A Device Software Platform for Consumer Electronics Based on the Internet of Things

Jaeseok Yun, *Member*, IEEE, Il-Yeop Ahn, Nak-Myung Sung, Jaeho Kim

**Abstract — ***The highly-fragmented and non-standardized landscape of the Internet of Things industry results in forcing both IoT developers and end-users to have to choose their proprietary consumer electronics by a company, eventually becoming a barrier to build an unfragmented IoT ecosystem. This paper proposes an oneM2M standards-compliant device software platform for consumer electronics based on the Internet of Things, called &Cube. It leverages a standardized resource model and REST APIs to work with oneM2M service platforms, leading to interoperability across various IoT consumer electronics built on the &Cube. The growing adoption of the &Cube in consumer electronics will help lower the barriers for the manufacturers and developers to create innovative products and entirely new services*[1].

Index Terms — Internet of Things, IoT consumer electronics, device software platforms, IoT middleware, oneM2M standards.

I. Introduction

The Internet of Things is rapidly becoming part of daily life, in particular, emerging as various Internet-connected consumer electronics around people. For example, it is possible to buy a WiFi-enabled, programmable thermostat with which the user can dynamically set its configuration like desired temperature setting and weekly heating/cooling schedules via a smartphone app. Similarly, a few conventional lighting system companies produce lighting solutions where a lightbulb is incorporated with a radio chip (e.g., ZigBee transceiver), and communicates with its gateway, providing an innovative way of interacting with users−turn on/off or change brightness/color through a smartphone at anytime and from anywhere.

As these examples demonstrate, the term Internet of Things (IoT), coined by Ashton in 1999 [1], has been an emerging technological trend in recent years in a wide range of vertical domains including consumer electronics. The IoT represents a technological revolution where all the objects in the real and virtual world can be connected each other through the Internet due to several technological advances including identification and contactless data exchange (RFID and NFC), distributed sensor networks, short-range wireless communication (ZigBee and Bluetooth), and universal mobile access (cellular network and WiFi hotspot). Internet connectivity for the world objects allows them to share the information about the change in their status and surroundings, and enables them to become reactive to external stimuli, providing smart services for their users in a proactive and intelligent manner.

However, the fast-growing IoT-based consumer electronics marketplace has been fragmented due to divergent IoT device software platforms mainly dominated by competitive driving forces. For example, the programmable smart thermostat and lighting systems explained above rely on their own software platforms. Indeed, future IoT consumer electronics will have a wide variety of heterogeneous hardware systems, ranging from small, low-power nodes having constrained resources such as battery-powered, short-range wireless sensors to full-fledged computing machines such as smartphones. These systems will also rely on different network protocol stacks and proprietary service layer platforms. As a consequence, the IoT consumer electronics marketplace will suffer from massive fragmentation problem, thereby hindering the creation of innovative products and entirely new services based on IoT-related technologies. In order to mitigate this situation, several standard bodies and company alliances have come together and been working on establishing software platforms providing a unified network, protocol, and service layer capabilities.

This paper demonstrates a device software platform, called &Cube, for IoT consumer electronics that are compliant with a widely-deployable IoT standards, oneM2M. The &Cube uses a modular architecture to support interworking with oneM2M service platforms using REST APIs, as well as a thing-specific layer, called TAL (thing adaptation layer), in order to facilitate the employment of a diverse range of ‘things’, i.e., sensors and actuators required for realizing consumer electronics functions. To present the potential use of the &Cube in developing IoT consumer electronics and services, several prototype IoT devices and a smart home service have been implemented.

The remaining of the paper is organized as follows: Section 2 introduces various device software platforms for consumer electronics. Section 3 summarizes the requirements analysis

[1] This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.B0184-15-1003, Development of oneM2M Conformance Testing Tool and QoS Technology).

Jaeseok Yun is with the Korea Electronics Technology Institute (KETI), Gyeonggi-do 463-816, Korea (e-mail: jaeseok@keti.re.kr).

Il-Yeop Ahn is with the Korea Electronics Technology Institute (KETI), Gyeonggi-do 463-816, Korea (e-mail: iyahn@keti.re.kr).

Nak-Myoung Sung is with the Korea Electronics Technology Institute (KETI), Gyeonggi-do 463-816, Korea (e-mail: nmsung@keti.re.kr).

Jaeho Kim is with the Korea Electronics Technology Institute (KETI), Gyeonggi-do 463-816, Korea (e-mail: jhkim@keti.re.kr).

Contributed Paper
Manuscript received 09/02/15
Current version published 12/29/15
Electronic version published 12/29/15.

for IoT device software platforms. Section 4 and 5 present the &Cube design and implementation. Section 6 presents the prototype IoT devices/service. Section 7 discusses remaining challenges, and Section 8 offers concluding remarks.

II. Related Works

Device software platforms for consumer electronics can be considered a middleware for supporting Internet connectivity across heterogeneous hardware systems. Middleware has been a long-standing research topic, especially in communications, because it can solve heterogeneity problem in a wide variety of hardware systems running different operating systems [2].

