

A Cloud-based Architecture for Remote Access to Digital Fabrication Services for Education

Gianluca Cornetta, Abdellah Touhafi, *Member, IEEE*, F. Javier Mateos and Gabriel-Miro Muntean, *Member, IEEE*,

Abstract—Fabrication-as-a-Service (FaaS) represents a method and a set of innovative solutions for the next generation fabrication laboratories (Fab Labs). It leverages cloud and IoT technologies to enable a wide learning community to have remote access to these labs' computer-controlled tools and equipment over the Internet. This paper introduces FaaS in the context of Fab Lab challenges and describes FaaS instantiation in NEWTON Fab Labs, part of an European Horizon 2020 project. The NEWTON Fab Labs architecture is described in details with a major focus on the communication protocol stack. The system has been deployed and a test scenario that simulates a real user behavior has been set-up in order to stress system performance and measure the system response time in different operating conditions.

Index Terms—Fabrication-as-a-Service, Remote education, Internet of Things, Machine to Machine Communication.

I. INTRODUCTION

LATELY formal education, vocational training and lifelong learning play an increasingly important role in society. These are seen not only as means to providing benefits in terms of enabling future economic development, but also ways to offering people support for knowledge gathering, skill acquiring, personal development, and better cultural understanding. At any moment, worldwide, millions of citizens of all ages benefit from diverse forms of education. This education is mostly *formal* (i.e. in schools, universities), but also *non-formal* (i.e. outside the education system) and *informal* (i.e. individuals are responsible for their own education). However, regardless of education type, it is a recognized fact that most developed countries are currently experiencing a negative trend in terms of their young generation interest to pursue scientific education. For instance in Europe alone, the proportion of graduates specializing in science, technology (e.g. computing), engineering and mathematics (STEM) has reduced from 12% to 9% since 2000 [1] and consequently Europe faces a concrete shortage of scientists.

There is strong evidence that many young people disengagement from the STEM area starts during secondary education [2]. The disengagement is mainly due to two factors:

The work described in this paper is part of the NEWTON project, which has been funded by the European Union under the Horizon 2020 Research and Innovation Programme with Grant Agreement no. 688503.

G. Cornetta is with San Pablo-CEU University, Madrid, Spain, e-mail: gcornetta.eps@ceu.es

A. Touhafi is with Vrije University, Bruxelles, Belgium, e-mail: abdelah.touhafi@vub.ac.be

G.-M. Muntean is with School of Electronic Engineering, Dublin City University, Dublin, Ireland, e-mail: gabriel.muntean@dcu.ie

F.J. Mateos is with San Pablo-CEU University, Madrid, Spain, e-mail: javier.mateossecano@ceu.es

Manuscript received July 1, 2018.

students' perceive scientific subjects as difficult, and they regard science-related careers as little attractive in terms of job quality-pay level balance. Many efforts are put worldwide trying to reverse this process, including part of large European Union projects such as NEWTON¹. However, most interesting and effective approaches involve acting as early as possible (e.g. secondary schools) and making use of the latest innovative technologies and solutions. Using Fabrication Laboratories or Fab Labs [3] is an innovative solution which is both attractive for students and highly effective in terms of STEM teaching and learning.

A Fab Lab is a small-scale workshop with a set of flexible computer-controlled tools and machines such as 3D printers, laser cutters, computer numerically-controlled (CNC) machines, printed circuit board millers and other basic fabrication tools which, usually, are not easily accessible. It is the perfect place for the instantiation of "learning by doing" because all the tools to bring a product to realization are within reach and the users get to take part in all the phases of the fabrication process. This is why a Fab Lab attracts students as they can experiment and materialize their ideas in engaging and stimulating ways. A limitation related to the current use of these labs is that they are not available remotely over the Internet, limiting their accessibility and their wide use.

This paper introduces Fabrication-as-a-Service (FaaS), an innovative method and associated solutions to enable remote access to Fab Labs as a Cloud-based service. This approach is a necessary evolution of Fab Labs, allowing them to become available to a wider community over the Internet. FaaS opens new opportunities by providing students with the means to perform design and testing in a remote experimental environment and come up with innovative solutions to concrete problems. FaaS is seen as an integral part of the 21st century teaching and learning paradigm, is dynamic and student-centric, and helps attract students to STEM education. The paper instantiates the proposed FaaS concept in form of NEWTON Fab Labs and describes the underlying architecture, technologies and the communication protocol stack employed.

The paper is structured as follows: current Fab Lab deployments and associated technologies are discussed in section II, FaaS NEWTON Fab Labs deployment and architecture are presented in section III and the communication protocol stack is described in section IV. The current status of FaaS NEWTON Fab Labs deployment and paper conclusions and future work directions are presented in sections V and VI.

