Middleware to Integrate Mobile Devices, Sensors and Cloud Computing

Thinh Le Vinh, Samia Bouzefrane, Jean-Marc Farinone, Amir Attar

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2015) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

The 6th International Conference on Ambient Systems, Networks and Technologies (ANT 2015)

Middleware to Integrate Mobile Devices, Sensors and Cloud Computing

Thinh Le Vinh, Samia Bouzefrane*, Jean-Marc Farinone, Amir Attar, Brian P. Kennedy

CEDRIC Lab, Conservatoire National des Arts et Métiers, Paris, France

Abstract

The emergence of the Internet of Things (IoT) and the advantages of Mobile Cloud Computing (MCC) have drawn a big absorption from technological experts. This paper investigates the integration of IoT and MCC by introducing the service architecture towards the applications of mobile devices, sensors and Cloud computing. The proposed approach not only presents the cutting edge convergence between IoT and MCC but also focuses on the smart home scenario to answer the question what the benefits are, coming from the integration of IoT and MCC. A developmental process of DropLock architecture dedicated to Smart City is also presented to demonstrate this convergence.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: Internet of Things; Mobile Cloud Comupting; Smart City; Service Architecture.

1. Introduction

Internet of Things, as we know today, had not been around for a long time. It was not until 1999 that Kevin Ashton was considered as the former of the concept of Internet of Things (IoT). Since then, the IoT has evolved tremendously from the personal guest room to the heavy factory floor by either using ordinary technology such as Radio Frequency Identification (RFID), Wireless Sensor Network, Bluetooth or taking advantage of the Cloud computing facilities availability. It has offered us a Smart City including smart home, smart city and clean living environment. The term of IoT is a simple concept that Internet evolves from a way for devices to be smarted and interconnected. These devices are able to gather processed data and make decisions appropriately. According to International Data Corporation (IDC), the IoT is a network of networks of uniquely identifiable endpoints (or

* Corresponding author. Tel.: +33 (0) 1 40 27 25 83; fax: 01-40-27-22-96.
E-mail address: samia.bouzefrane@cnam.fr

Peer-review under responsibility of the Conference Program Chairs.

"things") that communicate without human interaction using IP connectivity. The IoT world Forum of 2014[1] predicted that there is a huge market for IoT solutions growing from $1.9 trillion in 2013 to $7.1 trillion in 2020.

In addition, Cloud Computing and IoT not only offer the most cutting- edge and sought after technology but also give a big step towards new demand communication vision. To be more specific, in order to get smarter in Things, there are arising challenges such as user scalability, data storage, data processing ability in heterogamous resources environment, and energy saving. To address these challenges, Cloud Computing delivers the latest services; namely Network as a Service (NaaS), Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS) and Storage as a Service (STaaS) to abstract all types of computing resources as services [9]. As a result, the shared Cloud infrastructure services work as the utility: the users with their smart things can access and pay for what services they need from remotely hosted resources over the Internet. These services not only upgrade automatically but also easily scale up or down.

Mobile Cloud Computing (MCC) is viewed as the combination of Cloud Computing and mobile networks to bring benefits for mobile users, while augmenting the resource-constraint mobile devices with varied Cloud-based resources in terms of data storage, computation and energy.

The contribution of our work is to investigate the state of art of the convergence of MCC and IoT in the context of Smart Home deployment. Furthermore, by delivering a proposed schema, we look into a software architecture which IoT and MCC converge. The paper is structured as follow: first, we are going to review the previous work in the combination of Cloud Computing and Internet of Things in Section 2. Then in Section 3, we are going to describe how the Mobile Cloud Computing and the Internet of Things work together. The System Model is depicted in Section 4, by designing a service architecture that meets the features of IoT and MCC. Section 5 is going to deliver our use case application called DropLock, with detailed descriptions of the implementation. Finally, the conclusion and our future plan are going to be presented in Section 6.