Middleware for interoperable computing systems, networks, and home appliances extended to consumer electronics [3]-[6]. Bae *et al*. presented an adaptive middleware for heterogeneous home networks to support seamless interoperability [7]. Uhm *et al*. developed adaptive middleware for context-aware light systems [8] and portable intelligent gateways [9]. Hwang *et al*. also created lightweight middleware-based lighting systems in smart homes [10]. Additionally, many types of middleware for consumer products have been developed, including set-top box [11], hand-held device [12], and homecare gateway [13]. Device driver for consumer electronics is also a research area of interest [14]. Recently, service-oriented architecture (SOA) and service-oriented standards have been studied for high-level interoperability with distributed embedded devices [15], [16]. Kasai illustrated a middleware for a wide range of embedded operating systems, and a software development kit with open APIs for distributed mobile cache system [17]. He emphasized that it could spur third-party developers to produce various applications, which is one of the motivation of this paper.

As in the research domains exemplified above, IoT systems also need IoT middleware platform that can abstract and adapt heterogeneous hardware, software, and services [18]. Several extensive surveys summarized M2M/IoT platforms, services, and enabling technologies, including IoT middleware [19]-[23]. In particular, semantic middleware for IoT systems has been developed for service interoperability and composability [24], [25]. Kranz *et al*. demonstrated ElToolkit that supports different transmission protocols and hardware devices, making application development easier in embedded interaction with IoT-based common objects [26]. Roalter *et al.* also proposed a robotic middleware for intelligent environments and IoT [27].

Recently, Hasan and Curry demonstrated ‘thingsonomies’ for semantic normalization in an event-based middleware to achieve scalability and interoperability in the IoT environment [28]. Mrissa *et al*. presented a software platform, called an ‘avatar’ [29]. It provides virtual abstraction to extend physical objects via Web languages and semantic annotation, obtaining autonomous behavior and collaboration between them.

In short, most work on creating IoT middleware is focused on semantic annotation and abstraction of devices and services for interoperability and scalability. However, it also needs to consider employing emerging IoT-related standards in creating IoT middleware. It is clear that devices running a standardized middleware could be easily collaborated together at low levels, though adding semantic technologies will help it at high levels. It also needs to consider software platforms for IoT consumer electronics that can help developers create easily and rapidly new products with little effort on hardware and software.

III. Requirements for Device Software Platforms of IoT Consumer Electronics

With an extensive literature survey and analysis on various existing software platforms, requirements for device software platforms to support future IoT consumer electronics can be summarized as follows: scalability, modularity, adaptability, portability, interoperability, and security & privacy.

A. Scalability

An IoT service platform (i.e., server) will have connections with a huge number of IoT consumer electronics, and provide them with common service functions, including registration, data and device management, discovery, security, and so on. It implies that a well-designed device software platform for IoT electronics needs to be harmonized with IoT service platforms via standard architecture and network interfaces, being able to scale to handle hundreds of thousands consumer electronics.

B. Modularity

Like most middleware for distributed systems, the modular architecture could allow developers to separate concerns when they build applications for IoT electronics. It will also increase reusability and manageability while maintaining independence between components, improving development efficiency.

C. Adaptability

IoT consumer electronics will be composed of a wide range of things, e.g., sensors and actuators. Thus, a desirable device software platform may need to provide a transparent way in which developers will be able to naturally incorporate things into IoT products, and an efficient way of converting collected data into the message format (e.g., XSD) its corresponding IoT service platform can receive and understand, and vice versa.

D. Portability

A desirable IoT software platform needs to allow portability across various hardware and operating systems by providing developers with an abstraction layer that masks the complex underlying systems and instead exposes a set of useful APIs to applications. Such independence of device software platforms will help developers focus only on developing service logics (i.e., applications).

E. Interoperability

Some consumer electronics may be more efficiently worked when they can share their status and surroundings each other.

For example, an interoperable set of alarm clock, toaster, and coffee maker would be able to prepare a fresh breakfast in wake-up time. Such a scenario could be realized, provided the home appliances are built on a common software platform.

F. Security & privacy

Internet-connected electronics will inevitably raise concerns about security and privacy, probably impeding the widespread adoption. Thus, it needs to incorporate a robust and reliable security solution into the IoT electronics platform in order to protect systems and data from likely hacking and cyber-attacks.

IV. &CUBE ARCHITECTURE

From the requirements analysis, a new general-purpose and standardized software platform for IoT consumer electronics, called *&Cube*, has been developed. The &Cube is designed to comply with a well-known machine-to-machine (M2M) and IoT standards, *oneM2M* so that it can naturally offer a global interoperability across oneM2M-compliant service platforms. The &Cube employs a modular architecture together with an intermediate layer, called the *thing adaptation layer* to support data conversion between the &Cube and things embedded into consumer electronics.

A. oneM2M Standards and Reference Model

The oneM2M is a global standards initiative established in 2012 by seven standards development organizations (SDOs) and five industrial consortia. The oneM2M's objective is to develop a standardized technical specification considering IoT/M2M software platforms for globally-applicable, access-independent IoT/M2M services. Therefore, the &Cube is designed to comply with the oneM2M reference model.

Swetina *et al*. summarized well the oneM2M standards such as the requirements, system architecture, protocols, security, and device management and abstraction [30]. The oneM2M architecture defines a layered model for supporting M2M and IoT services comprising application layer, common services layer, and underlying network services layer, each of which is respectively supported by the following functions: application entity (AE), common services entity (CSE), and underlying network services entity (NSE) (see Fig. 1). An AE represents an entity in the application layer residing in a number of nodes and providing various service logics. A CSE stands for an instantiation of a set of common service functions (CSFs) of the IoT environment. An NSE provides network services from the underlying network to the CSEs. The reference model divides IoT environments into two domains (infrastructure and field domain), and defines four types of nodes: infrastructure node (IN), middle node (MN), application service node (ASN), application dedicated node (ADN). The more thorough description is out of scope of this paper, and available in the oneM2M standards documents. Since the target consumer electronics could be either gateways or devices, the &Cube is designed to serve as MN-CSE as well as ASN-CSE.

