

RESEARCH

Open Access

Machine-to-machine platform architecture for horizontal service integration

Eui-Jik Kim¹ and Sungkwan Youm^{2*}

Abstract

Nowadays, machine-to-machine (M2M) communication is being considered as the most promising solution for revolutionizing the future intelligent pervasive applications. This article presents the design of an M2M service platform architecture and prototype implementation of a home automation system based on the designed M2M platform. The proposed service platform consists of three subsystems: M2M service enablement, M2M device management, and M2M communication management. These subsystems include the essential function blocks and interact with each other to horizontally accommodate various M2M applications. The prototype implementation of the full "smart home automation" system, including the M2M service platform, M2M area network, and user/administrator domains, shows the feasibility of the proposed M2M platform architecture.

Keywords: Horizontal integration, Machine-to-machine, Prototype implementation, Service platform

1. Introduction

Machine-to-machine (M2M) communication has emerged as a cutting edge technology for next-generation communications and is considered as the most promising solution for revolutionizing the future intelligent pervasive applications [1-6]. Wireless M2M communication is a form of data transfer that let machines communicate directly with one another with little or no human intervention. It can be used in various applications such as home automation, telemetry, energy management, and transportation, with the objectives of improving the efficiency and reducing the cost. Moreover, it is expected that M2M will play an important role as the infrastructure for a ubiquitous society in the near future.

However, from the perspective of M2M business, service enabler companies, such as mobile network operators, which are able to own their service infrastructure to provide M2M services, face many market and technical challenges, which can be described as follows: due to the long-tail market feature of M2M, most current business cases (i.e., the conventional model) use the vertical integration model, in which the service provider should configure the service infrastructure such as the application server on

a case-by-case basis per individual M2M application [7]. Kouji et al. [8] reported that the horizontal integration model is more desirable than the vertical model to stimulate the market for M2M services and that, in the horizontal model, a common M2M service platform is needed to continue collecting information on various devices and to be able to manage and control them. In addition, the technical challenge is tightly related to the functions or capabilities of this M2M service platform, which includes service enablement, device management, connection management, and M2M application deployment.

Recently, a lot of effort has been put into the standardization of such platforms [9,10]. For example, the European Telecommunications Standards Institute (ETSI) has launched the M2M Technical Committee with the purpose of developing an end-to-end architecture for M2M communications [11]. To support a wide range of M2M applications, the ETSI TC M2M is defining a set of standardized service capabilities that can be implemented in an M2M platform located in the network domain, in an M2M device (terminal) or M2M gateway. Also, to promote the wireless interconnectivity of the different M2M components, mobile operators around the world have been active in constructing platforms to integrate M2M services by using infrastructure networks and launching M2M projects (e.g., GSM Association's Embedded Mobile Initiative) [12]. Nevertheless, there is no consensus on

* Correspondence: sk.youm@samsung.com

²Telecommunication Systems Division, Samsung Electronics Co., Ltd., Maetan 3-dong, Yeongtong-gu, Suwon, Gyeonggi-do 443-742, South Korea
Full list of author information is available at the end of the article

the general network and platform architecture to use for M2M communications, and the related challenging issues are still open.

In this article, we first illustrate the network architecture, which is decomposed into three complementary domains: the M2M device/gateway, M2M service platform, and user/administrator. Then, we present a high-level functional architecture for the proposed end-to-end M2M system, while focusing on the detailed functionalities within the M2M service platform. The proposed service platform architecture consists of three subsystems: M2M service enablement (MSE), M2M device management (MDM), and M2M communication management (MCM), which collaboratively support the M2M applications deployed by a service provider, third party, or the general user. In this study, we defined only the essential function blocks for an M2M service platform that is able to horizontally accommodate various applications. In other words, the components related to real business or the integration issues with legacy systems of the mobile network operator are not considered here.

In order to confirm the feasibility of the proposed M2M platform architecture, we established a prototype implementation consisting of a “smart home automation” system, which is divided into three domains, namely, the M2M service platform, M2M area network, and user/administrator domains. First, the M2M service platform is developed using the connected objects operating system (COOS) open source middleware [13]. To set up the M2M area network domain, we used the home miniature test bed, and developed two types of user interface, viz. smart phone application and web browser interfaces for the user/administrator domain. This prototype implementation of the full system shows the feasibility of the proposed M2M platform architecture.