¹NEWTON Website: <http://newtonproject.eu>

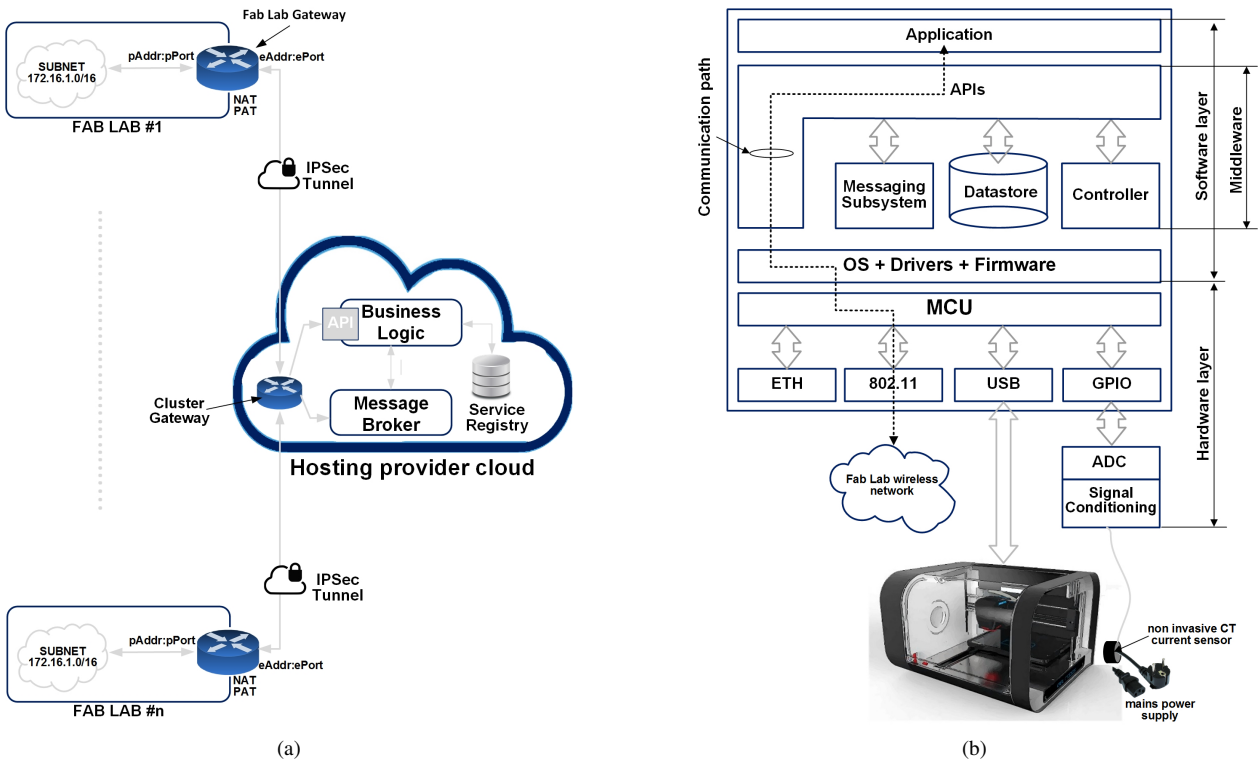


Fig. 1. NEWTON Fab Lab (a) Cloud Inter-networking Architecture, and (b) Fabrication Equipment Interfacing.

II. FAB LABS AND FAAS CHALLENGES

The Fab Lab concept is gaining worldwide interest and both governments and population are starting to recognize the importance of digital fabrication technologies even as early as primary and secondary level education². A direct consequence is that the number of Fab Labs is continuously increasing and to date there exists a worldwide network of more than 1.100 Fab Labs located in more than 40 countries, which are coordinated by the Fab Lab Foundation³.

Surprisingly, all the research efforts put to date in the digital fabrication area have been aimed at demonstrating the effectiveness of Fab Labs in education [5] and at incorporating digital fabrication in the curricula [6], [7], [8]. However, to the best of authors' knowledge no attempt has been made to address the challenges faced enhancing the Fab Lab functionality by providing support for pervasive and ubiquitous Internet access. The main factor that is actually limiting a wider diffusion of the Fab Lab concept is the lab set up cost. Fabrication machines and materials are expensive and not all educational institutions, especially in primary and secondary education streams, may afford the costs to start and especially maintain a Fab Lab. Providing a Fab Lab with ubiquitous access is not simply a matter of networking the digital fabrication equipment, but a challenging task that entails rethinking the whole software and hardware infrastructure and involves

the design of an ad-hoc communication stack to manage real time access and control of the networked equipment and address all security issues that might arise by exposing the equipment to the Internet. We call this complex evolution of the Fab Lab concept *Fabrication as a Service* (FaaS). This approach allows remote network access to fabrication services and addresses all software and logistic concerns including interaction, communication, and security. FaaS addresses a real social and economic need and arises from the application of the technological paradigms behind the Internet of Things and Industry 4.0 in the Fab Lab context.