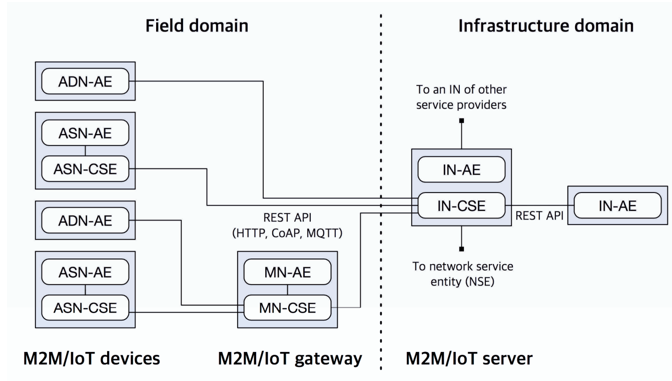

Fig. 1. The oneM2M architecture reference model.

B. &Cube Design

Fig. 2 presents a block diagram of the overall system for IoT services consisting of IoT service platform (i.e., IN-CSE) and the &Cube running on IoT consumer electronics. The IN-CSE needs to offer standardized capabilities to connect and interact with hundreds of thousands IoT devices embedded with the &Cube, as well as REST (representational state transfer) open APIs with which various IoT applications (i.e., IN-AEs) can be created rapidly and efficiently by IoT application developers.

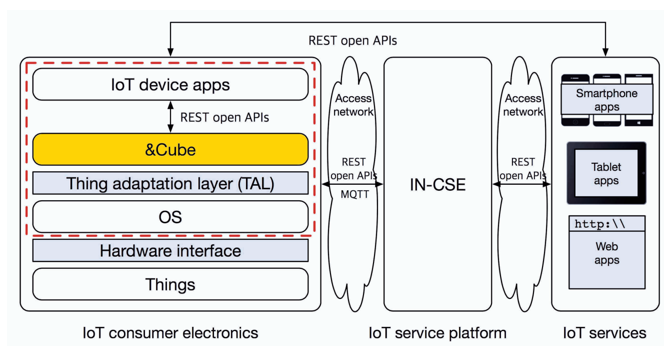

Fig. 2. A block diagram of overall system for IoT services via the &Cube running on IoT consumer electronics.

As shown in the layered architecture in Fig. 2, it assumes that target IoT consumer electronics would consist of a set of things (e.g., sensors and actuators) to provide their functions, for instance, LEDs and motors for washing machines. Also, it assumes that control systems for consumer electronics (e.g., microcontroller systems) are integrated with an operating system (OS), and supported with device drivers and relevant APIs through underlying hardware interfaces for the embedded things. Based on the assumption, a device software platform for IoT consumer electronics, &Cube, and the thing adaptation layer are proposed. The internal organization (red dashed rectangle in Fig. 2) of the proposed device platform and thing adaptation layer is shown in Fig. 3 in detail.

Considering the requirements analysis described in Section 3, the &Cube is designed to specifically address the needs of IoT device platform for consumer electronics, consisting of six components: *interaction manager, resource manager, thing manager, security manager, device manager, and application* *manager*. Fig. 4 presents the main functionalities provided by each &Cube component. The key features of the &Cube include support for oneM2M reference models (i.e., serve as MN-CSE or ASN-CSE) and standardized REST open APIs for interworking between IoT service platforms (i.e. IN-CSE) and devices (i.e., ASN-CSE or ADN-AE). Using the REST APIs, consumer electronics manufacturers would develop various device apps corresponding to the device's functions, e.g., rich washing features for a washing machine like hand wash, wool wash, and quick wash.

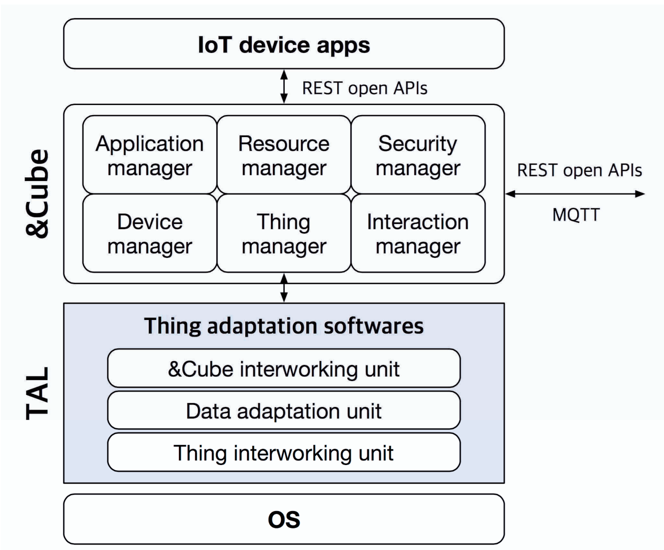

Fig. 3. High-level architecture for the &Cube and thing adaptation layer.

