IoT-DMCP: An IoT Data Management and Control Platform for Smart Cities

Sahar Boulkaboul[1,2], Djamel Djenouri[2], Sadmi Bouhafs[3] and Mohand Ouamer Nait Belaid[3]
[1]*Abderrahmane Mira University, Bejaia, Algeria*
[2]*CERIST Research Center, Algiers, Algeria*
[3]*ESI, Algiers, Algeria*

Keywords: IoT, Data Management, Big Data, Embedded Systems, Web/Mobile Application, Security, Micro-services.

Abstract: This paper presents a design and implementation of a data management platform to monitor and control smart objects in the Internet of Things (IoT). This is through IPv4/IPv6, and by combining IoT specific features and protocols such as CoAP, HTTP and WebSocket. The platform allows anomaly detection in IoT devices and real-time error reporting mechanisms. Moreover, the platform is designed as a standalone application, which targets at extending cloud connectivity to the edge of the network with fog computing. It extensively uses the features and entities provided by the Capillary Networks with a micro-services based architecture linked via a large set of REST APIs, which allows developing applications independently of the heterogeneous devices. The platform addresses the challenges in terms of connectivity, reliability, security and mobility of the Internet of Things through IPv6. The implementation of the platform is evaluated in a smart home scenario and tested via numeric results. The results show low latency, at the order of few ten of milliseconds, for building control over the implemented mobile application, which confirm realtime feature of the proposed solution.

1 INTRODUCTION

The Internet of Things (IoT) is an emerging technology in telecommunications and computer networks continuously evolving to fit various domains. This evolution will be accompanied by an evolution of the technological ecosystem in all its complexity. Indeed, this technology has nowadays taken a large part of the computer systems market due its flexibility and adaptability in several fields. The use of IoT in smart cities like other areas has been a major success. It is considered to be the extension of the Internet to things and places in the physical world and claims to model data exchanges from real-world devices to the Internet. Each object must be autonomous, able to interact and cooperate with other objects in order to achieve common goals. It has a unique address or ID, and should not consume a lot of energy. IoT-based solutions leverage advances in web and mobile applications, especially with the emergence of service-oriented architecture, to provide flexible, dynamic, adaptable, and above all, easy-to-use administrative and control applications. Software applications will be installed on mobile devices (smart phones, tablets, PCs, etc.) which will considerably increase the scope of the solutions. However, despite the wide efforts in web technologies adoption and standardization for the IoT, designing a scalable and extensible IoT platform that meets IoT functional requirements remains challenging. This has been considered in this work, where designed and implemented a data management platform to monitor and control of web/mobile applications in IoT using IPv4/IPv6 and higher layer protocols such as CoAP, HTTP, WebSocket. The platform allows anomaly detection in IoT devices and real-time error reporting mechanisms. Further, it is a standalone application and targets at extending cloud connectivity to the edge of the network with fog computing. It addresses the challenges in terms of connectivity, reliability, security and mobility of the Internet of Things through IPv6. As an instantiation of the proposed framework, we consider the smart home scenario for the implementation and tests. We thus implemented the different modules for smart home control via an IoT enabled mobile application, and made extensive tests. The results confirm realtime feature of the proposed solution and show latency at the order of few ten of milliseconds for building control actions.

Boulkaboul, S., Djenouri, D., Bouhafs, S. and Belaid, M.
IoT-DMCP: An IoT Data Management and Control Platform for Smart Cities.
DOI: 10.5220/0007861005780583
In *Proceedings of the 9th International Conference on Cloud Computing and Services Science (CLOSER 2019)*, pages 578-583
ISBN: 978-989-758-365-0

2 RELATED WORK

Even though IoT paradigm is somehow new, there are various technologies, protocols and platforms that target it. For such we can cite ZigBee, 6LoWPAN, BLE, IPv6, CoAP, MQTT, etc as known technologies and protocols used in Internet of things to enhance and evaluate the performance of product (Chouali et al., 2017),(Al-Fuqaha et al., 2015),(Tanganelli et al., 2015). Also there are different constructors of the IoT devices, which makes IoT industry more challenging. Existing IoT platforms present a variety of issues. We take as example SmartThings framework (Jia et al., 2017), it allows third party apps over privileged access to all devices capabilities, which exposes the insecurity of the framework as described in the research paper (Fernandes et al., 2016). The other known alternative taken from the open-source world is openHab (Heimgaertner et al., 2017), many parts of its installation require text configuration, potential access to log files for debugging, and other insecure practices. The configuration of openHab is therefore mainly reserved for technophiles. It is not a commercial product available on the market, ready to go. In addition openHab does not support many types of devices which present a compatibility issue. Thus, we designed our platform to take the main IoT issues and challenges.

