IoT Home Gateway for Auto-Configuration and Management of MQTT Devices.

Seong-Min Kim, Hoan-Suk Choi, Woo-Seop Rhee
Department of Multimedia Engineering
Hanbat National University, 305-719 Daejeon, Korea
Email: {bigstars213, hkrock7904}@gmail.com, wsrhee@hanbat.ac.kr

***Abstract*—IoT service in home domain needs common and effective ways to manage various appliances and devices. So, the home environment needs a gateway that provides dynamical device registration and discovery. In this paper, we propose the IoT Home Gateway that supports abstracted device data to remove heterogeneity, device discovery by DPWS, Auto-configuration for constrained devices such as Arduino. Also, the IoT Home Gateway provides lightweight information delivery using MQTT protocol. In addition, we show implementation results that access and control the device according to the home energy saving scenario.**

Keywords— internet of things; auto-configuration; MQTT; DPWS; REST; IoT home gateway;

I. INTRODUCTION

The Internet of Things (IoT) is intelligent environment that connects the devices to the internet. There will be connected with more devices in the next few years [1]. The IoT is utilized for smart environment in various fields. A typical domain is a home that has many devices and appliances. And, there are many requirements about convenience and energy management. So, to satisfy these requirements, the home needs the gateway that can manage devices in its environment. But, these devices producing by different producers may cause interoperability problems because they use different communication technologies and data format [2]. This gateway should support various devices that generate massive raw data and satisfy interoperability. Also, it should be able to search device for dynamic environment. Many device management and discovery protocol are actively developed such as Devices Profile for Web Services (DPWS) and Universal Plug and Play (UPnP). DPWS protocol [3] defines a profile to enable web service messaging, discovery, description, and eventing on resource-constrained devices, which is based on Web Service Description Language (WDSL) and Simple Object Access Protocol (SOAP). But, some devices not support DPWS such as Arduino. The Arduino is open source hardware platform that can be applied in many fields and support various interface (e.g., Ethernet, Bluetooth, ZigBee, etc.). It provides Software Development Kit (SDK) and Application Programming Interfaces (API). But, DPWS is too heavy to apply on Arduino. Therefore, the user has to register the Arduino to the gateway manually. So, lightweight Auto-configuration mechanism is needed.

In this paper, we propose the IoT Home Gateway. It has the following features; various communication protocol support, device management and device information exposure. The device information exposure is the function for integrating services and devices in real world. So, we use Representational State Transfer (REST) [4] that is an architectural style, and accessible resource using unique Uniform Resource Identifiers (URI). It is available CRUD (Create, Read, Update, Delete) based on Hyper Text Transfer Protocol (HTTP). We use DPWS for discovery and messaging of devices. Also, we propose Auto-configuration for constrained devices such as Arduino. It utilizes advertisement message based on User Datagram Protocol (UDP). And, for interconnection, the devices use Message Queue Telemetry Transport (MQTT) protocol [5] that is designed for devices to use constrained networks and effectively low message overhead using publish/subscribe (Pub/Sub) messaging transport based on TCP/IP.

The rest of paper is organized follows. In section 2, we discuss related works of device registration and management. In section 3, we propose the IoT Service overall architecture, Auto-configuration and management mechanism. In section 4, we show implementation results of home energy saving service according to the target scenario. Finally we conclude our work.

II. RELATED WORKS

One of the requirements in the IoT is dynamical device discovery that searches necessary devices. The Service-Oriented Cross-layer Infrastructure for Distributed smart Embedded devices(SOCRADES) [6] is a paradigm that is the Service-Oriented Architecture (SOA) common infrastructure for integration from enterprise level to device level applications. This project has been structured in order to address the main problem that intelligent systems composed of embedded devices. It supports the DPWS that is able to discover smart embedded devices. Therefore, SOCRADES executes the real-world services running on devices with enterprise services through dynamic device discovery.

Also, in order to aggregate the device data, it requires interconnection with devices. There are two paradigms of protocols. These paradigms are resource-oriented paradigm and message-oriented paradigm. Typical mechanism of the resource-oriented paradigm is REST over HTTP. It has the

978-1-4673-9398-0/15/$31.00 ©2015 IEEE

advantage of being accessible to an external host. The [7] proposed integrating functions of real devices as REST interface. Also, it provided monitoring and controlling through the IoT Gateway.