III. NEWTON FAB LAB ARCHITECTURAL DESCRIPTION

As already mentioned, FaaS is a natural evolution of Fab Labs, which are small scale digital fabrication facilities equipped with computer controlled tools. One major limitation of current Fab Labs is their lack of external connectivity and infrastructure flexibility, requiring constant human supervision to carry out any given task. This section describes the NEWTON Fab Lab architecture, which overcomes these intrinsic limitations. NEWTON Fab Labs design and deployment are part of the NEWTON European Horizon 2020 project⁴ and instantiates the proposed FaaS approach in a concrete real life lab context. The architecture provides Fab Labs with an abstraction layer that wraps the underlying hardware infrastructure into a programmatic interface which consists of a set of Application Programming Interfaces (API) that offer the Fab Lab as a Web service to third-party applications.

²"National Curriculum in England: Design and Technology Programmes of Study", UK Department for Education, 2013, <http://www.gov.uk/government/publication/national-curriculum-in-england-design-and-technology-programmes-of-study>

³Fab Lab Foundation Website: <http://www.fabfoundation.org/>

⁴EU-funded NEWTON Project Website: <http://newtonproject.eu>

The following functions are performed by the API:

- 1) Fab Lab equipment remote control and configuration;
- 2) Inter-Fab Lab communication and task synchronization;
- 3) Intra-Fab Lab communication and task synchronization.

These features allow remote monitoring and automatic synchronization of the machines involved in a fabrication batch with minimum human intervention, and enable support for new scenarios in which any complex design can be implemented in a distributed fashion by splitting it among several networked Fab Labs. Moreover, the API approach allows the possibility to develop Web interfaces allowing remote access to fabrication resources, and thus enabling implementation of distance-learning courses and Ed-to-Ed (i.e. Education-to-Education) scenarios in which partner institutions share expensive fabrication equipment for teaching and research.

Fab Lab networking is very challenging and raises several issues related to system interfacing, scalability, security, quality of service, as well as real-time and non-blocking communications. The possibility to manage collaborative and distributed fabrication batches with little or no human supervision is a powerful feature, but also a potential source of harmful failures and physical damages due to design errors that could result in fabrication equipment malfunction. This in turn, entails the development of a monitoring software and hardware infrastructure to guarantee safe equipment operation and fault tolerance. Next is described how all of the aforementioned issues have been addressed by proposing a robust, scalable and inexpensive architecture.

A. Cloud-based Architecture

The NEWTON Fab Labs employ a two tier architecture composed of a hub and a network of distributed Fab Labs. Each Fab Lab interacts with the hub and other labs via a dedicated specialized node called Fab Lab Gateway, which will be described later on. The hub is deployed in the cloud. Although some novel Internet of Things architectural paradigm exists, in which some data processing is performed at the edge of the network by the devices themselves [9], we have preferred a standard approach with all the data to be processed directly in the cloud. This is as the amount of data to be processed (i.e. the Fab Lab status) is not large enough to justify an extra middleware layer between the Fab Labs and the Cloud hub.

The NEWTON Fab Labs are exposed as services to the Internet through REpresentational State Transfer (REST) APIs that provide Create, Read, Update and Delete (CRUD) methods to access the underlying data model as well as communication primitives to publish and subscribe to services over the Web. The Service Oriented Architecture (SOA) guarantees the inter-operability of the different system components, regardless of the implementation technology and allows easy system scalability. The service registry is held on cloud hosting premises and acts as a centralized communication hub among service providers and subscribers. The overall NEWTON Fab Lab system architecture is illustrated in Fig. 1a.

In order to allow inter-Fab Lab communication, each networked Fab Lab should have at least one public IP address

eAddr:ePort. The router/gateway maps the inbound traffic into a private address *pAddr:pPort* by means of a Network Address Table (NAT) and a Port Address Table (PAT). Similarly, the router performs the same task on the outbound traffic by forwarding it to the default gateway or by redirecting the requests for a private address to the private network. The message flow between the cloud application and the networked Fab Labs is managed by a cloud-deployed message broker that implement a publish/subscribe protocol.

B. Equipment Interfacing

Providing networking support for the Fab Lab is difficult because usually fabrication equipment is controlled through USB ports only. In addition, hardware specifications as well as device drivers are proprietary, which makes the development of low-level control and monitoring APIs difficult. However, NEWTON Fab Labs employ an innovative master-slave approach which enables monitoring and controlling the activity of expensive fabrication equipment through inexpensive external hardware as shown in Fig. 1b.

The "master" unit is basically an off-the-shelf micro-controller unit (MCU) with basic Ethernet (ETH) and wireless (802.11) connectivity, as well as an USB and a General Purpose Input Output (GPIO) port. The application is designed using an API-first approach and communicates with the "slave" digital fabrication equipment through the USB port of the host MCU and with the other Fab Lab networked equipment through the 802.11 interface. The status of the fabrication equipment is monitored by a non-invasive current transformer (CT) sensor that measures the mains AC current drawn by the equipment. The CT sensor is interfaced to the GPIO digital interface by a high-resolution analog-to-digital converter (ADC) and a simple signal conditioning circuit. This simple configuration allows fast detection of the equipment status (i.e switched-off, idle or busy); however, we also investigate a *deep learning approach* [10] to infer the machine behavior from the current drawn from the supply. This allows monitoring more complex behaviors of the fabrication equipment, detecting errors and even switching off the equipment in the case of potentially harmful commands.