In the following, we describe the design and implementation of this study in detail.

2. System architecture

Figure 1 illustrates the high-level M2M system architecture described in this study to which we applied ETSI's network architecture concept [11] with some modifications. In this figure, the M2M system is divided into three main domains: the M2M area network domain (or M2M device domain), M2M service platform domain, and user/administrator domain. The M2M devices can connect to the M2M service platform directly through a wide area network (WAN) connection (e.g., cellular 3G/4G) or an M2M gateway (aggregation point). The M2M gateway collects and processes data from simpler M2M devices and manages their configuration/operation. Also, it ensures that the M2M devices interoperate with and are interconnected with the communication network. The M2M devices and M2M gateway are comprised of an M2M area network, which provides connectivity among them. Typically, the use of an M2M area network is preferred when the cost, power, or location of the M2M devices is a deciding factor. In this case, several wireless personal area network (WPAN) technologies, such as Wi-Fi, ZigBee/IEEE 802.15.4, and Bluetooth, can be adopted, through which these devices can communicate. The network domain includes the core and access networks, and the application domain contains the M2M-specific business-processing engines. All of these are beyond the scope of this study. Thus, the domains of the M2M system architecture are newly defined, since in this study, we focus on the design of the M2M service platform itself and the implementation of the full system prototype to confirm the feasibility of the framework used for the platform.

3. Proposed M2M service platform

3.1. Framework design

As mentioned above, the vertical market feature of M2M business causes the service providers (e.g., mobile network operators) to require a flexible and scalable service

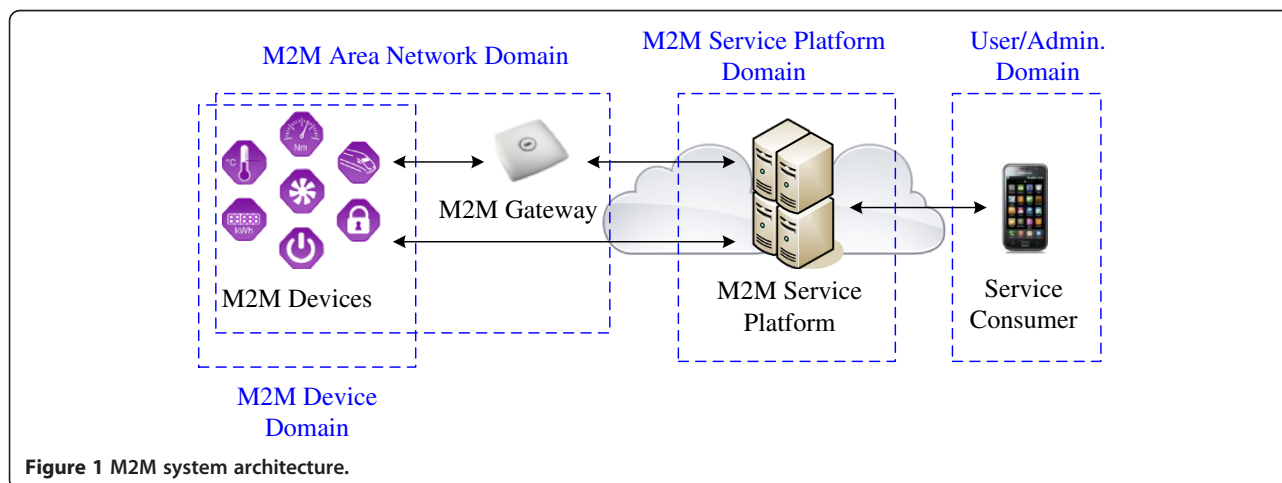


Figure 1 M2M system architecture.

enablement solution for M2M. In other words, the M2M service providers wish to consolidate and extend their vertical business in the M2M market regions by introducing a unified horizontal service platform, which can support multiple verticals from a single-code base. To satisfy this requirement, we propose a generic architecture for the M2M service platform. In this section, we describe the proposed M2M service platform and its components.

