Security and IoT Cloud Federation: Design of Authentication Schemes

Luciano Barreto[1], Antonio Celesti[2], Massimo Villari[2], Maria Fazio[2](✉), and Antonio Puliafito[2]

[1] Federal University of Santa Catarina, Florianópolis, Brazil
lucianobarreto@das.ufsc.br
[2] Università Degli Studi di Messina, Messina, Italy
{acelesti,mvillari,mfazio,apuliafito}@unime.it

Abstract. The advent of both Cloud computing and Internet of Things (IoT) is changing the way of conceiving information and communication systems. Generally, we talk about IoT Cloud to indicate a new type of distributed system consisting of a set of smart devices interconnected with a remote Cloud infrastructure, platform, or software through the Internet and able to provide IoT as a Service (IoTaaS). In this paper, we address such a challenging paradigm focusing on security in IoT Cloud Federation. In particular, we discuss several authentication schemes fitting different types of scenarios.

Keywords: Cloud federation · IoT · Authentication scheme

1 Introduction

Nowadays, in the Internet of Things (IoT) panorama, the number of smart devices that can be integrated in different physical environments is rapidly growing. Considering such a context, smart devices can be deployed for collecting sensing data (e.g., temperature, pressure, etc.) and to perform (actuate) actions (e.g., turn on/off a light, send an alert, etc.). The success of IoT is due to the recent investments on both hardware and software technologies that are allowing IoT infrastructure, platform and applications to quickly evolve. Another factor that is contributing to the rapid evolution of IoT is its combination with the Cloud computing paradigm that is pursuing new opportunities in delivering services, representing a strategic approach for IT operators of increasing their business. The emerging business perspectives coming from IoT are pushing private, public, and hybrid Cloud providers to integrate their system with smart devices (including sensors and actuators) in order to provide together with the traditional Infrastructure, Platform, and Software as a Services (IaaS, PaaS, SaaS) even a new type of transversal service level, that is *IoT as a Service* (IoTaaS). An *IoT Cloud* represents a new type of distributed system consisting of several smart devices interconnected with a remote Cloud infrastructure, platform, or software through the Internet that is able to provide IoTaaS. We believe that the

B. Mandler et al. (Eds.) IoT 360° 2015, Part I, LNICST 169, pp. 337–346, 2016.
DOI: 10.1007/978-3-319-47063-4_35

near future evolution of IoT Clouds will be the establishment of federated environments, in order to extend context-based capabilities and increase flexibility in IoTaaS provisioning. In a federated scenario, how to access IoT devices and services in a secure way is a very big concern. In this paper, we address security issues in federated IoT Clouds, specifically focusing on authentication strategies, presenting a new system model for secure IoT Cloud Federation and discussing several authentication schemes that allow users and manufacturers to access IoT devices and IoTaaS in a secure way. In particular, our use cases are based on the Identity Provider/Service Provider (IdP/SP) and Trusted Computing models.

The paper is organized as follows. Section 2 briefly describes the state of the art of IoT Cloud security. In Sect. 3, we provide an overview on IoT Cloud federation, specifically focusing on IoT resources and identity federation. In Sect. 4, we present a system model for IoT Cloud federation. In Sect. 6, we describe several authentication schemes for IoT Clouds and the resulting protocol flows formalized by means of different sequence diagrams. Section 7 concludes the paper.

2 Related Work

Cloud federation is a topic that has been studied for years and several experiences in this research area have been discussed n literature. One of the first scientific works on Cloud federation was presented in [1], where a federation of Clouds was described as a model of multiple providers aimed at resource sharing. Users are associated with a provider that is responsible for fulfilling all customer requests. Similar approaches can be found in [2,3]. All these models are based on a central Cloud broker, that looks for and allocate resources into the Clouds. Some security concerns on Cloud federation were presented in [4], where the authors defined authentication protocols.