MQTT is typical mechanism of message-oriented paradigm. It is an open application protocol that focuses on constrained devices with limited factors (e.g., low-bandwidth, limited battery). Private Assisted House (PAss) [8] provides novel service through home gateway focusing on people's needs. The Pass ensures interoperability between software application and different smart object using MQTT. Also, the [9] argued that Pub/Sub pattern of MQTT fits in home automation where more entities can be interested in the same information.

And, the QEST broker [10] proposed the broker that bridges two ecosystems with different application protocols which are REST and MQTT. QEST has highlighted the benefits of REST and MQTT. It solves the gap between things and web with two protocols, and presents a communication with each other.

In the related work, DPWS solved problem for device discovery. But, it has message form that complex and heavy. And, The related works for MQTT and REST showed advantage of the devices interconnection for each of the paradigms. So, we present lightweight device discovery mechanism for constrained devices, and divide the parts to utilize MQTT and REST for data delivery.

III. Auto-Configyration and Management

A. *IoT service overall architecture*

We propose overall architecture of IoT service as shown in the Fig.1. It consists of the IoT Home Gateway for aggregating data from devices, the Web Based Service Definition Engine for defining the user's service requirement and the IoT Service Platform for executing the service via the aggregated data and user's service requirement.

The users want to define special services that provide some functions in the particular situation. The Web Based Service Definition Engine provides web based User Interface (UI) to define services through the combination of service elements(i.e., Time, Person and Phenomenon) and abstracted device data. So, the users can create the target service through web based UI and apply with various scenarios [11].

The IoT Service Platform consists of Ontology Management, SPARQL Engine and Service Executor. To judge service condition of defined service through context around the user, the IoT Service Platform acquires data such as device information and service information through Web Based Service Definition Engine and IoT Home Gateway. So, in order to manage these data, Ontology management provides abstraction and management of data according to the ontology model [12]. And, SPARQL Engine queries the ontology to extract the semantic data. Using these semantic data, the Service Executor determines to satisfy of the service condition. And, the Service Executor will send the control request message to target devices.

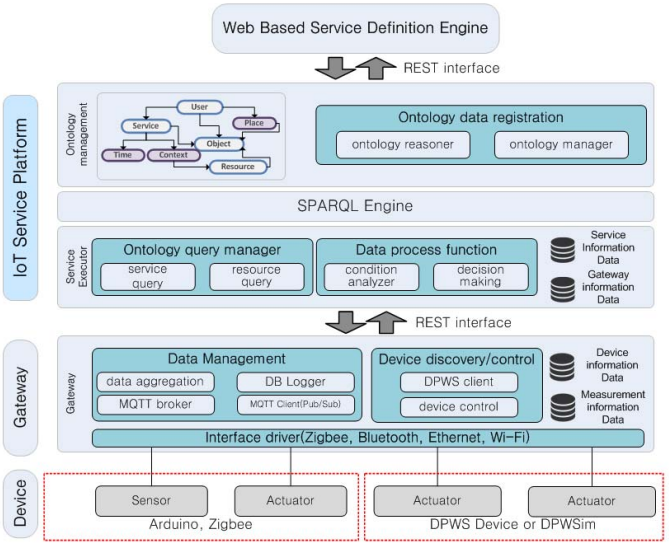

Fig. 1. IoT Service overall architecture

The IoT Home Gateway manages all devices in home environment. And, it aggregates device data using various protocols. These data are provided to the IoT Service Platform with REST interface. Also, when the IoT Home Gateway receives the control request message from the IoT Service Platform, it generates the control message and discovers target actuator. And, it sends the control message with interface of target actuator.

As a result, this architecture is able to provide interoperability with each device, device management, service definition, service execution and target device control.