2. Related work

Recently, many research works were interested in combining Cloud Computing and IoT to design systems for Smart City, like in [1, 2, 3, 4, 6, 7, 8 and 9]. In fact, many authors have highlighted the benefits of using Cloud Computing with IoT and proposed a Cloud infrastructure to extend the limited resources of the sensors and to facilitate the management of the sensor-centric applications in many domains. However, with regard to MCC and IoT convergence, there are fewer research works. For example, Gachet et al. in [5] discussed about the Virtual Cloud Carer project, with social and health objectives, and presented the overview of the architecture that they developed while using intelligent mobile devices capable of integrating a variety of biosensors and a Cloud that collects and processes the collected data. Pereira et al. in [10] detailed a developed platform based on MCC where Mulle sensors can interact with a mobile device which has access to the Cloud via Internet using Bluetooth. Based on RESTful web services, the framework is feasible on resource constrained devices. Soliman et al. in [11] proposed a software solution dedicated to Smart Home by integrating IoT with the Cloud computing. The methodology relies on: (1) Arduino platforms as sensor and actuator devices; (2) Zigbee technology for communication between things; (3) interactions with smart things using Cloud services; (4) JSON data format for data exchange. In addition, they demonstrate the feasibility of their approach by using some use cases such as: controlling home access or measuring home conditions.

Unlike the existing solutions that target specific domain applications when designing their middleware, our proposed work aims basically to come up with a general service architecture for Smart City applications, that combines the features of mobile devices, sensors and Cloud computing in order to offer to the user the enhanced services that are accessible anywhere while guaranteeing scalability and security. Our general service architecture is

[1] http://iotinternetofthingsconference.com.

illustrated through a framework called DropLock that we have developed to demonstrate the benefits of the new information technologies in the context of Smart City. DropLock is implemented thanks to different technologies like REST, Bluetooth, Arduino and Android. Unlike [11], our scenario based on virtual keys for home access does not require RFID tags. Moreover, the DropLock infrastructure can be used for other Smart City applications.

3. Internet of things and mobile Cloud computing

Taking full advantage of the availability of MCC facilities, the IoT can be integrated into the basic things which have the limitation of processing power, data storage and battery life. This section aims to highlight the principles behind these new concepts.

3.1. Mobile Cloud Computing

The major benefit of Cloud Computing for mobile devices [12] is the ability to run applications between resource-constrained devices and Internet-based Clouds. Hence, resource-constrained devices can outsource computation/communication/resource intensive operations to the Cloud. The principal motivation of offloading is to achieve less execution time and less energy consumption within mobile devices. Commonly, MCC means to prolong the capabilities of storage/computation-limited devices and to provide seamless access to data/application on the remote resource of rich server from anywhere. The network connectivity from the device to the Cloud server needs to be optimized to ensure the quality of service and seamless handover.

3.2. Internet of Things

The interaction between people and smart things is the basic concept of IoT. A wide variety of applications rely on enabling smart things to communicate with one another in order to exchange the information they have collected from their surrounding environment. Smart things can also serve as routing nodes to assist in communication and to forward data to terminals or to remote servers via the Internet.

3.3. IoT and MCC convergence

With its facilities to access to the Cloud, the mobile device can be used to collect data from its own sensors and from external sensors using Wifi or Bluetooth connection for example, so that data are processed locally by the mobile device or externalized for processing and/or storage by the Cloud. In this case, the mobile device acts as a service provider of sensor data *vis-à-vis* the Cloud. Many scenarios based on e-health [5, 22] for example, rely on this architecture. In fact, in these architectures, many sensors like those for temperature, high blood pressure, blood sugar, weight, etc. provide data for a patient, that are collected by the mobile device allowing the doctor to get an idea on the patient health and to contribute in the patient diagnostic. Generally, these kinds of sensors have not enough resources to process the sensed data; and the data collected by the mobile device can be stored in the Cloud for analysis or statistical purposes.