Recent scientific works on the integration of IoT devices and Cloud computing providers was presented in [5], where protocols and use cases on how to integrate IoT devices with a Cloud computing provider were described. In [6], the authors present a system model for the development of applications for processing sensing data collected by IoT devices. The main idea is that the processing system runs over the Cloud, and IoT devices are exclusively exploited to collect sensing data. Another example of integration between IoT and Cloud computing is described in [7], where the authors describe an hybrid storage system specifically aimed to store Big Data collected for smart environment monitoring.

Regarding IoT Cloud security, in [8,9] the authors discussed how to perform a self-identification process in order to achieve a secure auto-configuration of IoT devices joining the Cloud. In [10] the authors present the challenges of integrating IoT devices with the Cloud (Cloud of Things as defined by the authors). The authors present a business model for this kind of architecture as well as the limitations and issues related to the security of IoT devices.

The limited number of scientific works focusing IoT and Cloud Federation security proves how currently this topic is still at an early stage and needs to be investigated with more attention.

3 IoT Cloud Federation Overview

Cloud federations has been widely discussed in terms of federation of datacenters [4,11]. In such a distributed system model, Clouds and, hence, related providers connect to each other in order to share their resources, typically Virtual Machines (VMs). Interactions among providers are based on pre-established trust relationships, so that they share their resources with other trusted providers.

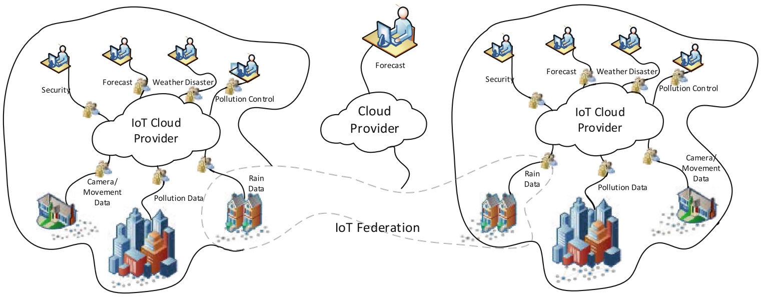


Fig. 1. IoT federation overview

Following the same idea, it is possible to think to share IoT devices as resources. We define IoT Cloud federation as a mesh of IoT Cloud providers that are interconnected to provide a wide decentralized sensing and actuating environment where everything is driven by agreements in a ubiquitous infrastructure. In such an environment, smaller, medium, and large IoT Cloud providers can federate themselves to gain economies of scale and an enlargement of their sensing and actuating capabilities, in order to arrange more flexible IoTaaS. Providers managing IoT devices make them available to other federated providers and their users. This allows users to access different kinds of data from different sensors, possibly deployed in different geographical regions. Figure 1 shows an example of such a scenario, where several providers share their IoT devices, allowing external users to collect data coming from different locations. From a business perspective, IoT Clouds can elastically enlarge the set of available IoT devices to deliver advanced IoTaaS to their users.

4 System Model for Federated IoT Clouds

In order to design authentication schemes, in this Section, we present a basic system model for IoT Cloud federation. As shown in Fig. 2, the model includes several components. The main building block represents an IoT Cloud and, in our model, the federation involves many IoT Clouds. At the high-level, each IoT Cloud is basically composed by two elements: the Cloud Provider and the Identity and Access Management (IAM) system. The *Cloud Provider* is a piece of middleware that manages the IoT resources of an administrative domain.

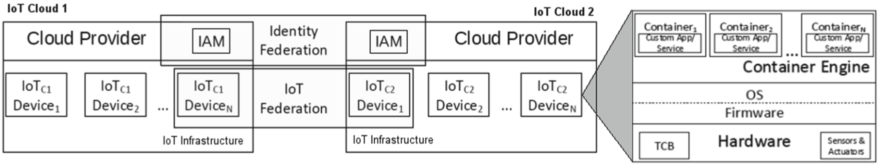


Fig. 2. Model overview of IoT federation

In addition, this element is responsible for managing the access to resources among federated IoT Clouds, IoT device, and users. The IAM is responsible to secure the access to IoT resources and IoTaaS. It is responsible for managing user identities, as well as maintaining secure IoT devices that are connected to the IoT Cloud. Besides the local management of users, the IAM element manages the authentication credentials required to establish federated relationships among IoT Cloud providers. The federated identity management is based on agreements that enable organizations to share their users' identities [12]. A challenging mechanisms to support identity federation solutions is the Single Sign-On (SSO) [13], that allows users to pass through the authentication process once accessing to different trusted service providers.