3 SYSTEM DESCRIPTION (IoT-DMCP)

The proposed model allows heterogeneous resources, connectivity reliability and mobility, ensures security and contains application framework and platform architecture. The composition of each layer is illustrated in Figure 1 and described in the following.

3.1 Heterogeneous Resources

Since IoT ecosystems will be composed of a high range of technologies, a suitable support for the heterogeneity needs to be provided by the IoT communication architecture.In our proposed solution, we take into consideration sensors, actuators and tags.

The use a REST full API to expose every object in its part boosts heterogeneity and facilitate the extensibility of the system. In addition the wireless technologies exposed in our solution permit integration of different resources easily.

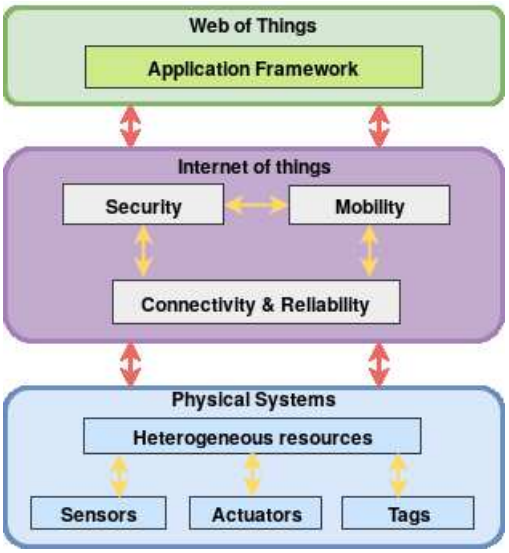

Figure 1: IOT-DMCP description.

3.2 Connectivity/Reliability

The base of the Internet of Things is providing connectivity and reliability. IPv6 is the main enabler for extending the Internet of Things to the Future Internet. This work presents how the architecture has been powered by the IPv6 connectivity in order to provide an homogeneous, scalable, and interoperable medium for integrating heterogeneous devices built on technologies such as 6LoWPAN, Bluetooth Low Energy (BLE). Using IPv6 allows each device to connect directly to the Internet and would allow billions of devices to connect and exchange information in a standardized way over the Internet.

As BLE doesn't natively communicate with IP, the best way to achieve this is to use 6LoWPAN. This simplifies the IP headers, compresses data and encapsulates the IP packets to allow them to be sent via Bluetooth efficiently, conserving bandwidth and power. The combination of BLE and IPv6 brings us much closer to the goal of having small, low-power devices that can communicate directly with each other and the Internet without using different hubs from each manufacturer sitting in the middle.

Nowadays, a variety of IEEE 802.11n wireless access-points (APs), including a dedicated commercial AP, a software AP, and a mobile router, can be used for wireless local-area networks (WLANs). Along this trend, we use the Raspberry Pi as a Gateway that interconnects beacons and also as a device to run the software AP, because it provides a cost effective, energy saving, and portable embedded system. In this paper, we present a configuration of Raspberry Pi for the software AP using IPv4/IPv6. We configure

our software AP package which provides a script that combines hostapd package, which has WPA2 support, dnsmasq and iptables for the good functioning of the access point.

Since ipv6 infrastructure is not yet deployed in some countries, the communication to an IPv6 network is not possible through an IPv4 one, our platform surpass this constraints by offering a translation mechanism by the implementation of a CoAP-http proxy that translates the http-IPv4 requests to a CoAP-IPv6 requests and the same is done for responses.

3.3 Mobility

The use of an IPv6 based architecture will permit addressing large set of devices, and make them uniquely identifiable in the internet. Further, the adoption of a micro-services architecture will address more than one object using the same ip address and differentiate between them using the URI. This will make the communication with the objects independent of their location in the globe, which improves objects mobility. Additionally the exploitation of fog computing will essentially simplify direct communication with mobile devices and therefore enhance mobility.

3.4 Security

The security of our platform is ensured using Token-based authentication, DTLS (The Datagram Transport Layer Security) protocol and passwords encryption.