The alternative architecture is to have more sophisticated sensors with resource capabilities such as data sensing and data processing including cryptographic calculations with a low-range bandwidth connection (Wifi, Bluetooth, ZigBee, NFC). Mulle[2] and Arduino[3] sensors are examples of these kinds of rich sensors. The mobile device remains important even if the mobile device and the sensor can communicate with each other, in respect to a machine-to-

[2] http://www.eistec.se/mulle/.
[3] http://arduino.cc/.

machine model. Because the mobile device, thanks to its large-range bandwidth communication, allowing it the access to the Internet, can interact with the Cloud. Using the mobile device as a gateway, the Cloud can either collect data from the sensors or send data to the sensors. This architecture is suitable for many smart cities scenarios such as the scenario called DropLock that we take to illustrate the building blocks to compose our middleware.

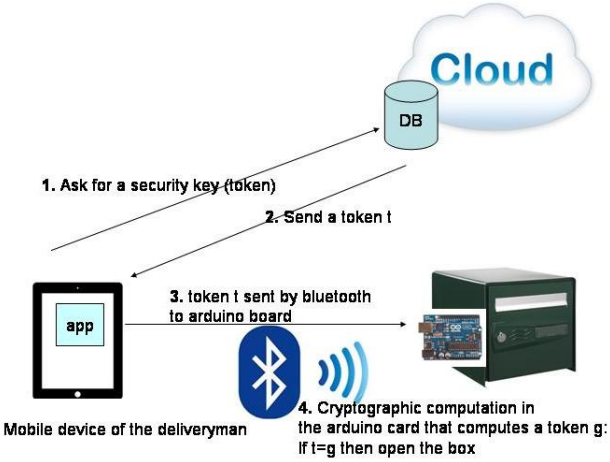

Fig.1. DropLock system.

3.4. Example of Smart City scenario: DropLock

To illustrate our approach, we have considered a use case inspired from Smart City, that is, a mechanism that relies on virtual keys instead of real keys of our daily life. Called DropLock, this framework can not only be applied for the product delivery as presented here, but also helps granting authorized access into anything that is locked and that requires unlocking like buildings, hotels, etc. The current objective of Droplock's approach is to improve the security of the service delivery from the supplier to the client while illustrating the need of MCC and IoT convergence. Currently, if a user orders a product on Amazon website for example, the delivery man will not deliver the product to the user if the latter is not available at the time the package is delivered. Our system overcomes this problem and enables the user to authorize the delivery man to access to the user mail box to deliver the product through an authentication and an authorization process provided by the Cloud to the mobile devices respectively of the user and the delivery man, in case the user is not at his home to receive the package. Once done, the mobile device of the delivery man can interact with an electronic board on the mailbox that will manage the box access. In the same context, the mailbox can be located at the post office. In this case, it is the delivery man that authorizes the user to have access to the secure mail box. Figure 1 represents the DropLock system. In the next section, we will describe the general architecture before detailing the middleware developed for DropLock.

4. System Model

In this section, we will present the general architecture of any system that relies on machine-to-machine Cloud computing model for smart cities.

The developer of these solutions must provide support to inherent requirements such as:

- Data communication methods;

- Security methods to secure the communications and control the access to the Cloud
- Resources registration methods that allow the registration of new users within the Cloud system;
- Execution on constrained environments since data processing is partly performed on sensors and mobile devices.

Based on these requirements, as can be seen from figure 2, we propose a distributed based machine to machine platform that integrates many services operating at different levels such as network management services, system management services, data management services, security and authentication services. The following details demonstrate generally the contribution of each service layer as in figure 2.

4.1. Network management services

These services offer the applications the suitable means to achieve communication between different smart entities. According to our context, the framework should handle different ranges of communication protocols.

a) Communication between smart devices: This peer to peer communication focuses on the IoT context. It may occur between the sensors using low-band communication protocols such as ZigBee or Bluetooth or between the mobile device and the sensors using Bluetooth or NFC/RFID technology.

b) Communication between the mobile device and the Cloud: The high-band communication is adopted to allow interaction between the mobile device and the Cloud thanks to wireless networks managed by telecom mobile operators.

c) Sensor discovery: The mobile device may integrate services that allow it to discover neighbouring sensors using a peer-to-peer communication.