At the low level of the system model depicted in Fig. 2, there are several IoT devices. Each IoT device component exploits a relatively new technology, i.e., Container Virtualization. The concept of container applied to computers is currently object of studies in IoT devices. Following the same concept applied to computers, using containers in IoT devices means abstracting hardware resources creating virtual execution environments. Pushing containers into IoT devices is a very innovative approach and more and more manufactures are looking at container engines to simplify the packaging, distribution, installation and execution of complex applications on IoT devices. For example, a popular emerging solution consists in deploying Docker [14] on multi-core Raspberry PI devices.

The IoT device model adopted in this paper is shown on the right part of Fig. 2. We can break up the device into three layers: Container Engine, OS/Firmware and Hardware. The *Container Engine* is a software layer that enables the deployment and execution of containers. Cloud users can request to instantiate a container on a device to perform a specific task/application. Several Container per user can be instantiated on the same IoT devices thanks to isolation mechanisms, that is a Container is accessible only by its owner. In this sense, authentication and authorization controls must be enforced for accessing containers. The *Firmware* layer in the IoT device manages hardware resources. It abstracts hardware resources and provides an interface to control them. In addition, the firmware can be updated to provide new features or for bug correction. Therefore, the firmware of an IoT device is critically important because possible failures can compromise the behavior of the whole IoT device. The firmware can be either integrated as part of the Operating System (OS) (e.g., in Raspberry) or independent (e.g., in Arduino Yun). Given the importance of this layer in the IoT device, the access must be completely secured. In our proposal, only two entities may have access to this software layer: the Cloud provider, that controls and manages the device, and the manufacturer, that produced it. At the hardware layer, there are two main elements: *sensors/actuators* and the *Trusted Computing Base (TCB)* [15]. The latter allows IoT devices to be identified only by the Cloud provider and ensures that malicious entities do not have corrupted the hardware/software configurations. The TCB can be a Trusted Platform Module (TPM), a FPGA Boards, or an USB Crypto Token. The latter is a technology that is really promising for IoT devices security [5].

5 IoT Resource and Identity Federation

As highlighted in Fig. 2, the authentication schemes proposed in this paper address two main aspects of IoT Cloud federation, i.e., *Identity Federation* and *Resource Federation.*

IoT Resource Federation is aimed at sharing IoT devices among IoT Cloud providers. Each IoT Cloud provider manages accesses to its own IoT devices. Moreover, it has an updated list of external IoT devices belonging to other federation providers and can ask for temporary access to these devices on demand. Thus, for example, a user can request through is provider data from an IoT device belonging to a federated IoT Cloud in a transparent way. In addition, IoT Cloud federation allows providers to arrange new IoTaaS that they could not provide only using their own infrastructures, i.e., an IoTaaS can be composed combining features from different federated IoT devices. A preliminary requirement for the federation establishment is the creation of trust relationships among providers according to particular Service Level Agreements (SLAs). After that, the design and implementation of an infrastructure that allows the management of federated IoT devices is required. Such an infrastructure may involve mechanisms for resource discovery, resource allocation, identity management, and so on.

IoT Identity Federation allows IoT Cloud users to access IoT devices and IoTaaS belonging to a federated Cloud environment by forwarding the request to the federated provider. Thence, the authentication process should be extended beyond the administrative domain of a single provider. Identity federation allows users to access other IoT Cloud provider in a transparent way through a SSO authentication process. In particular, a user, with single assertion or authentication token can access the IoT devices belonging to different trusted federated IoT Cloud providers. The main advantage of this kind of federation is that the user does not need to keep multiple accounts for different providers. Generally, the identity federation requires one or more trusted Identity Providers, which manage users' credentials and a Certification Authority responsible for issuing digital certificates needed for authentication.