B. *IoT Home Gateway*

The IoT Home Gateway is a main issue proposed in this paper. Fig. 2 is functional architecture of the IoT Home Gateway. The IoT Home Gateway consists of functions that are the Device Manager for management of device data and the DB query, the MQTT Proxy for interacting with MQTT devices, the DPWS Client for interacting with DPWS devices, the Device Discovery for registration and discovery of DPWS and MQTT devices, the MQTT Broker for sending MQTT massage, and non-IP Device Interface for supporting various non-IP protocols such as ZigBee, Bluetooth.

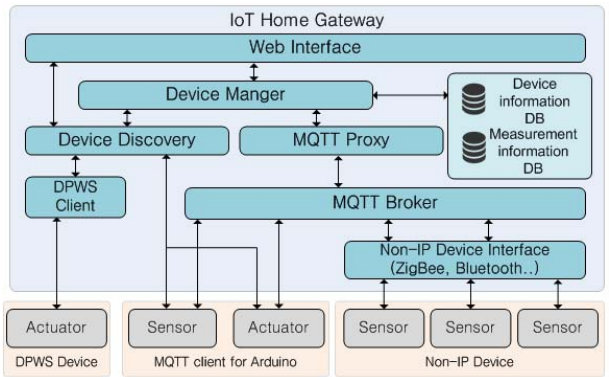

Fig. 2. IoT Home Gateway functional arcitecture.

- Device Management

  The IoT Home Gateway supports various protocols to access devices such as IP enable (Ethernet, Wifi, etc.) and Non-IP (ZigBee, Bluetooth, etc.) devices. But, raw data generated by these devices have heterogeneity such as data structure, functionality, etc. So, the reuse of sensor data and device access is difficult. To solve this problem, we propose abstraction of raw data. It converts raw data to a common data format [13] by the Device Manager function in the Fig. 2. So, it provides interoperability between the heterogeneous devices. And, to access abstracted device data easily, these data are managed on Device information Database (DB). In addition, MQTT broker accepts data of Non-IP devices in shown the Fig. 2. This is to manage and access distributed data with a common protocol. The MQTT is one-to-many communication protocol for message delivery. Also, it is lightweight and asymmetric communication protocol. So, this is suitable to accept devices using different protocols. In order to aggregate raw data with MQTT protocol, the Device Manager allocates specific topics to the devices.

- Device Auto-Configuration

  The Device Discovery supports device discovery using DPWS. The DPWS devices send their own advertisement message based on SOAP-over-UDP including device information. With the advertisement message, DPWS supports the dynamic device discovery to search required device. But, not all devices support DPWS. One of these devices is open source hardware device like Arduino. This cannot support DPWS protocol based SOAP due to constrained issue. So, for automatic registration of open source hardware devices in local network, we propose the Auto-configuration mechanism. Therefore, the devices send the advertisement of UDP in local network and the IoT Home Gateway registers automatically as detecting it. Fig. 3 shows the device registration procedure. The IoT Home Gateway assigns IP address via DHCP to target devices. And, the devices send an advertisement message like the Fig. 3(1) with UDP multicast. This message contains their device information such as Device_name, Universally Unique Identifier (UUID) and URI. It describes JSON format [15]. In the Fig. 3(2)(3), The Device Discovery in the IoT Home Gateway detects the advertisement and registers device information with the URI that is defined such as *"http://DestinationAdress:8080/GatewayREST/api/Device={UUID}/description"*. The URI provides detail device description based on JSON format. It includes Device_type, Device_name, Operation, Operation_ value, and URI. And, in the Fig. 3(4), the Device Discovery registers the devices with abstracted device data on the Device Information DB.

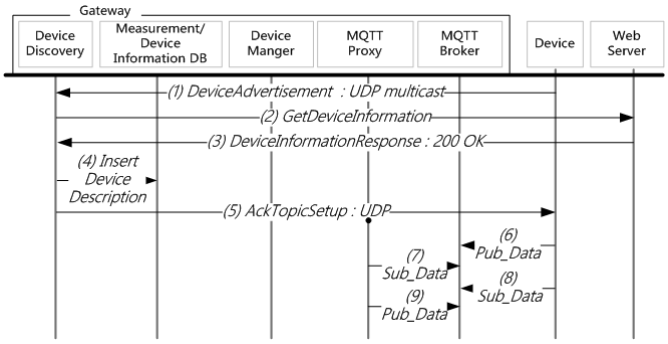

Fig. 3. Device registration procedure.