4.2. Data management services

Data can either be sensed by smart devices and processed by the Cloud or sent by the Cloud towards the actuators. Depending on the platform, data has different roles as in the following.

a) Data collection/aggregation: The basic function of a sensor is to sense data from its surrounding environment. Using the data collection/aggregation services, a mobile terminal can collect data from sensors, process and report them using the communication service.

b) Data externalization from the mobile device to the Cloud: Because of their constraints in terms of energy, computational power and memory, smart devices may need to outsource their data storage and computation on the Cloud computing.

c) Data Centers in the Cloud: Data centers are used to store big data collected from multiple sources. They need also to be managed efficiently regarding energy management concerns.

4.3. System management services

Because smart devices are constrained environments, they host minimum services with a low footprint. On the contrary, in the Cloud computing context, system management services offer the applications the necessary services that allow task scheduling, load balancing, offloading, migration, virtualization and energy optimization. The system manager relies on distributed infrastructures (IaaS) and platforms (PaaS) to meet SaaS requirements. In addition, Sensing as a Service enables the smart devices such as sensors or mobile devices which may be physically accessible from the Cloud or virtualized so that the Cloud can access to the virtual images of these devices, hence facilitating the management of these devices.

4.4. Security and authentication services

The platform should provide security mechanisms in order to ensure trust in the platform and improve data privacy. In fact, unlike the access to the Cloud that can rely on a trusted party (called Trusted Service Manager in figure 2), the other platforms such as mobile devices and sensors are generally not trustworthy. However, mobile devices can build trusted environment by relying on hardware trusted platforms as stated in [16] such as SIM cards, Secure Elements (SE), TEE (Trusted Execution Environment). Some sophisticated sensors like Arduino offer specific libraries that implement some cryptographic algorithms, even if they don't have crypto-processors.
In our approach, the following steps must be considered:

- *The smart devices need to be authenticated* before communicating with any other platforms (sensor, mobile device, Cloud). For devices with limited resources, the authentication could be implemented as a protocol with lightweight cryptographic algorithms.
- *Data must be encrypted and sent in secure channels.*
- *Privacy*: Protecting the privacy of user data is the most important factor in Cloud computing. Thus, privacy management related to IoT, mobile devices and Cloud computing should be considered to make it possible for the convergence.

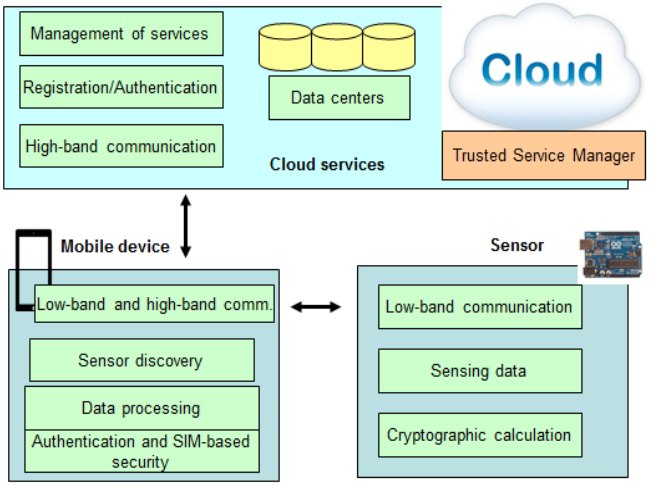

Fig.2. Service architecture for Sensor, Mobile and Cloud computing.

5. Application Scenario and Implementation

To illustrate our approach, we have considered in this paper a use case for Smart City, that is, a mechanism that relies on virtual keys instead of real keys of our daily life. Called DropLock, this framework can be applied for the product delivery as presented here, but also helps granting authorized access into anything that is locked and that requires unlocking like buildings, hotels, etc. The current objective of Droplock's approach is to improve the security of the service delivery from the supplier to the client, by demonstrating the need of MCC and IoT convergence. Generally, if a user orders a product online, the delivery man will not deliver the product to the user if the latter is not at home at the time the package is delivered. To overcome this situation, our system will enable the user if he is not at home to authorize the delivery man to access to his mail box in order to deliver the product through an authentication and an authorization process. Once done, the mobile device of the delivery man interacts with an electronic board on the mailbox that manages the box access. After the product's deposit in the secure

mailbox, the delivery man informs the user of this deposit thanks to Android apps and the Cloud. In the next subsections, we will describe the whole framework.