Interaction manager	Support interaction with IoT service platforms via REST open APIs, Register/deregister IoT electronics, upload data, receive request, Support protocol bindings (HTTP, CoAP, MQTT).
Resource manager	Support the resource architecture of the oneM2M standards, Response to resource request from the other components, Trigger other components based on changes in resources.
Thing manager	Manage the profile and connection of things embedded into IoT consumer electronics, Support interworking with the thing adaptation layer (TAL).
Security manager	Support security and authentication process, Support encoding and decoding of data.
Device manager	Manage the other components, Support platform update, reboot, and shutdown, Start components and request registration at first boot.
Application manager	Manage IoT device apps, Support download/start/stop/remove device apps.

Fig. 4. Main functionalities of six components of the &Cube.

The thing adaptation layer (TAL) serves as an intermediate level for bridging between the &Cube and embedded things. It would be filled with thing-specific *thing adaptation softwares* (TAS), each of which is intended to support interworking between the thing manager of the &Cube and a specific thing embedded into the consumer electronics, as shown in Fig. 3. The main role of a TAS is data adaptation, that is, converting data collected from a given thing into the standardized format the &Cube can understand for further processing. Similarly, a TAS supports translating a command message sent from the &Cube into a corresponding control command for the thing. This adaptation process, of course, may be implemented within the &Cube. However, through the introduction of the TAL and TAS, code reusability could be improved, i.e., a TAS for a particular thing could be reused without modification across distinct consumer electronics that employ the thing. The TAL-based architecture can also at some level allow developers to implement the device's functions independent of thing vendors with proper OS-specific device driver and relevant APIs.

V. &CUBE IMPLEMENTATION

A. *Software Implementation*

The &Cube has been implemented as a Java program so that it could be easily ported on any type of embedded machines installed with the Java virtual machine (JVM). As shown in previous works [31]-[33], Java-based implementation allows natural portability across widespread embedded systems and distinct OSs, allowing developers to create new IoT consumer electronics rapidly and efficiently. In the interaction manager, a lightweight web server is implemented for HTTP protocol, and a MQTT client is implemented for Pub/Sub messaging.

A TAS can be implemented as a program that runs on the embedded machine using the device drivers and the libraries of a programming language (e.g., C, C++, or Java) supported by thing vendors. As explained in the previous section, a TAS developed for a particular thing can be naturally reused by only compiling again the code, without having to modify it.

Finally, it has been decided to publicly open source code for the &Cube in the OCEAN (Open allianCE for iot stANdard), an open-source based global partnership for IoT industry verticals. It opens the &Cube source code and binary files, TAS and device app samples. Developers can freely modify and share their works under the BSD3-clause license.

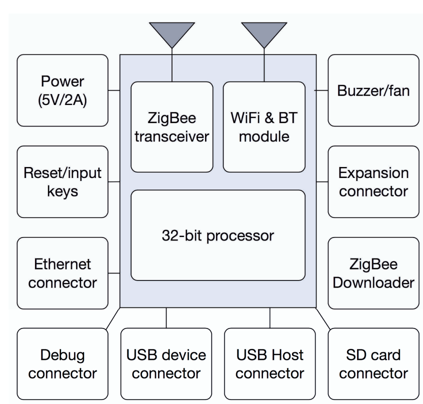

Fig. 5. The block diagram of the IoTGW.

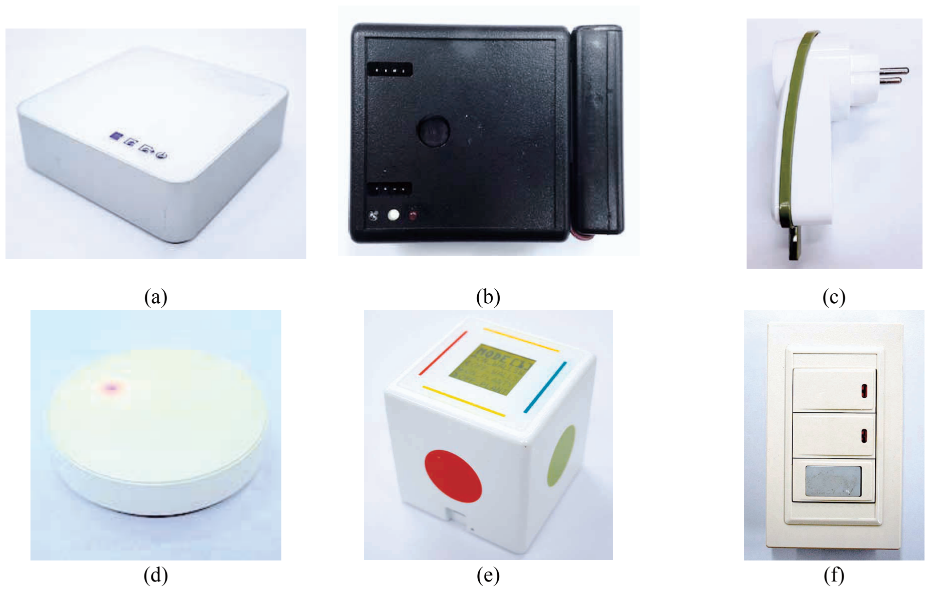

Fig. 6. The IoTGW container and prototype IoT devices; (a) IoTGW container, (b) multi-functional sensor for measuring temperature, humidity, light, and occupancy, (c) smart plug, (d) wireless switch, (e) smart dice, (f) wall light switch.