To access these registered the devices, we use MQTT protocol. The IoT Home Gateway configures the MQTT topic to Pub/Sub their interest in the message for connecting target devices to the MQTT broker. The MQTT topic is set based on the device description elements such as Device_type (Actuator, Sensor), UUID and Sensor_type (temperature, humidity, photo, etc.). And, the Device Discovery function sends acknowledgement included MQTT topics to target devices as shown in the Fig. 3(5). The device can control and sense by the Pub/Sub message based on the topic like Fig. 3(6)(7)(8)(9). The devices have different structure topics according to device types. In case of actuators, Pub topic provides status information and Sub topic receives the device control message. In case of sensors, Pub topic provides sensing values and Sub topic may need to receive messages when the user wants to change status of devices. Then, the device connects with the MQTT broker and interconnects with the IoT Home Gateway via the MQTT topic. Therefore, the Auto-configuration can dynamically register and discover constrained devices like Arduino. Also, it is the method to use light messages and ensure scalability for devices more than DPWS using SOAP. And, MQTT topics are fit by the basic frame, the IoT Home Gateway can systematically manage the topic and be convenient to control the devices. In the Fig. 4, with set topic, we show control procedure of the registered device. In the Fig. 4(1), the Device Manager in the IoT Home Gateway receives the control message from the IoT Service Platform and parses the message. Fig. 4(2)(3) checks target device in the Device Information DB. In the Fig. 4(4), the Device Manager requests checked device control. The MQTT Proxy publishes control parameter to set topics like Fig. 4(5). And, in the Fig. 4(6)(7), the device subscribes the published message and executes equipments connected to the device and subscribes device status. Finally, in the Fig. 4(8)(9)(10), it stores the device status on the Device information DB. Implementation results of the Auto-configuration will show Section 4.

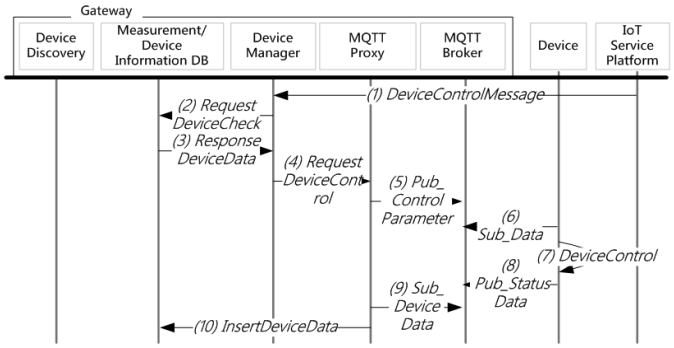

Fig. 4. MQTT Device control procedure of Gateway.

- Device Information Exposure

  The IoT Home Gateway should accept third-party's requirements. This requirements are the functions to receive aggregated device data for reusing the data and to send messages for device control. So, the IoT Home Gateway supports REST interface using GET/POST methods. It extends limited home environment into web and configures to interact between human and device. The GET method uses to provide aggregated data and the POST method uses to accept the user's requirements.

IV. Implementation

A. Home Energy Saving Scenario

In this section, we present results of implementation according to target scenario, which is home energy saving services. In the Home, there are many appliances (e.g., TV, water purifiers, bidets, set-top boxes, etc.) that use standby power. It takes most part of energy consumption. So, we need to reduce the standby power. And, the energy saving target scenario is as follow:

The user does not have much time to stay at home. the user wants to reduce the unnecessary standby power when the user is not at home.

In order to show a service according to this scenario, we compose the home energy saving environment as Fig. 5.