5.1. Our scenario

First, the user is aware of receiving a product and uses his mobile phone to authorize the delivery office to access to his mailbox, thanks to an Android application that interacts with a Cloud application. He obtains, by his smartphone, a virtual key from this Cloud application. Then, the same Android mobile device makes a Bluetooth connection with the Arduino card hosted within the box and a communication is made between the Arduino subsystem and this mobile device. So, the mobile device presents the virtual key received from the Cloud and gives this key to the Arduino card. Then, the smartphone sends a notification to the Cloud and thus, the Cloud informs the delivery man and sends him the key to his smartphone.
When the delivery man needs to access to the secure mailbox to deliver the product, he uses his Android mobile device to connect to Arduino card by providing the key. The Arduino board compares the key presented by the delivery man, and the one given by the owner of the mailbox. If they are equal, the Arduino board unlocks the mailbox.
The delivery man deposits the product, closes the door of the mailbox and informs the Arduino board. The latter locks the door and informs the smartphone of the delivery man. This smartphone sends a message of good delivery to the Cloud, which informs the user of the mailbox. When the user of the mailbox goes home, he can retrieve his product.

This functioning assumes that in the Cloud side, the owner of the secure box has previously registered the users/devices which are authorized to access to his mailbox. In fact, to send the virtual key, the Cloud checks before if the delivery man/device is allowed to access to the mail box. The One-Time-Password based security solution has been privileged because it avoids replay attacks. Figure 3 shows the work flow of the DropLock system.

5.2. Software Architecture

The objective of DropLock is to give the right to access a secure box and to ensure a secure communication between the users and the outer environment. The device's parts of the architecture are Arduino DUE, Bluetooth LE shield and smart phones acting as accessible points to the Internet. The Cloud part consists of Platform as a Service (PaaS) and Software as a Service (SaaS) for processing and storing data at the server side. Our framework is based on four subsystems: a management Arduino subsystem, a mobile management subsystem, a Cloud management subsystem and a Security management subsystem.

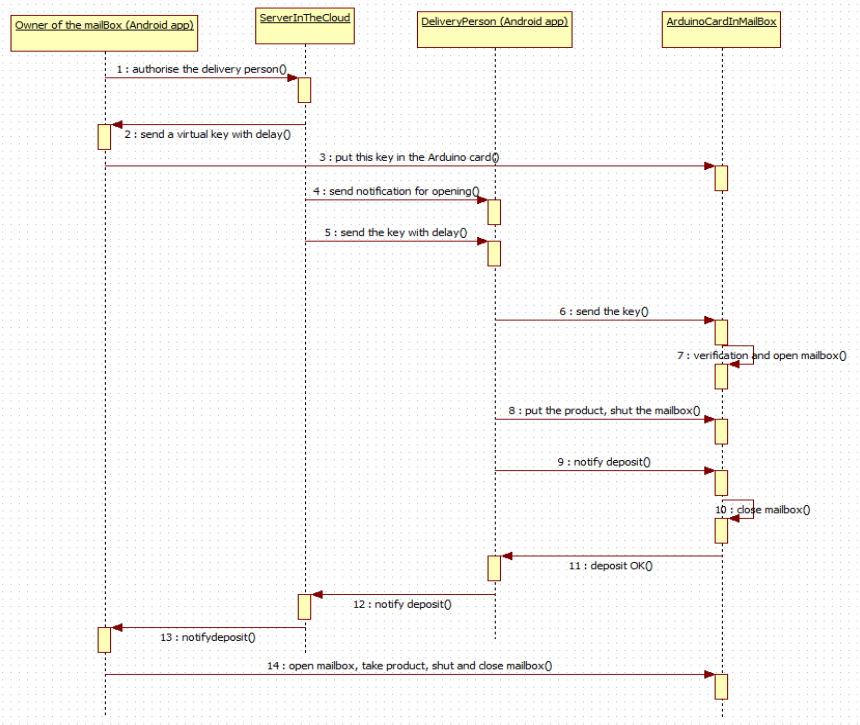

Fig. 3. System workflow.