3.4.1 Token-based Authentication

Token-based authentication is a very powerful concept that provides a very high level of security. It is used to manage access to system resources by using a token without the need to send authentication data whenever a resource is requested. The sequence is as follows:

1. The client requests an access token for authentication by sending an HTTP request to the authentication server that contains the user name and password.
2. A response containing an access token and a refresh token is sent by the server if the received data is correct. An error message is sent otherwise. The sent token is unique and associated to a single user. It is characterized by an expiration date after which refreshment is necessary.
3. The client uses the token to request access to a resource. The token will be included in the header of each sent request.
4. Upon receipt of a request, the server checks the validity of the token. If it is not valid, an error message is returned.

the figure below shows the exchanges between the client and the server:

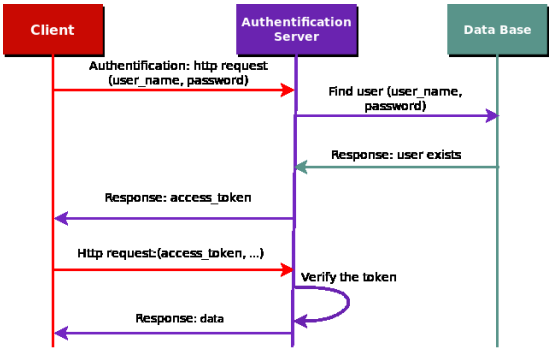

Figure 2: Token mechanism.

3.4.2 DTLS

The Datagram Transport Layer Security (DTLS) protocol (Granjal et al., 2015) provides communications privacy for datagram protocols. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery. DTLS uses two algorithms ECDHE (Elliptic Curve Diffie-Hellman Exchange) and ECDSA (Elliptic curve digital signature algorithm). In our solution, we implement DTLS in two parts: i) server: which contain implementation of DTLS in Contiki(Dunkels et al., 2004) using tinydtls(Vučinić et al., 2015) library. tinydtls has become an important tool for experimenting with DTLS in constrained devices by users from the academia as well as the industry, this part is used in the devices. ii) client : which contains implementation of scandium (Kovatsch et al., 2014) and is used in mobile application.

3.4.3 Passwords Encryption

The user database is a critical element of the system, if a hacker gets access to it he will have access to all accounts of users, hence the importance of securing the users passwords. To ensure security, there are several hash functions such as Hash function with a fixed salt, Hash function with one salt per user and Slow hash functions (Bcrypt). After studying the different hash functions, we opted for the use of Bcrypt which gives us a high level of security. Bcrypt takes approximately 100 milliseconds to execute. It's fast enough so that the user does not notice it when connecting, but still slow enough to make it extremely expensive

to run it to chop up a large list of frequent passwords. Bcrypt runs an internal encryption function multiple times, which slows down its execution. The number of loops is configurable which means that if we ever get processors or GPUs 1000 times more powerful than the ones we have today, we can just re-configure the system and increase the number of loops. This will cancel the advantage of the new processors.

3.5 Application Framework

There are a various application frameworks available for web and mobile development, each one has its specificities that make it good for use in some cases and a bad choice in some others. Because our platform aims at an IoT environment, for the development of the web platform we opted to use a Javascript stack technologies in both frontend and backend because of the advantages offered by this language such as rapidity, asynchronous requests, interoperability with other languages, simplicity and extended functionality with third party add-ons. This stack is called MEAN Stack which refers to (MongoDB, ExpressJs, Angular, NodeJs). The mobile application will exploit the backend beside the frontend. Hereafter we explain in details each part and the technologies used:

3.5.1 Backend

It uses NodeJs an open-source, cross-platform JavaScript run-time environment and exposes a Restful api with ExpressJS, to access data stored in a NoSql database managed by MongoDB, the communication with objects insured by an IoT application protocol such as MQTT(Chouali et al., 2017) and CoAP(Shelby et al., 2011), in our case we use CoAP with the implementation Node-CoAP.

3.5.2 Frontend

It uses a component oriented framework which is Angular based on typescript which is a superset of JavaScript, that allows to speed up development and facilitate code reuse.

This used technologies stack called MEAN Stack works together as follows:

3.5.3 Mobile Client

The mobile client is implemented with Java, the communication with objects is achieved using Californium, an implementation of CoAP with java. Our choice of Node-Coap and Californium CoAP implementations is based on a comparison done by (Tanganelli et al., 2015), he shows that these implementations offer all CoAP functionalities and extensions. Node-Coap is more suitable for applications WEB, its implementation in Javascript which is a fast and non-blocking language and makes it easier to integrate with WEB applications. For the mobile application, the most suitable implementation for Android is Californium which implements all the features of CoAP.