C. Scalability Issues

The supporting hardware and software infrastructure must be carefully designed in order to enable scalability and high availability of the fabrication services, while keeping the bandwidth requirements as low as possible. This can be accomplished by:

- 1) a two-level parent-child broker architecture,
- 2) individual command pooling,
- 3) broker clustering.

As already mentioned, in a two-level architecture employed like the one presented in Fig. 3, the Fab Lab equipment can be only accessed through a Fab Lab Gateway. The Fab Lab Gateway decouples the centralized server from the fabrication equipment, controls the inbound traffic among the machines and the outbound traffic to the server in the cloud premises.

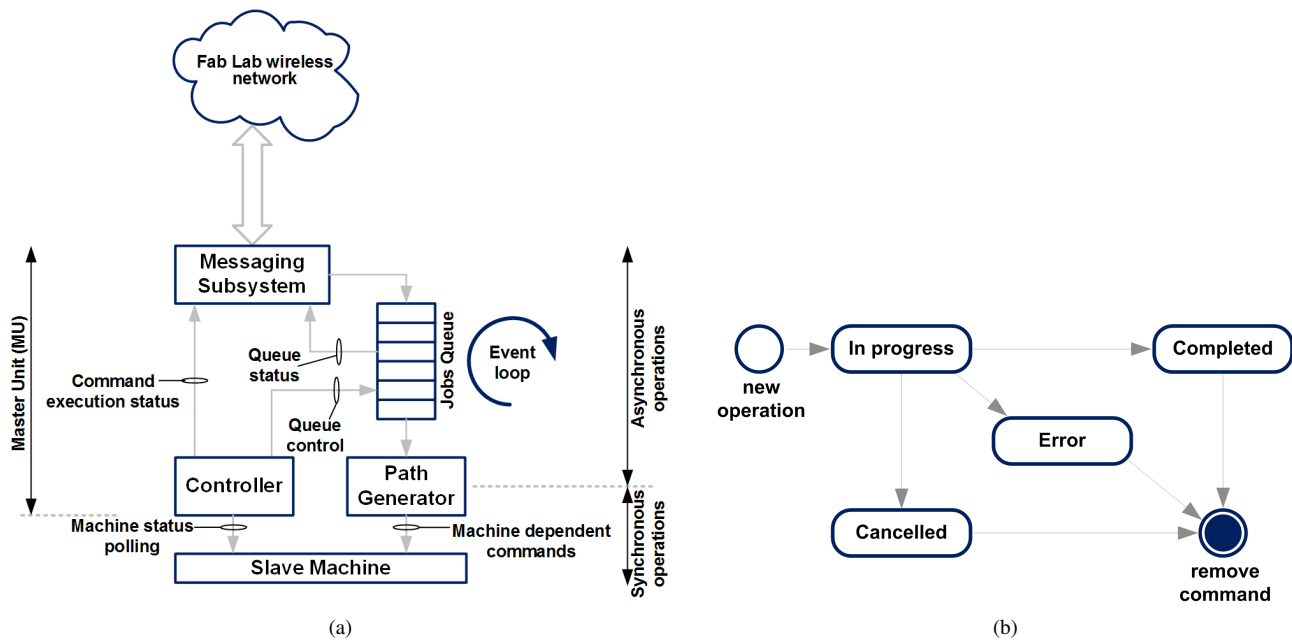


Fig. 2. Machine Software Wrapper. (a) Software Architecture, and (b) Machine State Controller.

The parent (on the cloud side) and child broker (on the Fab Lab Gateway) share the responsibility to route the end-to-end traffic between the client application and the remote fabrication machine. The centralized server in the cloud will particularly benefit from this approach since the number of publishing nodes will be limited to the Fab Lab Gateways and not to all the networked fabrication equipment. Moreover, the use of a pool of gateway-specific message queues instead of a unique centralized queue allows for the deployment of smart message passing protocols capable of routing the traffic only towards specific subscribers (for example according to geographic proximity) instead of continuously broadcasting the incoming messages to all the subscribers. Finally, clustering the parent broker by replicating the number of broker instances will provide high availability and better fault tolerance since the central server will not have a single point of failure.

D. Security Issues

Security in a distributed networked environment such as NEWTON Fab Labs is of paramount importance. Security issues are addressed at three different levels:

- 1) *Network level.* Encapsulating the network nodes into a Virtual Private Network (VPN) is an easy way to provide a trustworthy connectivity.
- 2) *Transport level.* In order to provide confidentiality, transport level encryption based on either Secure Socket Layer (SSL) or Transport Layer Security (TLS) is used.
- 3) *Application level.* The publish/subscribe protocol which NEWTON Fab Labs rely on provides a unique connection identifier which can be used to authenticate devices at application level. This protocol feature can be used by the broker to define authorization levels for each connected device or application.