5.3. Arduino management subsystem

Arduino platform consists of a programming environment and an Arduino board. In our framework, the main role of Arduino and its related attached shield (e.g Bluetooth) is to ensure full access and communication between the secure box and the user. In our experiment, we used the Arduino DUE[4] as it has an ARM cortex M3 CPU enabling us to implement encryption algorithms easily compared to other boards.

The Arduino management subsystem is composed of a communication module that ensures the interaction with the external environment, a battery and a physical lock all managed by an electronic board making them all to work together. Additionally, for the communication module, we used Bluetooth low energy technology (e.g RedBearLab BLE shield) to communicate with the user via his smart device. Compared to "Classic" Bluetooth, BLE[5] consumes less energy while providing reduced cost with a similar communication range. To implement BLE on Arduino, our device profile contains one service called the communication service with two functionalities. The first functionality is to send data from the board to the outside environment and the other is to receive data from outside. Each functionality is identified by its Universally Unique Identifiers (UUID). The profile described in an XML file is used by the BLE shield to advertise its two functionalities. The API developed by the shield manufacturers is used to manage the interaction between the Arduino board and the BLE shield.

At the setup phase,

- The shield is initialized and the Arduino DUE is loaded to the shield;
- The shield starts advertising for its two functionalities.

[4] Arduino DUE board specifications http://arduino.cc/en/Main/arduinoBoardDue.
[5] http://www.bluetooth.com/Pages/low-energy-tech-info.aspx

- The shield listens constantly to what comes from the outside environment.

5.4. Android mobile management subsystem

The development of the Android management subsystem is a crucial feature to the functionality of DropLock because it represents the user's key to access and manage their boxes. Bluetooth Low-energy becomes available as of Android version 4.3 (API 18) and thus that version is our target. The basic function of the Android subsystem is to manage and grant authorized users the access to the secure box (that is, a mailbox in our case) without requiring them to have or exchange the physical key. The need for the Android application is the key in fulfilling this function because, well, it replaces the physical key. By replacing the key with a smart phone as the way to open the box, the idea of what the key can be and do is opening up to many new possibilities. One possibility we sought to include is the ability for the owner of the box to remotely grant or remove the access of selected users to a box. From the Android app or the client-side, the native Android API supplies an Apache HTTP client-side library[6]. We used this library to perform our HTTP communications with the server.

To add the level of security to the information being transferred, we used HTTPS instead of just simple HTTP. Sending data between the Android client and the Cloud server could be done by using either an XML format or JSON (JavaScript Object Notation). We chose to use JSON because its structure lends itself more to the name-value of the pair structure we used for building SQL queries. In addition, the size of JSON data is generally smaller than the size of XML data.

5.5. Cloud management subsystem

In our approach, we developed a simple web API to receive the Android HTTP requests, query the MySQL database, and send back the response. The server-side code was written in PHP and based on the RESTful (REpresentational State Transfer) architecture. The RESTful architecture takes the advantage of the HTTP to provide the basic functionalities. In terms of Cloud technology, we used its advantages for data storage and processing. To be more specific, in our approach, we chose Google App Engine platform for developing, deploying and hosting our application and database.

5.6. Security management subsystem

In terms of security, DropLock implements a One-Time Password (OTP) solution [18, 19, 20 and 21] that is widely deployed by using cryptographic processing to generate a password from a synchronized parameter, a secret key, and a personal identification number. This solution can be integrated into either a mobile device or a tamper resistant hardware device like a SIM card. Different from an ordinary password, OTP solution generates a new password for each time the user logs in. OTP can be approached in two aspects: Time-synchronized OTP and Counter-synchronized OTP. In Time-synchronized OTP, the user must enter the correct password before its session is expired. In order to successfully login, the time of the authentication server and the user token must be synchronized. A counter-synchronized OTP solution synchronizes a counter between the user and the server. The security factor of this OTP solution is the counter value which increases and not reusable after an authentication attempt is made by a specific user. We have proposed to use with DropLock the Time synchronized OTP solution, which is considered as the most popular authentication mechanism.