Figure 2 shows the high-level functional architecture of the proposed M2M service platform. It consists of three subsystem layers, namely, the MSE, MDM, and MCM layers. When M2M applications (e.g., smart home, smart transportation, smart grid, security, e-healthcare, etc.) are deployed by a service provider, third party, or individual users, the service platform should provide the necessary frameworks as the means to simplify the integration of the

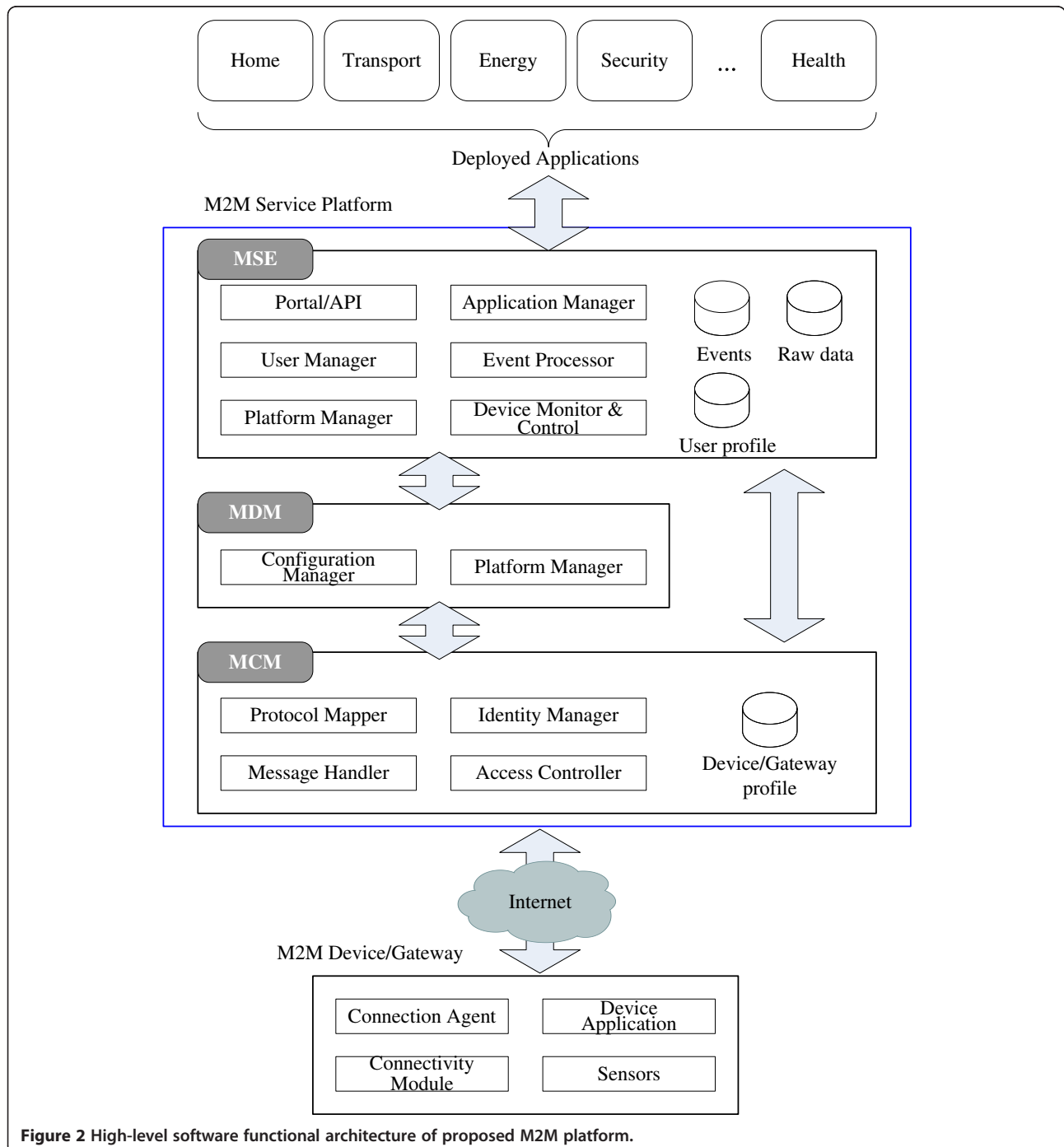


Figure 2 High-level software functional architecture of proposed M2M platform.