The IoTGW can work as an IoT gateway to IoT consumer electronics, as well as short-range, low-power wireless sensor nodes by providing Internet connectivity and then interacting with IoT service platforms. Of course, the IoTGW can also be equipped into an IoT consumer electronics, and thus enable itself to be a standalone IoT system, that is, can work directly with IoT servers (e.g., Internet connected washing machine)..

VI. Prototype IoT Devices and Services

To show the potential use of the &Cube in developing IoT consumer electronics, instead of developing a new consumer electronics, several prototype IoT-based devices (i.e., things with wireless connectivity) have been implemented, which would be able to be embedded into consumer electronics.

A. *Prototype IoT Devices*

IoT devices could be any 'thing' that will be equipped into consumer electronics. Specifically, three sorts of IoT devices have been created: a multi-functional sensor, a smart plug, and three user input devices, as shown in Fig. 6(b-f). First, multi-functional sensor modules have been implemented using temperature, humidity, light, and human presence sensors (Fig. 6(b)). Next, smart plugs have been developed, which can be used as both sensors and actuators having a power relay (Fig. 6(c)). A smart plug can measure electrical power used in a consumer electronics plugged in, and instantly turn on/off it by sending a command signal. The design and implementation of the smart plug is described in detail in the previous literature [34]. Finally, for easy-to-use, user-friendly input interfaces, wireless switches (Fig. 6(d)), smart dice (Fig. 6(e)), and wireless wall light switches (Fig. 6(f)) have been implemented. Wireless switches command smart plugs to turn electrical devices on or off just like a remote controller. The smart dice is embedded with a 3-axis accelerometer, and provides four different inputs by positioning each sensitive axis (X, Y) of the accelerometer at +G and –G, each of which can command smart plugs or another devices to perform a task. The smart dice could also change its mode with simple shaking gestures, and assign another set of four commands to devices. It has a small LCD panel that shows its mode and the description of four commands assigned (e.g., 'turn a lamp on'). For wireless wall light switches, an existing IR remote-control wall light switch is modified to be connected to the IoTGW. All the IoT devices employ a rechargeable Lithium-ion or AA battery for power source, and an IEEE 802.15.4 ZigBee transceiver for wireless connectivity. For interacting with the &Cube, a set of corresponding TASs have been developed for the IoT devices.

B. *IoT Service Development*

A possible scenario in the IoT era is that everyday objects at home, including home appliances and utilities will be able to be connected to each other, and therefore share their status and changes in the surroundings. Based on the interoperable IoT-based infrastructure and Internet-connected home appliances, a smart home service can be created, where all the consumer electronics ranging from the front door (e.g., door and window locks controllable by a smartphone app) to the basement (e.g., energy utilities helping avoid peak-time energy use) can work together cooperatively to perform daily tasks without explicit human intervention, and respond proactively to user needs.

Accordingly, smart home services would be a best example for showing the practical availability of the proposed software platforms and devices, in particular in developing IoT-based consumer electronics and relevant applications.

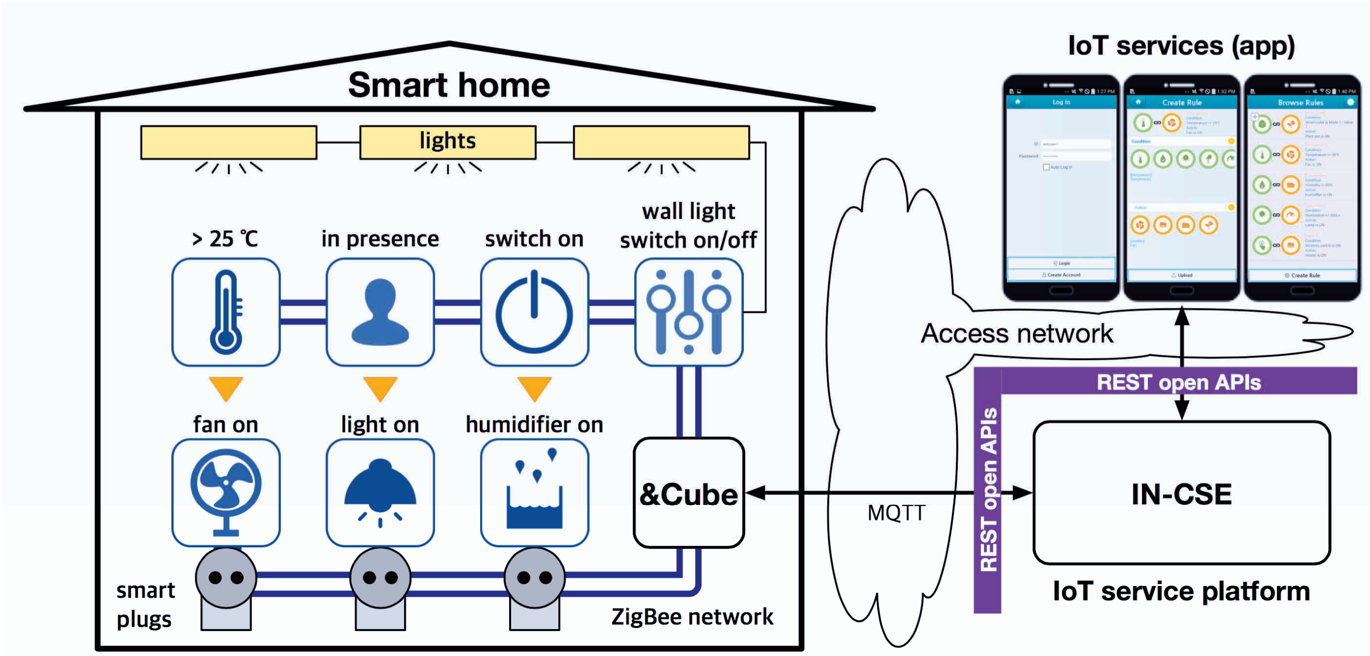

Fig. 7. The overall diagram for a service scenario for smart homes, called TTEO (Things Talk to Each Other), consisting of the IoT service platform, the device software platform for IoT consumer electronics (&Cube), the prototype IoT devices, and a smartphone application created with REST open APIs provided by the IN-CSE, for setting rules that configures autonomous operation between IoT devices.