6 Users, Roles, and Authentication Schemes

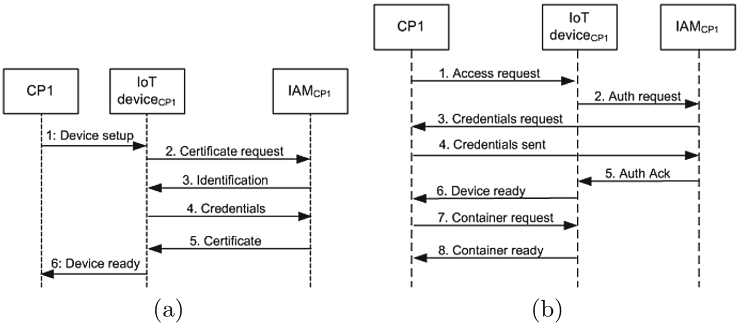
According to the previously described IoT Cloud federation system model, different types of entities need to access IoT devices for different purposes. In this

Section, we describe several authentication schemes addressing different scenarios. We remark that in this paper we focus on authentication, instead authorization and auditing are out of the scope of this paper.

6.1 Maintenance and Container Setup

The piece of firmware of an IoT device can be upgraded for multiple reasons, e.g., to offer new software capabilities, for bug fixing, and so on. Only authorized *Manufacturers* and *Cloud Providers* that manage these IoT devices must be able to perform such types of critical operations, because unauthorized users could corrupt IoT devices with serious risks for the security of the whole system. We define Cloud Providers and Manufactures as kinds of super users who hold the rights to performs the aforementioned operations on IoT devices through a direct access. Figure 3(a) shows the interaction required to install a digital certificate in the TCB of the IoT device in order to make it trusted with a Cloud provider/Manufacturer. In step 1, a Cloud Provider (CP1) starts a setup process contacting the $IoTdevice_{CP1}$ that in turn, in step 2, contacts the IAM_{CP1} requesting a digital certificate. In step 3, the IAM_{CP1} requests identification info from the $IoTdevice_{CP1}$ that is sent back in step 4. In step 5 the IAM_{CP1} generates a certificate that is sent to the $IoTdevice_{CP1}$ and installed. In step 6 $IoTdevice_{CP1}$ is ready for further configuration.

In our Cloud federation model, the Cloud provider acts also as *Container Manager*. In fact, it is able to manage containers on IoT devices along with other storage, processing, and networking resources of the datacenter in order to arrange IoTaaS. Therefore, a safe access to the container engine of IoT devices is required. The authentication process required to instantiate a new container on the IoT device is shown in the sequence diagram of Fig. 3(b). In step 1, $CP1$ sends an access request to the $IoTDevice_{CP1}$, that in step 2, sends an authentication request to IAM_CP1. In step 3 IAM_CP1 sends a credentials request to $CP1$. Credentials are sent to the IAM_CP1 in step 4. In step 5, the $IoTDevice_{CP1}$ is informed that $CP1$ is authenticated. In step 6 $CP1$ is notified that it can control



(a) (b)

Fig. 3. (a) Sequence diagram of digital certificate setup process. (b) Sequence diagram of a container instantiation process.