Note, the transport level encryption may add a significant bandwidth overhead that, in some cases may not be affordable.

In such cases, payload encryption at the application level is a valid alternative to a full fledged transport encryption to transmit secure information.

E. Machine Wrapper and Command Interface

The NEWTON Fab Lab architecture provides the hardware, software and communication infrastructure to network several Fab Labs distributed over a wide geographical area, enabling several collaborative fabrication scenarios. The two-level broker architecture introduced previously decouples the communication into two categories:

- 1) *Inter-Fab Lab communications,* managed by the centralized broker on the cloud premises, and
- 2) *Intra-Fab Lab communications,* managed by the Fab Lab Gateway in the Fab Lab VPN.

Inter-Fab Lab communications determine the outbound traffic of the Fab Lab network, whereas Intra-Fab Lab communications determine the local network inbound traffic. This architecture drastically reduces the load of the centralized broker, whose task is just to relay simple and short high-level commands from the source to the destination gateway. The gateway acts as a relay for the Fab Lab inbound traffic, routing the incoming command to the target machine according to specific policies that may include machine availability, type and complexity of the fabrication batch, etc. Thus, NEWTON provides the hardware and software infrastructure to enhance the capabilities of a conventional Fab Lab, empowering the pre-existing infrastructure with a message passing interface that, in turn, would allow the networked machines to operate with minimum or no human interaction. In this new context, a NEWTON Fab Lab is a local network of digital fabrication machines, as illustrated in Fig. 1a. The network connectivity and the ability to control a machine is provided by an external and inexpensive Master Unit (MU) connected to the "slave" equipment. The MU performs the following tasks:

- 1) decodes an incoming message;
- 2) translates the received message in a set of commands understandable by the slave machine;

These operations are performed by the set of software modules organized in the architecture depicted in Fig. 2a. Module communications rely on a system of event-driven asynchronous callbacks managed by an event loop. Conversely, commands dispatched to the "slave" machine, as well as machine monitoring commands are handled synchronously. Command execution is supervised by a controller, whose state diagram is presented in Fig. 2b. In each state an event is generated. These generated events are captured by the other software modules and trigger the execution of specific callback functions. For example, the Jobs Queue integrates an event listener that captures the *new_job* event emitted by the Messaging Subsystem and the *status* events emitted by the Controller and schedules the execution of several event-dependent callbacks as illustrated in the following pseudo-code snippet:

```
var queue = require ('queue');
var controller = require ('controller');

var cq = queue.connect;
var ctrl = controller.connect;

cq.on('new_job', function push(){
    //on new_job event push
    //job in queue
});
cq.on('idle', function dispatch(){
    //on idle event dispatch the
    //highest-priority job to the machine
});
ctrl.on('error', function abort(){
    //on error event abort job
});
...
```

The Messaging Subsystem emits the *new_job* event to push a new job to the Jobs Queue. When the event is fired, the attached callback is executed and the queue status is updated. The controller polls the machine at regular intervals in order to detect its state. When the machine is idle, and event is fired and the listener reacts by dispatching a new job to the machine. Similarly, another callback that aborts decoding is attached to the *error* event emitted by the Controller.

When a new job has been dispatched to the machine for fabrication, the Controller starts monitoring the current drawn by the equipment, building a current pattern for every executed job and saving it in the local data store. As mentioned before, a *deep learning approach* [10] is used to build a behavioral model of a machine; more specifically, a basic classifier based on logistic regression [11] and cross correlation of the measured current patterns is employed to classify jobs and to detect anomalous behaviors.

IV. NEWTON COMMUNICATION PROTOCOL STACK

A. Fab Lab Communication Overview

The communications between client applications and the remote NEWTON Fab Labs rely on a protocol stack which includes a simple publish/subscribe protocol. The fabrication equipment is accessed through the Fab Lab Gateway that routes incoming commands to a given machine depending on both availability and the specific task to be carried on.

The communication protocol relies on a server-to-server model in which some nodes act as message brokers collecting the incoming messages and relaying them towards a destination node.

B. Inter- and Intra-Fab Lab Communications

A fabrication job is routed to a networked Fab Lab by the Cloud Hub message broker; however, the message broker on the cloud side has not direct visibility of the Fab Lab network infrastructure. Its main task is to connect a client to the Fab Lab infrastructure or to perform inter-Fab Labs message routing. The networked machines in a Fab Lab can be accessed through the Fab Lab Gateway only. The gateway main task is routing the outbound traffic to the networked equipment and managing intra-Fab Lab communications.

Fig. 3 presents a simplified timing diagram that describes the communication between the cloud infrastructure and a networked Fab Lab. The message exchange has four stages:

- 1) link establishment;
- 2) topic subscription;
- 3) communication,
- 4) disconnection (not illustrated for the sake of simplicity).