---

[6] http://developer.android.com/reference/org/apache/http/client/package-summary.html

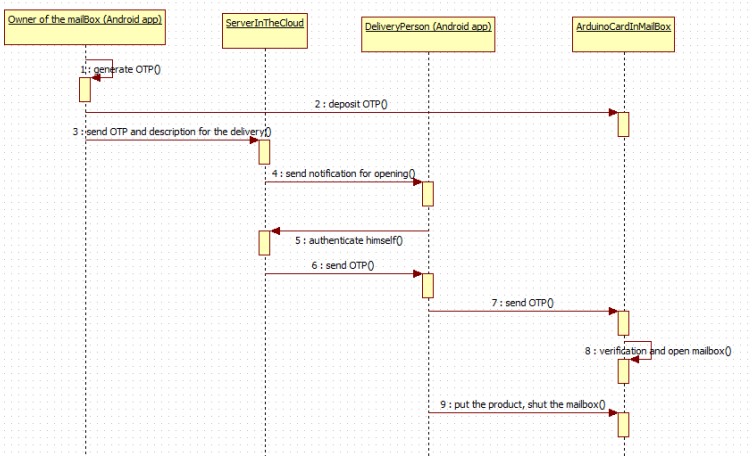

Fig. 4. Security protocol.

In our approach, we enhance the security protocol presented in figure 4, by introducing the OTP generator that is an Applet executed within a SIM card. It allows the user to generate OTPs. The DropLock application uses this OTP generator application for accessing the box when a product must be delivered. So, in this case, the DropLock application sends this OTP to the Cloud, associated with the product delivery description. This description populates a database in the Cloud which allows the delivery man to put the product into the mailbox. When the delivery man is ready to deliver the product, he sends, with his smartphone, his credentials to the Cloud for authentication. If the delivery man is authenticated by the Cloud, the OTP is sent to the delivery man's smartphone which sends it to the Arduino board. This board checks the value of this OTP and unlocks the door if the value is correct. This implemented security protocol is described in figure 4.

6. Conclusion

By proposing a general service-architecture illustrated with DropLock platform, this article highlighted the convergence between the Internet of Things and Mobile Cloud Computing. In our approach, we explored the technical features that are necessary to meet this convergence and we demonstrated the feasibility of the service architecture through the development of DropLock framework that is based on REST and BLE technologies coupled with the Android smartphone, while opening up a powerful gateway between the Cloud and Arduino platform. Even if the approach was successfully used for demonstrating services for controlling home access, the DropLock infrastructure can be used or adapted to other Smart City applications.

The next step will allow the things to connect directly to the Cloud to enhance its self-configuration and real-time interaction features. By adding current trusted platforms, we also planned to improve this approach to meet higher security requirements for other scenarios so as to be able to adapt in the near future.

References

1. Huijuan Wang, Penghua Zhu, Yuan Yu, Quanbo Yuan, « Cloud computing based on internet of things", pp. 1106- 1108, 2011 Second International Conference on Mechanic Automation and Control Engineering.
2. Geoffrey C. Fox, Supun Kamburugamuve & Ryan D. Hartman, "Architecture and Measured Characteristics of a Cloud Based Internet of Things", CTS 2012, pp. 6-12.
3. John Soldatos, Martin Serrano, Manfred Hauswirth, « Convergence of Utility Computing with the Internet-of-Things", 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 874- 879.
4. Wenying Zeng, Chao Huang, Banxiang Duan, Fagen Gong, "Research on Internet of Things of environment monitoring based on Cloud computing", International Conference on Automatic Control and Artificial Intelligence (ACAI 2012), pp. 1724- 1727.

