is the core of any peer-to-peer network, like VoIP telephony. An application of this concept to the online laboratory domain has been the subject of other works¹². We have been studying the possibilities and the most suitable technologies available to implement this. A potential candidate are WebSockets¹³, as described in RFC 6455 due to the support for a bidirectional full-duplex communication channel that could be used to convey messages to the experiment engines⁷.

We would like to highlight that the solution presented by this work is a tool for laboratory developers or laboratory owners that plan to or already host online laboratories. These stakeholders are usually specialists on the lab domain and not software engineers. The tools presented here are, however, transparent for other stakeholders such as students, instructors and teachers. It is not the focus of this work to provide any guidelines or recommendations concerning the pedagogy associated with the laboratory experiment. These aspects are left to the discretion of the developer. The ultimate aim of this work is to simplify the development and deployment of online laboratories by minimizing the requirement for laboratory owners and developers when creating and deploying a new laboratory, and therefore allowing them to focus on pedagogical aspects involved to bring students the best online lab experience.

References

1. Auer, M., Edwards, A., and Garbi Zutin, D. (2011). Online laboratories in interactive mobile learning environments. In N. Pachler, C. Pimmer, and J. Seipold, editors, *Work-based mobile learning: concepts and cases*. Peter Lang, Oxford ; New York.
2. L. D. Feisel and A. J. Rosa, "The role of the laboratory in undergraduate engineering education," *Journal of Engineering Education*, vol. 94, no. 1, pp. 121–130, 2005.
3. E.D. Lindsay and M.C. Good, "Effects of Laboratory Access Modes upon Learning Outcomes," *IEEE Trans. Education*, vol. 48, no. 4, pp. 619-631, Nov. 2005.

4. V. J. Harward, J. A. del Alamo, S. R. Lerman P. H. Bailey, J. Carpenter, et. al., "The iLab Shared Architecture: A Web Services Infrastructure to Build Communities of Internet Accessible Laboratories," Proceedings of the IEEE , vol.96, no.6, pp.931-950, June 2008.
5. P. Orduña, "Transitive and Scalable Federation Model for Remote Laboratories," Doctoral Thesis, Univerity of Deusto, Bilbao, Spain, Apr. 2013. [Online]. Available: <http://morelab.deusto.es/media/publications/theses/pablo-orduna.pdf>
6. S. Colbran and M. Schulz, "An update to the software architecture of the iLab Service Broker." IEEE, Feb. 2015, pp. 90–93. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7087269>
7. Tawfik, Mohamed, Christophe Salzmann, Denis Gillet, David Lowe, Hamadou Saliah-Hassane, Elio Sancristobal, and Manuel Castro. "Laboratory as a Service (LaaS): A Novel Paradigm for Developing and Implementing Modular Remote Laboratories." *International Journal of Online Engineering (iJOE)* 10, no. 4 (June 8, 2014): 13. doi:10.3991/ijoe.v10i4.3654.
8. C. Salzmann, S. Govaerts, W. Halimi, and D. Gillet, "The Smart Device specification for remote labs." IEEE, Feb. 2015, pp. 199–208. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7087292>
9. Zutin, D.G., Auer, M., 2014. A LabVIEW based experiment execution engine to ease the development of ISA batch lab servers. IEEE, pp. 1089–1092. doi:10.1109/EDUCON.2014.6826244
10. Zutin, Danilo, and Michael E. Auer. "A Toolkit to Facilitate the Development and Use of Educational Online Lab-Oratories in Secondary Schools." In *American Society for Engineering Education Conference, 2015*, 26:1. Accessed January 21, 2016.
11. Kegel, Dan, Pyda Srisuresh, and Bryan Ford. "State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs)." Accessed January 21, 2016. <https://tools.ietf.org/html/rfc5128>.
12. A. A. Kist, A. Maiti, A. D. Maxwell, L. Orwin, W. Midgley, K. Noble, and W. Ting, "Overlay network architectures for peer-to-peer Remote Access Laboratories." IEEE, Feb. 2014, pp. 274–280. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6784274>
13. I. Fette and A. Melnikov, "The WebSocket Protocol - RFC 6455," Dec. 2011. [Online]. Available: <https://tools.ietf.org/html/rfc6455>