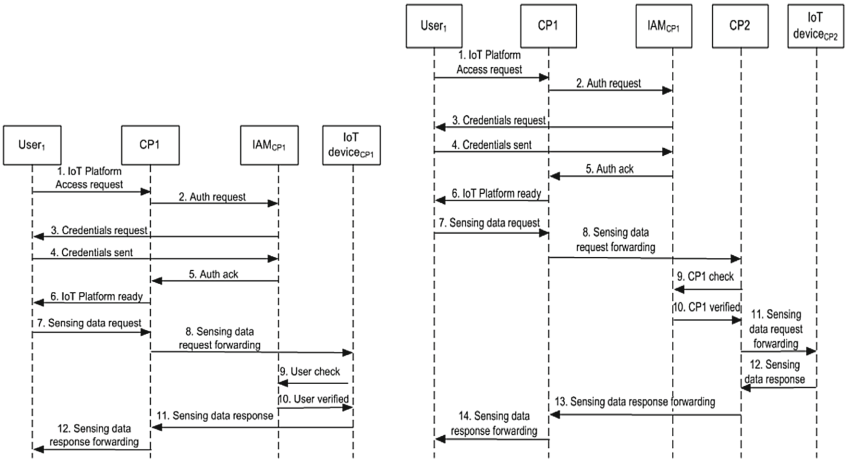
$IoTDevice_{CP1}$. In step 7 $CP1$ sends a container request to $IoTDevice_{CP1}$ that instantiates it. In step 8, $CP1$ is informed that it can use the new container.

6.2 Accessing IoT Devices

In this Section, we present two main user roles for accessing IoT devices: *Basic User* and *Advanced User*. A Basic User can access sensing data through the IoT Platform offered by Cloud Provider, whereas an Advanced User is able to perform a direct access to the IoT device in order to manage containers. Moreover, we consider both roles acting in local and in a federated Cloud.

Basic Local and Federated Users. A Basic User is a user who only needs sensing data coming from IoT devices. The user gets sensing data through the IoT Platform APIs supplied by a Cloud Provider. In this case, the Cloud Provider accesses the IoT device on behalf of the user and the IoT device is transparent for the user. The user is defined "local" when he/she accesses his/her own Cloud provider and "federated" when it access another Cloud that is federated with his/her Cloud provider.

Figure 4(a) shows the sequence diagram of a Basic Local User authentication. In step 1, the user sends an access request to $CP1$ in order to access his/her IoT Platform. In step 2, an authentication request is sent IAM_{CP1}. In step 3 credentials are requested to the $user_1$ and in step 4, they are sent to the IAM_{CP1} that authenticates the user. In step 5, an authentication acknowledgement is sent to $CP1$ and in step 6 $user_1$ is informed that he/she got grant access rights to access the $CP1$ IoT Platform. In step 7, a sensing data request is sent to $CP1$ IoT



(a) (b)

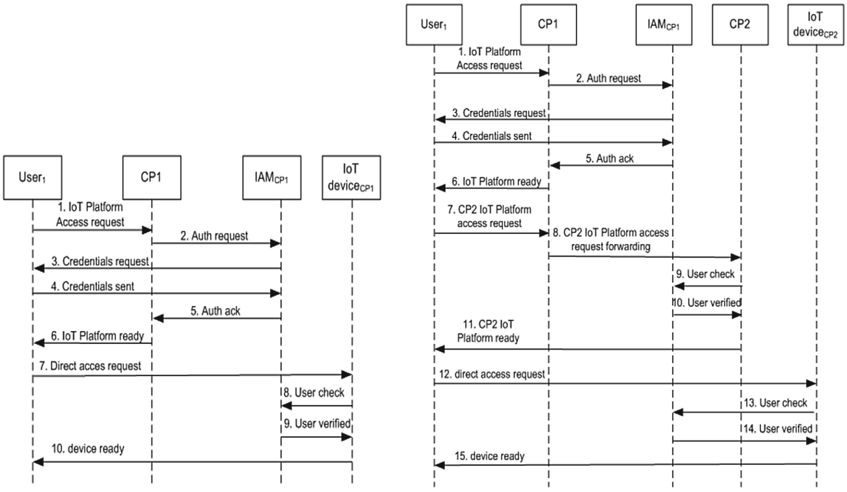
Fig. 4. (a) Sequence Diagram of Basic Local User Access. (b) Sequence Diagram of Basic Federated User Access.