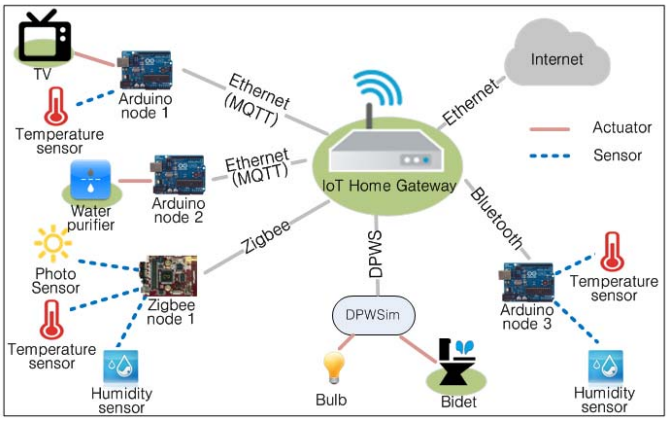

Fig. 5. Home Energy Saving Environment.

B. Implementation Result

The implementation environment uses Java EE (WS4D, Java client from Paho) and Apache Tomcat 7.0. Also, it uses the open source Mosquitto [16] as MQTT broker. And, the devices use Arduino and DPWSim [17]. DPWSim supports virtual simulation for DPWS device.

First, we show the Auto-configuration of Arduino device which acts as a MQTT client. DPWS device sends the advertisement message with SOAP-over-UDP multicast (239.255.255.250:3702). Utilizing this way, the Arduino device sends the advertisement message using UDP multicast with specific port. The advertisement message describes device information including device name and URI as shown in the advertisement message, and the URI provides detail device information as shown in the device description message in the Table 1.

TABLE I. Device Configuration Messages

Advertisement Message	{
	"name":"arduino1",
	"URI":http://localhost:8080/GatewayREST/api/Device=550a-8400-41d4-0001/description
	}

Device Description Message	{
	"Device_Description":{
	"Device_type":"Actuator",
	"Device_name":"550a-8400-41d4-0001",
	"operation_value":
	{
	"TV_On":"On",
	"TV_Off":"Off"
	},
	"operation":["TV_On","TV_Off"],
	"URI":"http://192.168.0.13/GatewayREST/api/Device=550a-8400-41d4-0001/Description"
	}
	}

Fig. 6 is the result of Auto-configuration. Top of Fig. 6 is the result of registered Arduino device information on the Device Information DB. Bottom of Fig. 6 shows the result of MQTT topics setting via the acknowledgement of the IoT Home Gateway. After Arduino sets up a MQTT topic, it is connected with the MQTT broker. The MQTT topic is provided differently depending on device operation and device type that stored in Device Information DB as follows.

In case of actuators, Pub provides status information with *"Actuator/status/{UUID}"* and Sub receives device control message with *"Actuator/control/{UUID}"*.

In case of sensors, Pub provides sensing values with *"Sensor/{UUID}/{Sensor_Type}"* and Sub may need to change status of device. Topic of the sub is *"Sensor/{UUID}/sub"*.

Therefore, the IoT Home Gateway can control the MQTT devices with device operation and UUID.

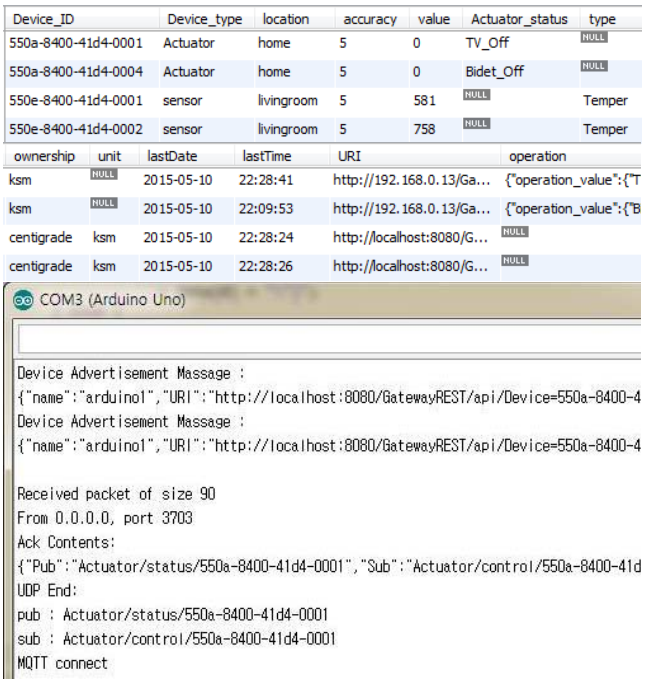

Fig. 6. Auto-Configuration result (Arduino based MQTT client).