5. Diego Gachet, Manuel de Buenaga, Fernando Aparicio, Víctor Padrón, "Integrating Internet of Things and Cloud Computing for Health Services Provisioning. The Virtual Cloud Carer Project", 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 918- 921.
6. Fei Tao, Ying Cheng, Li Da Xu, Lin Zhang, and Bo Hu Li, "CCIoT-CMfg: Cloud Computing and Internet of Things-Based Cloud Manufacturing Service System", IEEE Transactions on Industrial Informatics, Vol. 10, NO. 2, May 2014, pp. 1435-1442.
7. Xi Yu, Fuquan Sun, Xu Cheng, « Intelligent Urban Traffic Management System Based on Cloud Computing and Internet of Things », 2012 International Conference on Computer Science and Service System, pp. 2169- 2172.
8. George Suciu, Alexandru Vulpe, Simona Halunga, Octavian Fratu, Gyorgy Todoran, Victor Suciu, "Smart Cities Built on Resilient Cloud Computing and Secure Internet of Things", 2013, 19th International Conference on Control Systems and Computer Science, pp. 513- 518.
9. Jiehan Zhou, Teemu Leppänen, Erkki Harjula, Chen Yu, Hai Jin, Laurence Tianruo Yang, "CloudThings: a Common Architecture for Integrating the Internet of Things with Cloud Computing", Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design, pp. 651- 657.
10. Pablo Pu˜nal Pereira, Jens Eliasson, Rumen Kyusakov, Mia Johansson, "Enabling Cloud-connectivity for Mobile Internet of Things Applications", 2013 IEEE Seventh International Symposium on Service-Oriented System Engineering, pp. 517- 525.
11. Moataz Soliman, Tobi Abiodun, Tarek Hamouda, Jiehan Zhou, Chung-Horng Lung, "Smart Home: Integrating Internet of Things with Web Services and Cloud Computing", 2013 IEEE International Conference on Cloud Computing Technology and Science, pp. 317- 320.
12. H. XuhuiLi and Y. Zhang, "Deploying Mobile Computation in Cloud Service," in Proceedings of the First International Conference for Cloud Computing (CloudCom), 2009, p. 301.
13. G., Huerta-Canepa, & D., Lee, "A virtual Cloud computing provider for mobile devices.", in: Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, MCS'10, ACM, New York, NY, USA, 2010, pp. 6:1–6:5.
14. E. Marinelli,"Hyrax: Cloud computing on mobile devices using MapReduce,"Master thesis, Carnegie Mellon University, 2009.
15. M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The case for VM-based Cloudlets in mobile computing, IEEE Pervasive Computing 8 (2009) 14–23.
16. T. Le Vinh, S. Bouzefrane, "Trusted Platforms to secure Mobile Cloud Computing", in the 16th IEEE Internat. Conf. on High Performance Computing and Communication, Paris, August, 2014.
17. P. Parwekar, "From Internet of Things towards Cloud of Things", Intern. Conf. on Computer & Communication Technology (ICCCT)-2011, pp. 329- 333.
18. Hyun-Chul Kim, Hong-Woo Lee, Kyung-Seok Lee, Moon-Seog Jun, "A Design of One-Time Password Mechanism using Public Key Infrastructure", Fourth International Conference on Networked Computing and Advanced Information Management, pp. 18-24, 2008.
19. Jongpil Jeong, Min Young Chung and Hyunseung Choo, "Integrated OTP-based User Authentication Scheme Using Smart Cards in Home Networks", Proceedings of the 41st Hawaii International Conference on System Sciences, pp. 1-7, 2008.
20. Dinei Florencio and Cormac Herley, "One-Time Password Access to any Server without Changing the Server", Microsoft Research, One Microsoft Way, Redmond, WA, http://research.microsoft.com/pubs/78026/1569113867_final_email.pdf.pdf
21. E.Kalaikavitha M.C.A and Juliana gnanaselvi, "Secure Login Using Encrypted One Time Password (Otp) and Mobile Based Login Methodology", Research Inventy: International Journal Of Engineering And Science Vol.2, Issue 10, pp 14-17, April 2013.
22. Sriram Kailasam, Santosh Kumar, and Janakiram Dharanipragada, "Arogyasree: An Enhanced Grid-Based Approach to Mobile Telemedicine", Hindawi Publishing Corporation International Journal of Telemedicine and Applications Volume 2010, Article ID 536237, 11 pages.