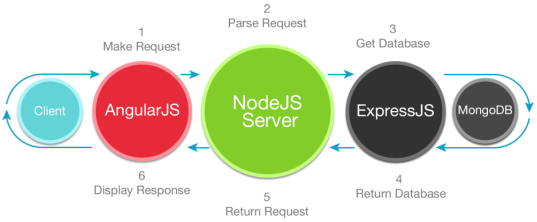

Figure 3: MEAN Stack.

3.6 The Platform Architecture

The IOT-DMCP platform architecture is depicted as shown in the figure 4 bellow,

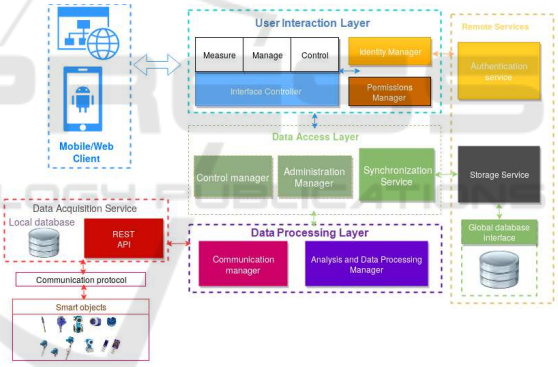

Figure 4: Platform architecture.

3.6.1 The Data Acquisition Service

Objects intercept changes in the environment and communicate the data to the data acquisition service via the CoAP protocol. This service is also responsible for translating HTTP requests from the client to CoAP requests for querying objects. The local database will permit the principal of fog computing.

3.6.2 The Data Processing Layer

It is responsible for the analysis of requests from the client, it deals with management of users and homes, and redirects the those concerning the control of objects to the acquisition service.

3.6.3 The Data Access Layer

It manages the different connections with the database(Add, update, delete), the databases synchronization and the control of objects.

3.6.4 The Presentation Layer

It represents the interface of the system with the user, it includes the mobile interface built with Android and the interface of the website built with Angular.

4 PERFORMANCE EVALUATION

Our solution is tested in a smart home environment, which provides a set of sensors and actuators including light, motion, temperature, power, fan and a lamp connected directly to a Nordic beacons nRF51822, these latter are connected using Bluetooth low energy to a Gateway represented by a Raspberry Pi, the IP6 camera is connected to the Internet. Figure 5 below illustrates the IoT hardware platform:

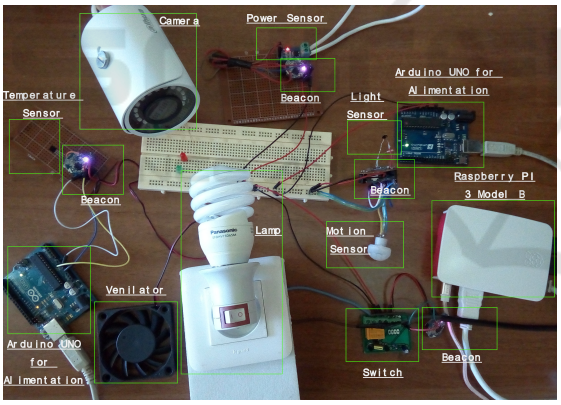

Figure 5: IoT platform.

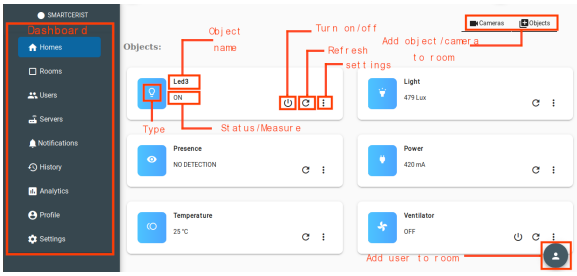

Figure 6: Web Application.

We evaluate our platform with several tests, we present the response time of objects to query GET in both web and mobile applications, the results is shown in table 1 and table 2.

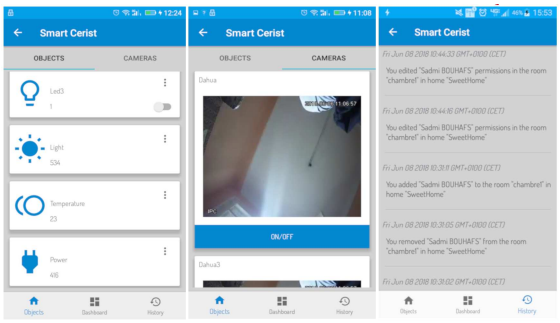

Figure 7: Mobile application.

Table 1: Web application requests delay.