When the Arduino device connects to MQTT broker, we can control the connected device as shown in the Fig. 7. It shows the standby power control of TV using the device control message with POST method of REST. The IoT Home Gateway received the control request message from the IoT Service Platform as shown in the C of Fig. 7. MQTT Proxy publishes the *Off* of control parameter in the subscribed topic of Arduino. The B, D of Fig. 7 are results of the delivered control message in Arduino and turn off TV

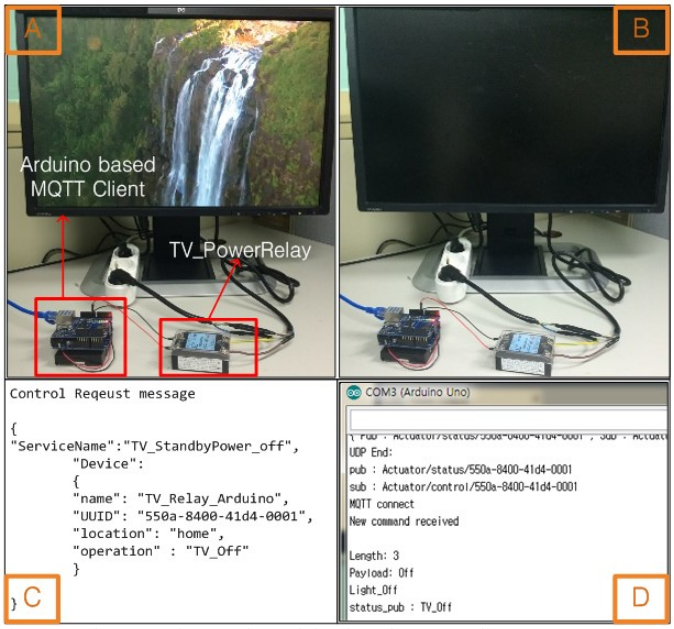

Fig. 7. Device operation result (Arduino based MQTT Client).

Likewise, the IoT Home Gateway receives the control request message as shown in the Fig. 8. Then, the virtual devices of DPWSim operate as shown in the B of Fig. 8. This turn off a bidet in toilet according to the scenario. The D of Fig. 8 is the results of debugging that shows invoke messages.

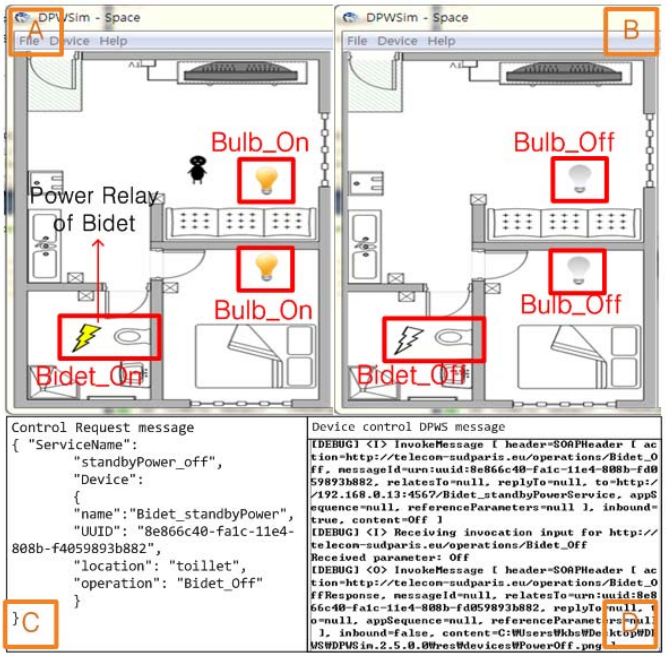

Fig. 8. Device operation result (Actuators based DPWsim).

Finally, we provide required information to third-party and other IoT service platform as shown in the Fig. 9, and the IoT Home Gateway provides abstracted device data with REST interface using JSON format. The message is used to deliver to the IoT Service Platform. There are two ways that the home gateway provides the forms. Top of Fig. 9 includes device information and information of actuator enables on/off power of TV. Bottom of Fig. 9 is value of sensor to minimize size.