Platform that is forwarded to $IoTDevice_{CP1}$ in step 8. In step 9, $IoTDevice_{CP1}$ checks if $user_1$ is authenticated, and since an authentication assertion already exist, exploiting the well-known concept of Single Sign On (SSO), in step 10, IAM_{CP1} notifies that he/she is already authenticated. In step 11, a sensing data response is sent to $CP1$ IoT Platform. Finally, in step 12, such a response is forwarded to $user_1$.

Figure 4(b) shows the sequence diagram for the Basic Federated User authentication. In this scenario, we consider two federated Cloud providers: CP_1 and CP_2. Let us assume that $user_1$ belonging to CP_1 wants to access sensing data supplied by the CP_2 Iot Platform. For this purpose, a federated SSO authentication process is required. Thanks to the concept of federated identity, $user_1$ can access the IoT Platform APIs of CP_2. Step 1, 2, 4, 5, 6, and 7 are similar to the Basic Local User authentication sequence diagram previously described. In step 8, a sensing data request is forwarded to the CP_2 IoT Platform. In step 9, a federated authentication process is performed. In this example, we assume that IAM_{CP1} acts as federated Identity Provider (IdP) and that $CP2$ is trusted with it, but it is also possible to consider either another trusted third party or a federated network of different IdP(s). In particular, $IoTDevice_{CP1}$ checks if $CP1$ is authenticated, and since an authentication assertion already exist, in step 10, exploiting the well-known concept of SSO, IAM_{CP1} notifies that it is already authenticated. In step 11, the sensing data request is forwarded to $IoTDevice_{CP2}$. In the end, pieces of sensing data are forwarded back to $user_1$ in steps 12, 13, 14.

Advanced Local and Federated Users. The Advanced User needs a direct access to IoT devices in order to customize them, control the container engine, manage containers, collect raw data and set actuators. Compared to the Basic Local User, the authentication is quite different. The user is defined "local" when he/she accesses his/her own Cloud provider and "federated" when it access another federated Cloud. Figure 5(a) shows the sequence diagram of an Advanced Local User authentication. Steps from 1 to 6 are similar to the sequence diagrams previously described. In step 7 the $user_1$ requests a direct access to $IoTDevice_{CP1}$. In step 8, an authentication process is performed between $user_1$ and $IoTDevice_{CP1}$. In particular, $IoTDevice_{CP1}$ checks if $user_1$ is authenticated, and since an authentication assertion already exist, in step 9, exploiting the well-known concept of SSO, IAM_{CP1} notifies that it is already authenticated. In step 10 $user_1$ gains the control of $IoTDevice_{CP1}$, e.g., from now on he/she can instantiate containers. In this case, sensing data are directly sent to $user_1$.

Figure 5(b) shows the sequence diagram of an Advanced Federated User authentication. Steps from 1 to 6 are similar to the sequence diagrams previously described. In step 7, $user_1$ sends a request to access the $CP2$ IoT Platform and such a request is forwarded to $CP2$ in step 8. In steps 9 and 10 $CP2$ verifies that a valid authentication assertion exists for $user_1$ in the trusted third party (i.e., IAM_{CP1} in this example). In step 11, $user_1$ gain the access to the $CP2$



(a) (b)

Fig. 5. (a) Sequence Diagram of Advanced Local User Access. (b) Sequence Diagram of Advanced Federated User Access.

IoT Platform. In step 12, $user_1$ sends a direct access request to $IoTDevice_{CP2}$. In steps 13 and 14 $IoTDevice_{CP2}$ verifies that a valid authentication assertion exists for $user_1$. Finally, $user_1$ gains the direct control of $IoTDevice_{CP2}$ in step 14.

7 Conclusion

In this paper, we combined several cutting-age topics, that are IoT, Cloud computing, and federation. In particular, we focused on security proposing several authentication schemes for IoT Cloud federation. From our study, we can conclude that designing and developing authentication schemes in emerging IoT Cloud scenarios is not trivial at all due to the current technological limitations. In fact, the real obstacle in the development of our scenario is represented by the development of TPB and related software features in IoT devices. In this regard, even though the Trusted Computing Group has recently started to look at IoT, at the time of writing of this paper, there are not concrete implementations yet. In this scientific work, we hope we succeeded in stimulating the interest of researchers and developers towards this topic.