newly added M2M devices and applications. The MSE subsystem contains the capabilities to implement a domain and a customer-specific service logic and includes six key function blocks, i.e., Portal/API, Application manager, User manager, Event processor, Platform manager, and Device monitor & control. First of all, MSE provides a unified portal and APIs for the end users and administrator on the northbound. These function blocks simplify the management and provision of the services. The application and user manager manages the creation and changes of application and user-related information, which are stored and retrieved to/from the application/user profile repository. The event processor, which is the primary function block for implementing event-driven applications, includes rules and reporting engines. These engines interact with the events and raw data repositories. The platform manager performs the configuration management, software/hardware management, and fault management for the platform server itself. Note that the MDM and MCM also maintain similar platform manager function blocks to manage their servers. Finally, the device monitor & control supports the management of the end devices from the perspective of the service level, which includes the creation of a new object model, remote device configuration, etc. Note that the MSE subsystem can interact with both MDM and MCM through an internal interface according to the operation scenario.

The MDM subsystem supports the remote management of M2M device/gateway and interacts with the MSE and MCM subsystems. When the configuration policy request arrives from the MSE, the configuration manager block of the MDM sends the configuration command to the MCM. The MCM subsystem maintains the primary functionalities for providing the required connectivity and consists of the following function blocks: Protocol mapper, Identity manager, Message handler, and Access controller. The protocol mapper block plays the role of a mediator for the different protocols or data formats between the M2M application (at the north bound) and M2M device/gateway (at the south bound). For example, at the north bound, it can convert the Java business integration (JBI) protocol to the application protocol or vice versa. At the south bound, this conversion can occur between the JBI and device protocols. The protocol mapper also carries out the data format translation between the device data and service platform data models. The identity manager and access controller blocks manage the IDs of the M2M device/gateway and conduct the authentication and authorization required for their valid usage for the service platform, respectively. The message handler block forwards the message to the appropriate subsystem according to the characteristics of the message. In other words, the sensing data from the end device are directly routed to the MSE subsystem, while the

configuration data, such as device fault alarms, are sent to the MDM.

The M2M device/gateway basically includes the device application, connectivity module, one or more sensors, and connection agent. The connection agent plays the role of the connectivity client for the M2M service platform server and also provides the functionality of protocol adaptation layer required to translate the data format between the different protocols.

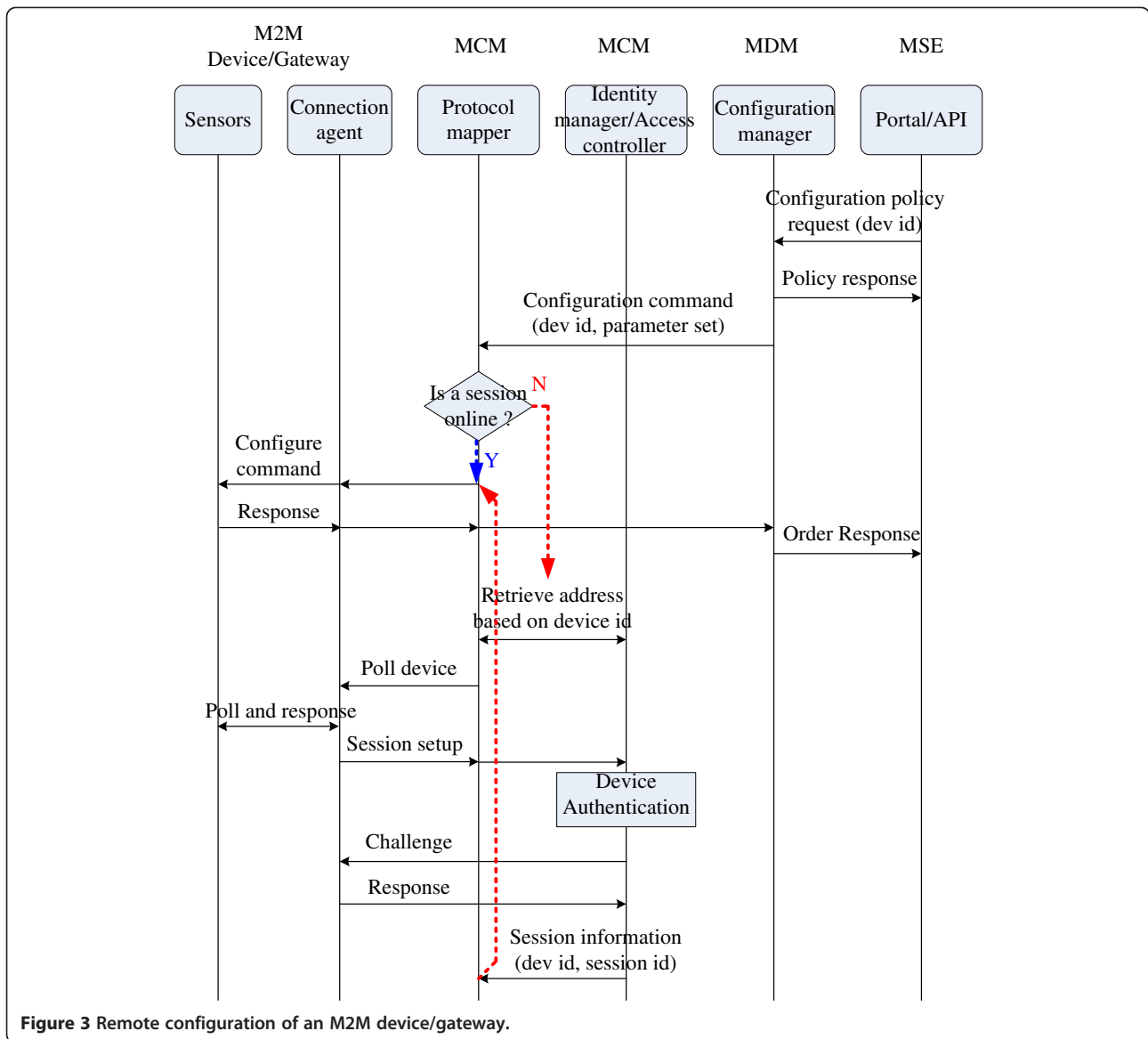
Note that the proposed service platform, including the three subsystems, MSE, MDM, and MCM, closely interacts with the M2M device/gateway, but is independent of the particular choice of underlying networking technology (WPAN or WLAN).