Fig. 7 illustrates the overall diagram for the example of IoT services in smart homes, called TTEO service (Things Talk to Each Other). It is composed of all the parts of IoT systems explained above, including the IoT service platform (i.e., IN-CSE), the software platform for IoT consumer electronics (&Cube), and IoT prototype devices described in the previous section. In addition, a smartphone application was developed, with which users can set simple rules between IoT consumer electronics.

The IN-CSE is implemented using the 'Mobius', which is an open-source IoT service platform provided by the OCEAN. However, it is also possible to implement the IN-CSE using other open sources or developer's own codes complying with the oneM2M specifications [35]. The In-CSE for the TTEO service is running on a Linux server.

As shown in Fig. 7, three `if-then` rules for autonomous operation between the prototype IoT electronics are set: (R1) `if` temperature > 25 °C `then` turn a fan on, (R2) `if` person presence `then` turn a lamp on, (R3) `if` a switch on `then` turn on a humidifier. In this scenario, each of all three home appliances is respectively plugged in the corresponding smart plug. Note that it is possible to add remote control function to the existing lighting system by only replacing its wall light switch with the IoT wall switch, without changing the whole system, as shown in Fig. 7. All the prototype devices in the smart home including sensors, switches, and smart plugs are wirelessly connected with the &Cube-based IoT home gateway (i.e., IoTGW), and subsequently connected to the IN-CSE via oneM2M standards interfaces. Such standardized Internet connectivity enables the IN-CSE to interact with the prototype IoT devices at home, and in turn to provide REST open APIs with which the IoT devices at home can be monitored and controlled. Finally, based on the REST open APIs offered by the IN-CSE, an IoT service for smart homes (e.g., TTEO) can be developed.

In summary, for the IoT service development with the given prototype devices and the &Cube along with proper TASs, all IoT service developers have to do is to consider a new idea, design service logics, and implement a service application with the IN-CSE APIs. Of course, if manufacturers want to create a new IoT consumer electronics (e.g., IoT washing machine), what they have to do will be to first incorporate a hardware control system running the &Cube into the existing washing machine (or new one), next develop a set of TASs appropriate for the sensors or actuators embedded, and finally develop corresponding device apps to the washing machine functions (e.g., soak, scrub, wash, etc). Based on its REST APIs it is possible to develop a smartphone app for users to connect and control the washing machine at anytime and from anyplace.

VII. Discussion and Remaining Challenges

The &Cube is designed to serve as an IoT gateway to things having no Internet connectivity, and can naturally be ported on any other hardware devices supporting the JRE. However, the Java dependency may limit its use to hardware systems having enough computing resources to provide the JRE. In this case, there would be several methods to resolve the problem. First, as shown so far, such devices with limited resources can also be connected to IoT service platforms via an &Cube-based IoT gateway together with proper TAS. It explicitly means that they cannot support Internet connectivity without the gateway. Another method is building a proprietary device software platform for a given hardware system using other native programming languages (e.g., C), though such implementation dependency will weaken its portability and reusability across heterogeneous hardware systems. Lastly, it will be possible to build a CoAP-enabled software platform for IoT devices with constrained resources and bandwidth.

Although the &Cube has been tested with the prototype hardware, IoTGW, it could be run on any other JRE-enabled hardware systems. However, considering the potential of smartphones as mobile gateways for daily objects with limited wireless connectivity, it would be significantly useful to build a new version of the &Cube for smartphones.

Besides oneM2M standards, there are company alliances for building *de facto* standards for IoT systems, including AllSeen Alliance and Open Interconnect Consortium (OIC). oneM2M working groups have a plan for interworking across oneM2M and other de facto standards-based platforms. As a conclusion, the future research includes building new &Cube to reflect the future released oneM2M specifications and interworking aspects across heterogeneous IoT platforms.

VIII. Conclusion

As highlighted from the Consumer Electronics Show (CES) this year, the Internet of Things (IoT) is the obvious trend, and has crept into the dwellings, particularly, consumer electronics. But, proprietary systems for existing IoT consumer electronics will make it difficult to extend systems to support new devices, share status and context in the surroundings, and work together cooperatively. Such highly-fragmented IoT landscape will also force developers and end-users to have to choose proprietary companies and relevant products.

This paper demonstrates an oneM2M standards-compliant device software platform for IoT consumer electronics. An extensive literature survey has been performed to analyze the requirements for device software platforms for IoT consumer electronics. Based on the requirement analysis, &Cube, a device software platform has been developed. The &Cube provides a standard resource architecture and REST APIs, and thus naturally interacts with the oneM2M service platform, IN-CSE. Finally, an IoT service scenario in smart homes has been illustrated consisting of prototype IoT devices. With the further development of the &Cube for interworking with other de facto IoT standards like AllSeen Alliance and OIC, &Cube will be able to be a cornerstone of the future IoT consumer electronics world.