Fig. 9. Abstracted device data using REST interface.

V. CONCLUSION

In this paper, to satisfy the requirements of the user about convenience and energy management in home, we proposed the IoT service overall architecture. The proposed architecture consists of the IoT Home Gateway for aggregating data from devices, the Web Based Service Definition Engine for defining the user's required services and the IoT Service Platform for executing the service via the aggregated data and the user's defined services. The proposed IoT Home Gateway provides device management to remove heterogeneity of various devices, the Auto-configuration for dynamic device discovery and the device information exposure to provide required information to third-party and other IoT service platforms. Also it supports discovery of constrained devices such as Arduino by the Auto-configuration mechanism. As a result, we showed implementation results that control various devices according to home energy saving scenario.

ACKNOWLEDGMENT

This research was supported by the research fund of Hanbat National University in 2014 and Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(grant number: NRF-2014R1A1A2059500).

REFERENCES

[1] Ericsson, “More than 50 Billion Connected Devices,” White Paper, February 2011, [Online]. Available: http://www.ericsson.com/res/docs /whitepapers/wp-50- billions.pdf.

[2] Rossello-Busquet, A., Soler, J., Dittmann, L., "A Novel Home Energy Management System Architecture," Computer Modelling and Simulation (UKSim), 2011 UkSim 13th International Conference on, pp. 387-392, March 2011.

[3] DPWS OASIS Standard, “Devices Profile for Web Services Version 1.1,” [Online]. Available: http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/

[4] R.T. Fielding, R.N. Taylor, “Principled Design of the Modern Web Architecture,” ACM Trans. Internet Technology, vol. 2, no. 2, pp. 115-150, May 2002.

[5] MQTT, OASIS Standard, [Online]. Available: http://mqtt.org/

[6] A. Cannata, M. Gerosa, and M. Taisch, “SOCRADES: A framework for developing intelligent systems in manufacturing,” in Industrial Engineering and Engineering Management, IEEE International Conference on, pp. 1904 –1908, December 2008.

[7] D. Guinard, V. Trifa, “Towards the Web of Things: Web Mashups for Embedded Devices,” Proc. Workshop Mashups, Enterprise Mashups and Lightweight Composition on the Web, April 2009.

[8] Private Assisted House (PAss), [Online]. Available: http://www.projectpass.eu

[9] M. Tucic, R. Pavlovic, I. Papp, and D. Saric, “Networking layer for unifying distributed smart home entities,” In Telecommunications Forum Telfor (TELFOR), pp. 368-371, November 2014.

[10] M. Collina, G.E. Corazza, and A. CorazzVanelli-Coralli, "Introducing the QEST broker: Scaling the IoT by bridging MQTT and REST," Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on, pp. 9-12, September 2012.

[11] J. Y. Lee, H. S. Choi, N. R Yang, W. S. Rhee, "UCSE: User created service engine based on MVC model for IoT service," TENCON 2014, IEEE Region 10 Conference, pp. 1-6, October 2014.

[12] N. R. Yang, H. S. Choi, J. Y. Lee, Y. J. Kim, W. S. Rhee, "GOMs: Generic ontology models to process context information in IoT environment," TENCON 2014, IEEE Region 10 Conference, pp. 1-6, October 2014.

[13] H. S. Choi, J. Y. Lee, N. R. Yang, W. S. Rhee, "User-centric service environment for context aware service mash-up," Internet of Things(WF-IoT), 2014 IEEE World Forum on, pp. 388-393, March 2014.

[14] JavaScript Object Notation (JSON), [Online]. Available: http://tools.ietf.org/html/rfc7159/

[15] Mosquitto, [Online]. Available: http://mosquitto.org/

[16] S. N. Han, G. M. Lee, N. Crespi, N. V. Luong, K. W. Heo, M. Brut, and P. Gatellier, "DPWSim: A Simulation Toolkit for IoT Applications Using Devices Profile for Web Services," Internet of Things (WF-IoT), 2014 IEEE World Forum on, pp. 544-547, March 2014.