# A demo of a PaaS for IoT Applications Provisioning in Hybrid Cloud/Fog Environment*

Ons Bibani[‡§], Sami Yangui[†], Roch H. Glitho[†], Walid Gaaloul[‡], Nejib Ben Hadj-Alouane[§], Monique J. Morrow [¶], Paul A. Polakos[*]

[†]Concordia Institute for Information Systems Engineering. Concordia University, Montreal, QC, H3G 2W1, Canada

[‡] SAMOVAR, Telecom SudParis, CNRS. Universite Paris-Saclay. 9, rue Charles Fourier. 91011 Evry Cedex, France

[§]National Engineering School of Tunis, OASIS, University of Tunis El Manar, Tunis, Tunisia

[¶]Cisco systems, Zurich, Switzerland

[*]Cisco systems, New York, NY, USA

***Abstract*—This demo will show the key features of a Platform as-a-Service (PaaS) we have proposed in a research paper accepted for presentation at IEEE LANMAN 2016 conference. The proposed PaaS enables IoT applications provisioning in hybrid cloud/fog environments. Two goals are assigned to the demo. On the one hand, we will highlight how IoT applications can be provisioned in such environments. On the other hand, we will show concretely the advantages these hybrid environments have over traditional cloud environments. The provisioning in these hybrid environments enables latency reduction and processing performance enhancement. Indeed, in order to reduce latency, the applications can have some of its components running in a distant cloud and interacting with the other components running in the fog, closer to IoT devices.**

***Index Terms*—Cloud computing; Fog computing; IoT; Latency; PaaS**

## I. Context, Use Case & Purpose of the Demo

In cloud computing, the PaaS hosts and executes end-user applications offered as SaaS. Such applications' components may be located far from the data sources such as wireless sensors. This may cause unacceptable delays for latency-sensitive applications such as fire detection applications. Fog computing has been introduced in the recent past. It extends the traditional cloud computing architecture to the edges of the network by enabling computation and storage closer to the end-users and/or data sources when appropriate, to avoid excessive delays [1]. With fog computing, some of the components of an application could be hosted and executed in a cloud PaaS and interact with the other components hosted and executed in the fog, closer to the end-user and/or data sources such as wireless sensors.

Let us illustrate this model in a component-based fire detection and fighting application depicted in Fig. 1. It is an Internet of Things (IoT) application, IoT presently being one of the most pertinent application domains for the fog. In this figure, the fire detection component (C1) is hosted and executed in the fog close to the data sources to enable a rapid fire detection. On the other hand, the fire-fighting strategy component (C2) and the robot dispatcher component (C3) are hosted and executed in the distant PaaS. Although not shown in the figure, there might be several fogs and the robot dispatcher module could be hosted and executed in another fog closer to the robot fleet.

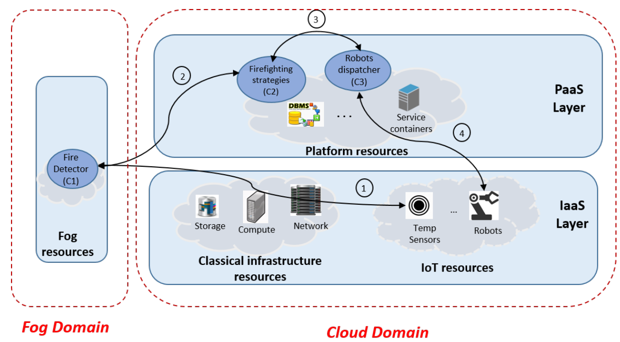


Fig. 1. Component-based fire detection and fighting application deployed in a hybrid cloud/fog environment

In traditional cloud setting, all the three application's components would be hosted and executed in the PaaS. This might cause unacceptable delays when it comes to fire detection and eventually robots dispatching. As part of our work, we proposed a PaaS that supports and automates the provisioning of applications in a hybrid cloud/fog environment [2]. The purpose of this demo, is to show the key features of the prototype that we have implemented. On the one hand, we will highlight how IoT applications can be provisioned in hybrid cloud/fog environments. On the other hand, we will show concretely the advantages these hybrid environments have over traditional cloud environments. Our prototype implements the fire detection and fighting application provisioning. It also enables the migration of application's components from cloud to fog and vice versa.

## II. Prototype, Demo Overview & Lessons Learned

The implemented prototype for the demo is depicted in Fig. 2. It consists of three domains. The first is the IoT domain

*This demo will show the key features of the PaaS proposed in the full research paper: *S. Yangui, P. Ravindran, O. Bibani, R. H. Glitho, N. B. H. Alouane, M. J. Morrow, and P. A. Polakos, A Cloud Platform as-a-Service for Hybrid Cloud/Fog Environments, in Proceedings of the 22nd IEEE International Symposium on Local and Metropolitan Area Networks, Rome, Italy, 2016.*