Once the TCP links between the machine and the Fab Lab Gateway on one side, and the Fab Lab Gateway and the Cloud hub broker on the other side, have been established, both the Gateway and the Hub subscribe to topics they are interested in. The topic string is generated using the unique name and connection ID sent by the server that initiates the communications to the destination server during the link establishment. Both the link establishment and the subscription phases are terminated by an ACK message (Init ACK for the link establishment and Subscription ACK for the subscription phase). In other word, the Fab Lab Gateway and the Cloud Hub implement a double broker architecture: the former collects all the incoming messages from the Fab Lab machines whereas the latter collects all the incoming messages from the networked Fab Lab Gateways. The double broker architecture allows the implementation of Fab Lab access and security policies and of custom message filters mechanisms.

Once the subscription phase has terminated, the end nodes start exchanging messages. Each published message can be acknowledged by an optional Publication ACK message. The use of a Publication ACK is mandatory in those cases when it is necessary to guarantee the delivery of a message and to implement retransmission mechanisms to increase the QoS of protocol.

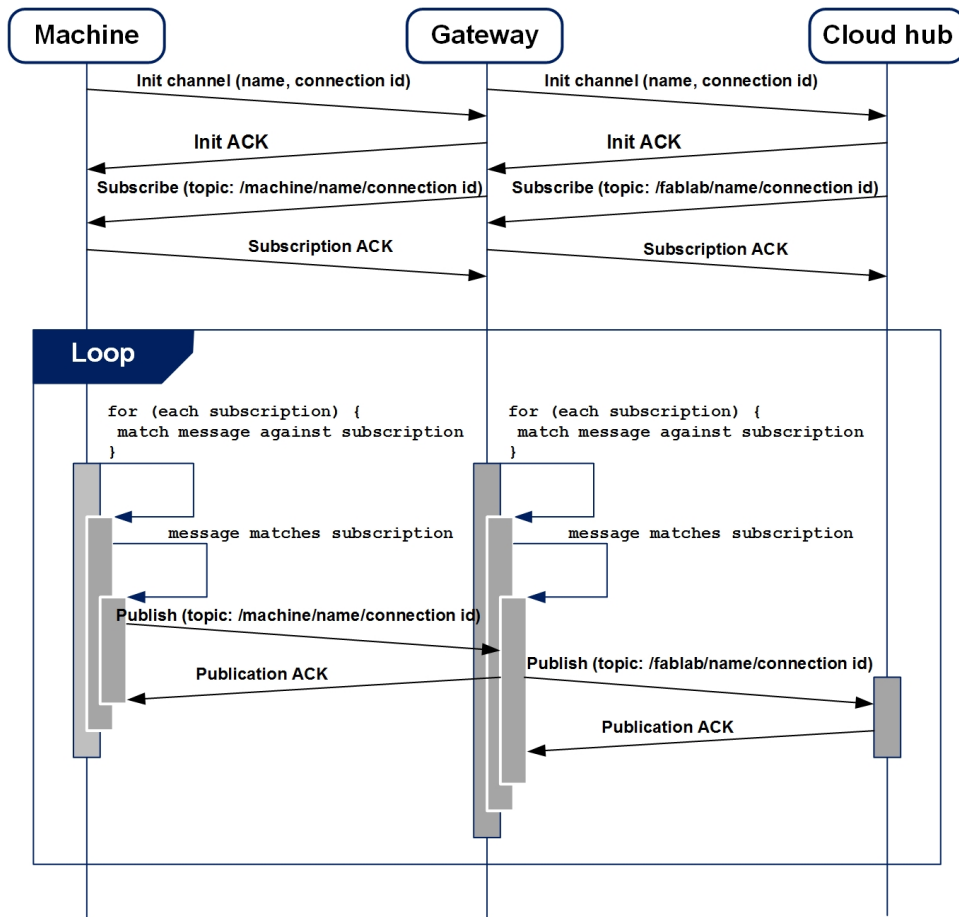


Fig. 3. Overview of the Inter- and Intra-Fab Lab Messaging Flow.

C. Performance Evaluation

The system infrastructure has been tested in real scenarios through small-scale pilots that have involved the participation of six schools and universities located in three European countries as part of the EU-funded NEWTON project. The test pilots have been used to stress the system infrastructure and evaluate the performance of the proposed algorithms for task scheduling and automated service composition [13], [14]. The following figures of merit have been targeted:

- 1) responsiveness of the infrastructure;
- 2) QoS-oriented service composition;
- 3) state and anomaly detection.

In order to detect system peak performance, system infrastructure and APIs have been also load tested using *Locust*⁵. *Locust* allows to simulate user behavior using a Python script. We have designed a simple use case that stresses all the Fab Lab APIs and provides a unified picture of the system performances.

The Fab Lab infrastructure described in Section V has been tested in the following scenarios:

- 1) 50 concurrent users with a hatch rate of 5 users per second.
- 2) 100 concurrent users with a hatch rate of 5 users per second.

- 3) 150 concurrent users with a hatch rate of 5 users per second.