Acknowledgements. The research leading to the results presented in this paper has received funding from the European Union's FP7 2007–2013 Project frontierCities under grant agreement number 632853.

References

1. Calheiros, R.N., Toosi, A.N., Vecchiola, C., Buyya, R.: A coordinator for scaling elastic applications across multiple clouds. Future Gener. Comput. Syst. **28**(8), 1350–1362 (2012)
2. Carlini, E., Coppola, M., Dazzi, P., Ricci, L., Righetti, G.: Cloud federations in contrail. In: Alexander, M., et al. (eds.) Euro-Par 2011, Part I. LNCS, vol. 7155, pp. 159–168. Springer, Heidelberg (2012). doi:10.1007/978-3-642-29737-3_19
3. Villegas, D., Bobroff, N., Rodero, I., Delgado, J., Liu, Y., Devarakonda, A., Fong, L., Masoud Sadjadi, S., Parashar, M.: Cloud federation in a layered service model. J. Comput. Syst. Sci. **78**(5), 1330–1344 (2012)
4. Celesti, A., Tusa, F., Villari, M., Puliafito, A.: How to enhance cloud architectures to enable cross-federation. In: 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), pp. 337–345 (2010)
5. An Authentication Model for IoT Clouds, Paris, France, 26–27 August 2015, IEEE Computer Society (2015, in press)
6. Rao, B.B.P., Saluia, P., Sharma, N. Mittal, A., Sharma, S.V.: Cloud computing for Internet of Things & sensing based applications. In: 2012 Sixth International Conference on Sensing Technology (ICST), pp. 374–380 (2012)
7. Fazio, M., Celesti, A., Puliafito, A., Villari, M.: Big data storage in the cloud for smart environment monitoring. Procedia Comput. Sci. **52**, 500–506 (2015)
8. Villari, M., Celesti, A., Fazio, M., Puliafito, A.: A secure self-identification mechanism for enabling IoT devices to join cloud computing. In: Giaffreda, R., Cagáňová, D., Li, Y., Riggio, R., Voisard, A., Cagánová, D., Cagánová, D. (eds.) IoT 2014. LNICST, vol. 151, pp. 306–311. Springer, Heidelberg (2015). doi:10.1007/978-3-319-19743-2_41
9. Barreto, L., Celesti, A., Villari, M., Fazio, M., Puliafito, A.: Authentication models for IoT clouds. In: International Symposium on Foundations of Open Source Intelligence and Security Informatics FOSINT-SI. IEEE Computer Society (2015)
10. Parwekar, P.: From internet of things towards cloud of things. In: Computer, pp. 329–333 (2011)
11. Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I.M., Montero, R., Wolfsthal, Y., Elmroth, E., Caceres, J., Ben-Yehuda, M., Emmerich, W., Galan, F.: The reservoir model and architecture for open federated cloud computing. IBM J. Res. Dev. **53**(4), 1–11 (2009)
12. Sharma, A.K., Lamba, C.S.: Survey on federated identity management systems. In: Meghanathan, N., Boumerdassi, S., Chaki, N., Nagamalai, D. (eds.) NeCoM 2010. CCIS, vol. 90, pp. 509–517. Springer, Heidelberg (2010). doi:10.1007/978-3-642-14493-6_52
13. Maler, E., Reed, D.: The Venn of identity: options and issues in federated identity management. IEEE Secur. Priv **6**, 16–23 (2008)
14. Zheng, C., Thain, D.: Integrating containers into workflows: a case study using makeflow, work queue, and docker. In: Proceedings of the 8th International Workshop on Virtualization Technologies in Distributed Computing, VTDC 2015, pp. 31–38. ACM (2015)
15. Rushby, J.: A trusted computing base for embedded systems. In: Proceedings of the 7th Department of Defense/NBS Computer Security Conference, pp. 294–311 (1984)