978-1-4673-9882-4/16/ $31.00 

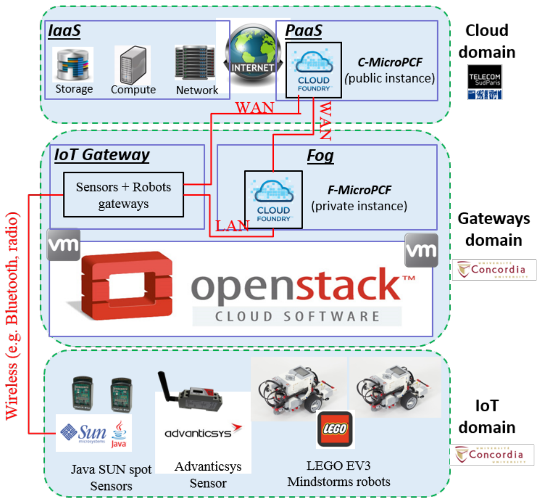


Fig. 2. The prototype overview

that includes the wireless sensors used for sensing temperature and the robots used as firefighters (i.e. 2 Java Sun SPOT wireless sensors, 1 Advanticsys wireless sensor and 2 LEGO EV3 Mindstorms robots). The second domain is the gateway domain located close to the IoT domain and accessible via short range radio (i.e. Bluetooth). It is made up of the IoT gateway required for protocol and information model conversion and of the fog execution environment required for the execution of the components located in the fog. The third and last domain is the cloud domain. It is further away and accessible via a wide area network. It includes the PaaS for a hybrid cloud/fog environment and the IaaS needed to execute the components located in the cloud. It should be noted that the fog execution environment and the PaaS are based on an extended Cloud Foundry that enables the integration of the fog.

In the demo, we show the whole provisioning process of the fire detection and fighting application in the hybrid cloud/fog environment. This covers: (1) Developing and composing the application components, (2) deploying them, and (3) managing (including executing) them. For the development phase, we use the Eclipse IDE for code editing and Activiti GUI to draw the BMPN process associated to the application. Such a process allows the modeling of how the application's components will be orchestrated during runtime (see Fig. 3). We also use Docker containers to package the application's components and the CF plugin to push them to Cloud Foundry instances at the end of this phase.

For the deployment phase, we use our extended Cloud Foundry solution to host the applications. The Docker nodes are then deployed over cloud foundry droplets which are deployed across cloud or fog cloud foundry instances. For the demo, the cloud foundry instance part of the cloud domain (i.e. C-MicroPCF) will be deployed in the TSE Concordia lab in Montreal and accessible through the Internet while the IoT and gateways domains, including the cloud foundry instance deployed in fog (i.e. F-MicroPCF), will be located locally in

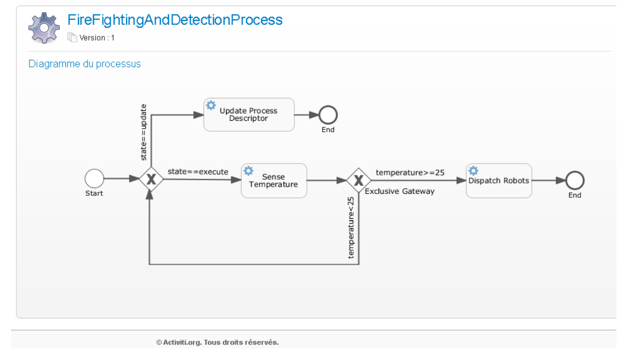


Fig. 3. The BPMN process associated to the fire detection and fighting application

the demo room.

For the management phase, we use Apache Activiti server as orchestration engine to execute the BPMN process associated to the application. We also use a migration engine that interacts with the Docker registry management framework and allows moving application's components within the Docker nodes from cloud to fog and vice versa.

A set of live experiments will also be made to show audience some insights in the gains associated with the use of fog. The application's components will be moved from fog to cloud and vice versa. For each manipulation, the corresponding end-to-end delay will be measured. This delay represents the necessary time from sending temperatures to dispatching robots. The results will show that such delay is minimized (i.e. *484ms*) when all the components run in the fog close to the IoT devices. However, it is interesting to note that the worst result (i.e. *2033ms*) is not when all the components are in the cloud; it is when C2 is in the fog and the other components are in the cloud. It is also interesting to note that placing in the fog the two components that interact with the IoT devices (i.e. C1 and C3) does not necessarily lead to the best results.

## III. Conclusion

This demo aims at showing our performed prototype that implements provisioning of realistic IoT use case in hybrid cloud/fog environments. Its ultimate goal is to highlight the benefits from integrating a fog layer in a cloud system in terms of latency and end-to-end delay.

## Acknowledgment

This work is partially supported by CISCO systems through grant CG-589630.

## References

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*. New York, NY, USA: ACM, 2012, pp. 13–16.

[2] S. Yangui, P. Ravindran, O. Bibani, R. H. Glitho, N. B. H. Alouane, M. J. Morrow, and P. A. Polakos, "A Cloud Platform as-a-Service for Hybrid Cloud/Fog Environments," in *Proceedings of the 22nd IEEE International Symposium on Local and Metropolitan Area Networks*, Rome, Italy, Jun. 2016, p. To appear.