Acknowledgment

Thanks to Min-Hwan Song for assistance with prototype IoT device development, Sung-Chan Choi for assistance with oneM2M standards analysis, and Chen Nan for assistance with thing adaptation softwares (TAS) development.

References

[1] K. Ashton, “That ‘internet of things’ thing,” *RFiD Journal*, vol. 22, no. 7, pp. 97-114, Jul. 2009.

[2] P. A. Bernstein, “Middleware: A model for distributed system services,” *Commun. ACM*, vol. 39, no. 2, pp. 86-98, Feb. 1996.

[3] J. Choi, D. Shin, and D. Shin, “Research and implementation of the context-aware middleware for controlling home appliances,” *IEEE Trans. Consumer Electron.*, vol. 51, no. 1, pp. 301-306, Feb. 2005.

[4] K.-D. Moon, Y.-H. Lee, C.-E. Lee, and Y.-S. Son, “Design of a universal middleware bridge for device interoperability in heterogeneous home network middleware,” *IEEE Trans. Consumer Electron.*, vol. 51, no. 1, pp. 314-318, Feb. 2005.

[5] J. Yoon and S. Park, “Interoperability enhancement for cable host navigation,” *IEEE Trans. Consumer Electron.*, vol. 56, no. 2, pp. 554-559, May 2010.

[6] T. Perumal, A. R. Ramli, and C. Y. Leong, “Interoperability framework for smart home systems,” *IEEE Trans. Consumer Electron.*, vol. 57, no. 4, pp. 1607-1611, Nov. 2011.

[7] Y.-S. Bae, B.-J. Oh, K.-D. Moon, Y.-G. Ha, and S.-W. Kim, “Design and implementation of an adaptive middleware based on the universal middleware bridge for heterogeneous home networks,” *IEEE Trans. Consumer Electron.*, vol. 56, no. 2, pp. 620-626, May 2010.

[8] Y. Uhm, I. Hong, G. Kim, B. Lee, and S. Park, “Design and implementation of power-aware LED light enabler with location-aware adaptive middleware and context-aware user pattern,” *IEEE Trans. Consumer Electron.*, vol. 56, no. 1, pp. 231-239, Feb. 2010.

[9] Y. Uhm, M. Lee, J. Byun, Y. Kim, and S. Park, “Development of portable intelligent gateway system for ubiquitous entertainment and location-aware push services,” *IEEE Trans. Consumer Electron.*, vol. 56, no. 1, pp. 70-78, Feb. 2010.

[10] Z. Hwang, Y. Uhm, Y. Kim, G. Kim, and S. Park, “Development of LED smart switch with light-weight middleware for location-aware services in smart home,” *IEEE Trans. Consumer Electron.*, vol. 56, no. 3, pp. 1395-1402, Aug. 2010.

[11] P. A. Cabarcos, R. S. Guerrero, F. A. Mendoza, D. Díaz-Sánchez, and A. M. López, “FamTV: An architecture for presence-aware personalized television,” *IEEE Trans. Consumer Electron.*, vol. 57, no. 1, pp. 6-13, Feb. 2011.

[12] F. Battaglia and G. Iannizzotto, “An open architecture to develop a handheld device for helping visually impaired people,” *IEEE Trans. Consumer Electron.*, vol. 58, no. 3, pp. 1086-1093, Aug. 2012.

[13] H. Y. Tung, K. F. Tsang, H. C. Tung, K. T. Chui, and H. R. Chi, “The design of dual radio ZigBee homecare gateway for remote patient monitoring,” *IEEE Trans. Consumer Electron.*, vol. 59, no. 4, pp. 756-764, Nov. 2013.

[14] C. Ma, P. Zhao, and S.-m. Hu, “SerialDriver: Improving the reliability of device drivers through serialization,” *IEEE Trans. Consumer Electron.*, vol. 58, no. 3, pp. 1070-1076, Aug. 2012.

[15] A. Sleman and R. Moeller, “SOA distributed operating system for managing embedded devices in home and building automation,” *IEEE Trans. Consumer Electron.*, vol. 57, no. 2, pp. 945-952, May 2011.

[16] R. Bosman, J. Lukkien, and R. Verhoeven, “Gateway architectures for service oriented application-level gateways,” *IEEE Trans. Consumer Electron.*, vol. 57, no. 2, pp. 453-461, May 2011.

[17] H. Kasai, “Embedded middleware and software development kit for area-based distributed mobile cache system,” *IEEE Trans. Consumer Electron.*, vol. 59, no. 1, pp. 281-289, Feb. 2013.

[18] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Comput. Netw.*, vol. 54, no. 15, pp. 2787-2805, Oct. 2010.

[19] D. Bandyopadhyay and J. Sen, “Internet of Things: Applications and challenges in technology and standardization,” *Wireless Pers. Commun.*, vol. 58, no. 1, pp. 49-69, 2011.

[20] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, “Internet of Things: Vision, applications and research challenges,” *Ad Hoc Netw.*, vol. 10, no. 7, pp. 1497-1516, Sep. 2012.

[21] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Gener. Comp. Sy.*, vol. 29, no. 7, pp. 1645-1660, Sep. 2013.