All the incoming requests are forwarded to the same fabrication machine, each test has a duration of 2 minutes and each simulated user performs the following operations:

- 1) GET the available Fab Lab status.
- 2) POST a job to the available Fab Lab.
- 3) GET the status information of the submitted job.
- 4) DELETE the submitted job.
- 5) GET the information of the jobs running in the available Fab Lab

The most time-consuming operation is the POST request to submit a fabrication job since it involves the following steps:

- 1) Uploading the image on the cloud hub.
- 2) Sending the image to the Fab Lab Gateway.
- 3) Sending the image to the target fabrication machine.
- 4) Update the jobs queue in the fabrication machine

Fig. 4 shows the load tests results for the three scenarios under test (i.e., the cases with 50, 100 and 150 concurrent users respectively). Fig. 4a summarizes the overall results for all the request types, whereas Fig. 4b depicts the results only for POST requests. Test results are excellent, considering the the Fab Lab infrastructure has been deployed on inexpensive Raspberry Pi III boards. For example, the 90% of the incoming requests are served in maximum 680 ms for 50-user scenario,

⁵Project Website: <https://locust.io>

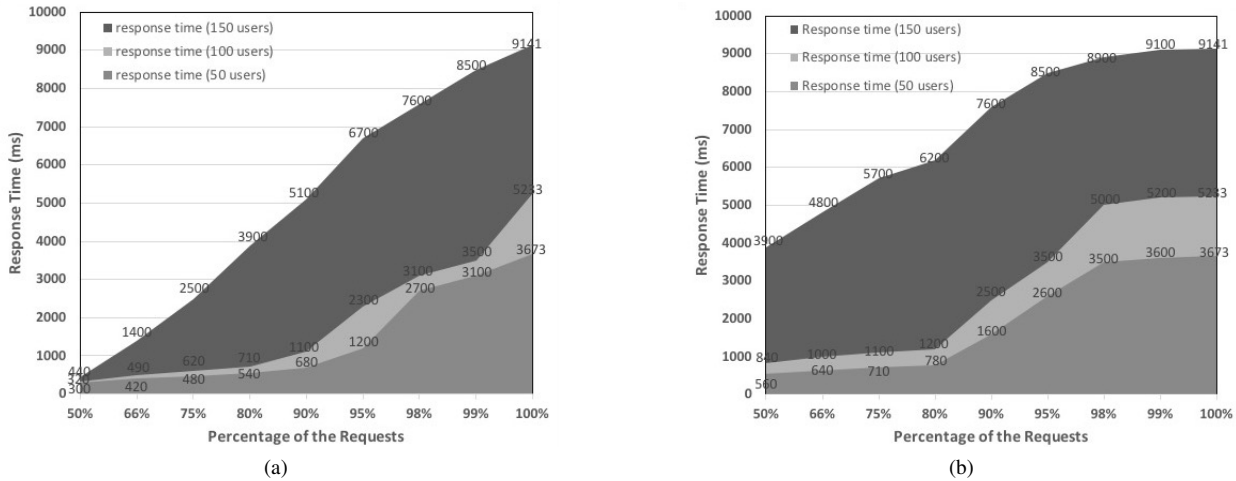


Fig. 4. Percentage of Requests Completed in a Given Time Interval (a) Total Requests, and (b) POST Requests.

1.100 ms for the 100-user scenario, and 5.100 ms for 150-user scenario. Of course, as anticipated earlier in this section, the most time-consuming operations are the POST requests whose delay can be as high as 9.141 ms in the case of 150 concurrent users.

An overview of our findings is summarized in Table I. The table reports the median, minimum, maximum and average response time in milliseconds for each one of the API called by our simulated scenario for all the test cases studied (namely for the 50-, 100- and 150-user load respectively).

The measured values confirm the excellent performance already outlined by Fig. 4. From Table I, the total average response times for the 50-, 100- and 150-user test cases are 452 ms, 568 ms and 1.680 ms respectively, whereas the maximum average response times are 801 ms, 1.158 ms and 3.883 ms respectively. An average response time of 3.883 ms is acceptable and, according to Fig. 4a and 4b allows, on the average, the completion of the 100% of the requests for the 50-user scenario, the 99% of the requests for the 100-user scenario and almost the 80% of the total requests and the 50% of the POST requests for the 150-user scenario.

V. FAAS FAB LAB DEPLOYMENT

As already mentioned, FaaS Fab Lab deployment has been performed as part of the NEWTON platform. The platform is now in production phase and includes the cloud hub (deployed on an Amazon AWS EC2 cluster) and the on-premises interface infrastructure (implemented with inexpensive Raspberry Pi III boards) has been deployed and is presently under test at CEU Madrid, Spain. This deployment has helped gain significant insights on several design and implementation aspects and trade-offs that include hardware design and interfacing, system monitoring and cloud deployment, data security as well as service deployment and orchestration in a multi-cloud environment. Several architectural aspects and implementations have been evaluated and tested so far, with particular emphasis on:

- 1) system replicability and scalability;
- 2) system costs and maintainability;

- 3) service availability and auto-discovery in multi-cloud environments;
- 4) API architecture and design;
- 5) functional and load tests design.