Object	Test 1	Test 2	Test 3
Lamp	133 ms	141 ms	119 ms
Motion	141 ms	121 ms	111 ms
power	89 ms	98 ms	138 ms
Light	158 ms	189 ms	138 ms
Temperature	88 ms	109 ms	156 ms

Table 2: Mobile application requests delay.

Object	Test 1	Test 2	Test 3
Lamp	120 ms	118 ms	147 ms
Motion	129 ms	96 ms	111 ms
power	112 ms	162 ms	97 ms
Light	78 ms	96 ms	110 ms
Temperature	128 ms	131 ms	138 ms
Ventilator	103 ms	127 ms	127 ms

A response time shown in both mobile and web application dont exceed 190ms, according to (Miller, 1968) to evaluate the performance of our system, Results is perceived as instantaneous and confirm that our system enables realtime services .

5 CONCLUSION

In this paper, we presented an IoT data management and control platform that supports ipv6/ipv4 and takes into consideration heterogeneous resources, connectivity/reliability, security and mobility. The proposed framework is applied in smart home environment. The system allow to manage and interrogate smart objects (sensors and actuators), the collection of data in real time to follow the evolution of each object as a graph, and to apply rules allowing objects to cooperate to accomplish a specific tasks like turning light on when presence detected or start the ventilation when the temperature exceeds a certain degree. The different rules can be customized by the user. The implemented system allows the user to receive real-time notifications, and to be alerted for any changes in the

environment. An access rights management is applied to limit the actions that can be performed by a user. The security of the system is ensured by the exploitation of some techniques such as token access, DTLS, encryption of passwords and secure sessions. The platform has been used to develop a smart home control mobile application, which has been extensively tested. The results confirm realtime feature of the proposed solution and show latency at the order of few tens of milliseconds for building control actions. This work open perspectives for future works in the application we considered for the implementation. This include the integration of a virtual assistant to facilitate interaction with the system, the integration of machine learning to allow the system to learn habits of users and plan actions, and the use of artificial intelligence algorithms to enable the system to make autonomous decisions when necessary.

REFERENCES

Al-Fuqaha, A. I., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys and Tutorials*, 17(4):2347–2376.

Chouali, S., Boukerche, A., and Mostefaoui, A. (2017). Towards a formal analysis of mqtt protocol in the context of communicating vehicles. In *Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access*, MobiWac '17, pages 129–136, New York, NY, USA. ACM.

Dunkels, A., Gronvall, B., and Voigt, T. (2004). Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, LCN '04, pages 455–462, Washington, DC, USA. IEEE Computer Society.

Fernandes, E., Jung, J., and Prakash, A. (2016). Security analysis of emerging smart home applications. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 636–654, Los Alamitos, CA, USA. IEEE Computer Society.

Granjal, J., Monteiro, E., and Silva, J. S. (2015). Security for the internet of things: A survey of existing protocols and open research issues. *IEEE Communications Surveys and Tutorials*, 17(3):1294–1312.

Heimgaertner, F., Hettich, S., Kohlbacher, O., and Menth, M. (2017). Scaling home automation to public buildings: A distributed multiuser setup for openhab 2. In *GIoTS*, pages 1–6. IEEE.

Jia, Y. J., Chen, Q. A., Wang, S., Rahmati, A., Fernandes, E., Mao, Z. M., and Prakash, A. (2017). Contexlot: Towards providing contextual integrity to appified iot platforms. In *24th Annual Network and Distributed System Security Symposium, NDSS 2017, San Diego, California, USA, February 26 - March 1, 2017*.

Kovatsch, M., Lanter, M., and Shelby, Z. (2014). Californium: Scalable cloud services for the internet of things with coap. In *4th International Conference on the Internet of Things, IOT 2014, Cambridge, MA, USA, October 6-8, 2014*, pages 1–6.

Miller, R. B. (1968). Response time in man-computer conversational transactions. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, AFIPS '68 (Fall, part I), pages 267–277, New York, NY, USA. ACM.

Shelby, C. Z., draft Sensinode, I., St, I. S., Frank, A. T. B., and Sturek, D. (2011). Constrained application protocol (coap. In *IETF Internet-Draft draft-ietf-core-coap-08, work in progress http://tools.ietf.org/html/draft-ietf-core-coap-08*.

Tanganelli, G., Vallati, C., and Mingozzi, E. (2015). Coapthon: Easy development of coap-based iot applications with python. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pages 63–68, Los Alamitos, CA, USA. IEEE Computer Society.

Vučinić, M., Tourancheau, B., Rousseau, F., Duda, A., Damon, L., and Guizzetti, R. (2015). Oscar. *Ad Hoc Netw.*, 32(C):3–16.