[22] J. Kim, J. Lee, J. Kim, and J. Yun, “M2M service platforms: survey, issues, and enabling technologies,” *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 61-76, 2014.
[23] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Context aware computing for the Internet of Things: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 414-454, 2014.
[24] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, and V. Terziyan, “Smart semantic middleware for the Internet of Things,” *in Proc. 5th International Conference on Informatics in Control, Automation and Robotics*, Portugal, pp. 11-15, May 2008.
[25] Z. Song, A. A. Cárdenas, and R. Masuoka, “Semantic middleware for the Internet of Things,” *in Proc. Internet of Things Conference*, Tokyo, Japan, pp. 1-10, Nov. 2010.
[26] M. Kranz, P. Holleis, and A. Schmidt, “Embedded interaction: Interacting with the Internet of Things,” *IEEE Internet Comput.*, vol. 14, no. 2, pp. 46-53, Mar. 2010.
[27] L. Roalter, M. Kranz, and A. Möller, “A middleware for intelligent environments and the Internet of Things,” *in Proc. 7th International Conference on Ubiquitous Intelligence and Computing*, pp. 267-281, Xi’an, China, Oct. 2010.
[28] S. Hasan and E. Curry, “Thingsonomy: Tackling variety in Internet of Things events,” *IEEE Internet Comput.*, vol. 19, no. 2, pp. 10-18, Mar. 2015.
[29] M. Mrissa, L. Médini, J.-P. Jamont, N. L. Sommer, and J. Laplace, “An avatar architecture for the Web of Things,” *IEEE Internet Comput.*, vol. 19, no. 2, pp. 30-38, Mar. 2015.
[30] J. Swetina, G. Lu, P. Jacobs, F. Ennesser, and J. Song, “Toward a standardized common M2M service layer platform: Introduction to oneM2M,” *IEEE Wireless Commun. Mag.*, vol. 21, no. 3, pp. 20-26, Jun. 2014.
[31] M. Vidakovic, T. Maruna, N. Teslic, and V. Mihic, “A Java API interface for the integration of DTV services in embedded multimedia devices,” *IEEE Trans. Consumer Electron.*, vol. 58, no. 3, pp. 1063-1069, Aug. 2012.
[32] V. F. de Lucena Jr., N. S. Viana, O. B. Maia, J. E. C. Filho, and W. S. da Silva Jr., “Designing an extension API for bridging Ginga iDTV applications and home services,” *IEEE Trans. Consumer Electron.*, vol. 58, no. 3, pp. 1077-1085, Aug. 2012.
[33] N. Lukic, N. Teslic, T. Maruna, and V. Mihic, “A Java API interface for the search of DTV services in embedded multimedia devices,” *IEEE Trans. Consumer Electron.*, vol. 59, no. 4, pp. 875-882, Nov. 2013.
[34] J. Yun, S.-S. Lee, I.-Y. Ahn, M.-H. Song, and M.-W. Ryu, “Monitoring and control of energy consumption using smart sockets and smartphones,” *in Proc. International Conference on Green and Smart Technologies*, pp. 284-290, 2012.
[35] M. Ryu, J. Kim, and J. Yun, “Integrated semantics service platform for the Internet of Things: A case study of a smart office,” *Sensors*, vol. 15, no. 1, pp. 2137-2160, 2015.

BIOGRAPHIES

Jaeseok Yun (M’06) received his M.Sc. and Ph.D. degrees in mechatronics from the Gwangju Institute of Science and Technology (GIST), S. Korea, in 1999 and 2006, respectively. He was a post-doctoral research scientist in the Ubiquitous Computing Research Group, School of Interactive Computing, Georgia Institute of Technology, from 2006 to 2009. He is currently a Senior Researcher at the Embedded Software Convergence Research Center, Korea Electronics Technology Institute (KETI), S. Korea. His research interests include ubiquitous computing, Internet of Things, human computer interaction, and context-aware computing.

Il-Yeop Ahn received his M.Sc. degree in electronics and information engineering in 2001, and is now a PhD candidate in the electronics control engineering from the Hankuk University of Foreign Studies, S. Korea. He is currently a Managerial Researcher at the Embedded Software Convergence Research Center, Korea Electronics Technology Institute (KETI), S. Korea. His research interests include Internet of Things, wireless sensor network, signal processing, and computer vision.

Nak-Myung Sung received his B.Sc. and M.Sc. degrees in electronic and information engineering from the Hankuk University of Foreign Studies, S. Korea, in 2010 and 2015, respectively. He is currently a researcher at the Embedded Software Convergence Research Center, Korea Electronics Technology Institute (KETI), S. Korea. His research interests include Internet of Things, embedded system software, future internet architecture, and standard for M2M/IoT device systems.

Jaeho Kim received his B.Sc. and M.Sc. degrees in computer science and engineering from the Hankuk University of Foreign Studies, S. Korea, in 1996 and 2000, respectively. He is now a Ph.D. candidate in the electrical & electronic engineering from the Yonsei University, S. Korea. He is a Managerial Researcher at the Embedded Software Convergence Research Center, Korea Electronics Technology Institute (KETI), S. Korea. He is leading the research team for IoT and M2M platforms in KETI. He is also serving as the IoT Convergence Project Group chair of TTA (Telecommunications Technology Association) and Device Working Group chair of Korea IoT Association. His research interests include Internet of Things, medium access protocols, and wireless sensor networks.