The next step is setting-up the system staging environment that involves networking and interfacing to the cloud hub the Fab Labs at CEU Madrid and Vrije University of Bruxelles, Belgium. This will enable testing the system in a distributed, yet still controlled environment.

VI. CONCLUSION

This paper introduces a novel concept: Fabrication as a Service (FaaS). FaaS enhances existing Fab Lab capabilities by providing the digital fabrication equipment with the possibility to communicate over the Internet in order to remotely control fabrication activities. Using this approach, the fabrication facilities are exposed to the Internet as software services, which may be consumed by third-party applications.

The paper describes FaaS deployment in the context of NEWTON next generation Fab Labs; however, the proposed solution is general, hardware-independent and targets all those scenarios which involve collaborative fabrications. We foresee that this capability will have a huge impact not only on education, but also on industry helping to develop new business models in which fab-less companies may schedule medium or large-scale fabrication batches hiring third-party remote fabrication services.

REFERENCES

- [1] B. Convert, "Europe and the Crisis in Scientific Vocations," *European Journal of Education*, 40(4), 2005
- [2] E. K. Henriksen, et al. "Understanding student participation and Choice in Science and Technology Education", Springer, 2015
- [3] N. Gershenfeld, "How to Make Almost Anything. The Digital Fabrication Revolution", *Foreign Affairs*, vol. 91(6), pp. 43–57, Nov. 2012.
- [4] P. Blikstein, "Digital Fabrication and 'Making' in Education: The Democratization of Invention", In J. Walter-Herrmann & C. Büchling (Eds.), "FabLabs: Of Machines, Makers and Inventors", Bielefeld: Transcript Publishers, 2013.

TABLE I
SUMMARY OF SYSTEM PERFORMANCE (VALUES ARE IN MILLISECONDS)

| API Call | 50 users | | | | 100 users | | | | 150 users | | | |
|---|----------|-----|------|------|-----------|-----|------|------|-----------|-----|------|------|
| | median | min | max | avg. | median | min | max | avg. | median | min | max | avg. |
| GET /fablabs | 220 | 137 | 2990 | 313 | 220 | 137 | 3515 | 347 | 280 | 138 | 3786 | 478 |
| GET /fablabs/fablab:id/jobs?job=job:id | 205 | 139 | 664 | 246 | 212 | 137 | 623 | 256 | 285 | 141 | 3679 | 504 |
| DELETE /fablabs/fablab:id/jobs?job=job:id | 438 | 285 | 3508 | 546 | 568 | 285 | 4803 | 800 | 3750 | 320 | 7600 | 3164 |
| GET /fablabs/jobs | 210 | 139 | 3370 | 354 | 210 | 138 | 3211 | 282 | 230 | 140 | 2948 | 372 |
| POST /fablabs/jobs?machine=type&lat=... | 560 | 374 | 3673 | 801 | 830 | 398 | 5233 | 1158 | 3900 | 502 | 9141 | 3883 |

- [5] T. Martin, S. Brasiel, D. Graham, S. Smith, K. Gurko, D. A. Fields, "FabLab Professional Development: Changes in Teacher and Student STEM Content Knowledge", Annual FabLearn Conference, October 2014, Stanford, CA, USA.
- [6] L. F. Gul, and L. Simisic, "Integration of Digital Fabrication in Architectural Curricula", Digital Fabrication in Education Conference, FabLearn Europe, June 2014, Aarhus, Denmark
- [7] S. Tesconi, and L. Arias, "MAKING As a Tool to Competence-based School Programming", Digital Fabrication in Education Conference, FabLearn Europe, June 2014, Aarhus, Denmark
- [8] N. Padeld, M. Haldrup, and M. Hoby, "Empowering academia through modern fabrication practices", Digital Fabrication in Education Conference, FabLearn Europe, June 2014, Aarhus, Denmark
- [9] W. Shi and D. Schahram, "The promise of edge computing", IEEE Computer, 49(5), 2016, pp. 78-81
- [10] Y. Bengio, "Learning Deep Architectures for AI", Foundations and Trends in Machine Learning, vol. 2(1), Nov. 2009, pp. 1-127
- [11] J. M. Hilbe, "Logistic Regression Models", CRC Press, May 2009.
- [12] F. Shrouf, J. Ordieres and G. Miragliotta, "Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm", IEEE International Conference on Industrial Engineering and Engineering Management, Bandar Sunway, Malaysia, 2014, pp. 697-701. doi: 10.1109/IEEM.2014.7058728
- [13] A. Touhafi, A. Braeken, A. Tahiri, M. Zbakh, "CoderLabs, A Cloud based platform for Real Time Online Labs with User Collaboration", Conference on Cloud Computing Technologies and Applications (CloudTech), Marrakech, Morocco, 2016
- [14] N. T. Cuong, A. Touhafi, K. Steenhaut, "Service Composition with quality of service management in environmental sensor networks", Int. J. of Ad Hoc and Ubiquitous Computing, vol. 23(3,4), 2016.