3.2. Operation scenario examples

In this section, we demonstrate the operation flow based on the primary services, namely remote configuration for the M2M devices and sensing data collection (aggregation)/processing, when applying the proposed platform to the M2M communication business.

Figure 3 shows the operation flow of the scenario in the former case. The user can manage and change the configuration of a single object or group of objects via either the exposure API or user interface in the configuration manager of the MDM subsystem. The detailed procedure is as follows:

- The configuration procedure starts when the user asks the system to activate a given policy for the M2M device/gateway via the Portal/API. The request goes to the configuration manager block in the MDM subsystem.
- For each device/gateway, the MDM sends a configuration command, including the device identity and parameter set, to the protocol mapper block of the MCM. The protocol mapper block verifies whether the device is online.
- If it is online, it simply routes the command to the right connection agent, which deploys the configuration changes to the device via the internal API. Each of the entities involved is provided with the outcome.
- In the case where there is no ongoing session, the protocol mapper block retrieves the device's address from the identity manager and access controller of the MCM subsystem. Using its address, it polls the device by sending an appropriate message, which can be an SMS (specific to the device type and vendor).
- The device makes a request to open a session and is challenged to prove its identity. For instance, it can be requested to provide some configuration information as part of its response. Once the authentication process ends, the identity manager



and access controller clears the device to the protocol mapper block, which can then proceed with the configuration process.

The latter scenario is shown in Figure 4 which describes the sequence of actions involved in collecting data from the M2M end devices and applying user-specific business logic to it. As mentioned above, communication between the service platform and the device is made possible by the deployment of a connection agent and protocol mapper blocks on the device and MCM subsystem, respectively. For instance, the connection agent is responsible for interacting with the sensor module over an internal API and periodically polling it to retrieve measurements from the M2M device. In some cases, the sensor module itself is preprogrammed to perform periodical reporting.

- As the first step, the reporting policy is sent via the API to the event processor block in the MSE subsystem.
- This process configures the connection agent to start periodically polling the device.
- Once the MSE subsystem receives the data, the event processor block validates it and then forwards it to the “Events” repository for the user-specific business logic.
- In addition, the event processor stores all of the raw data in a database, “Raw data”, which can be accessed by the application.

4. Prototype implementation

We built a full prototype for the system architecture, in order to validate the framework of the proposed M2M

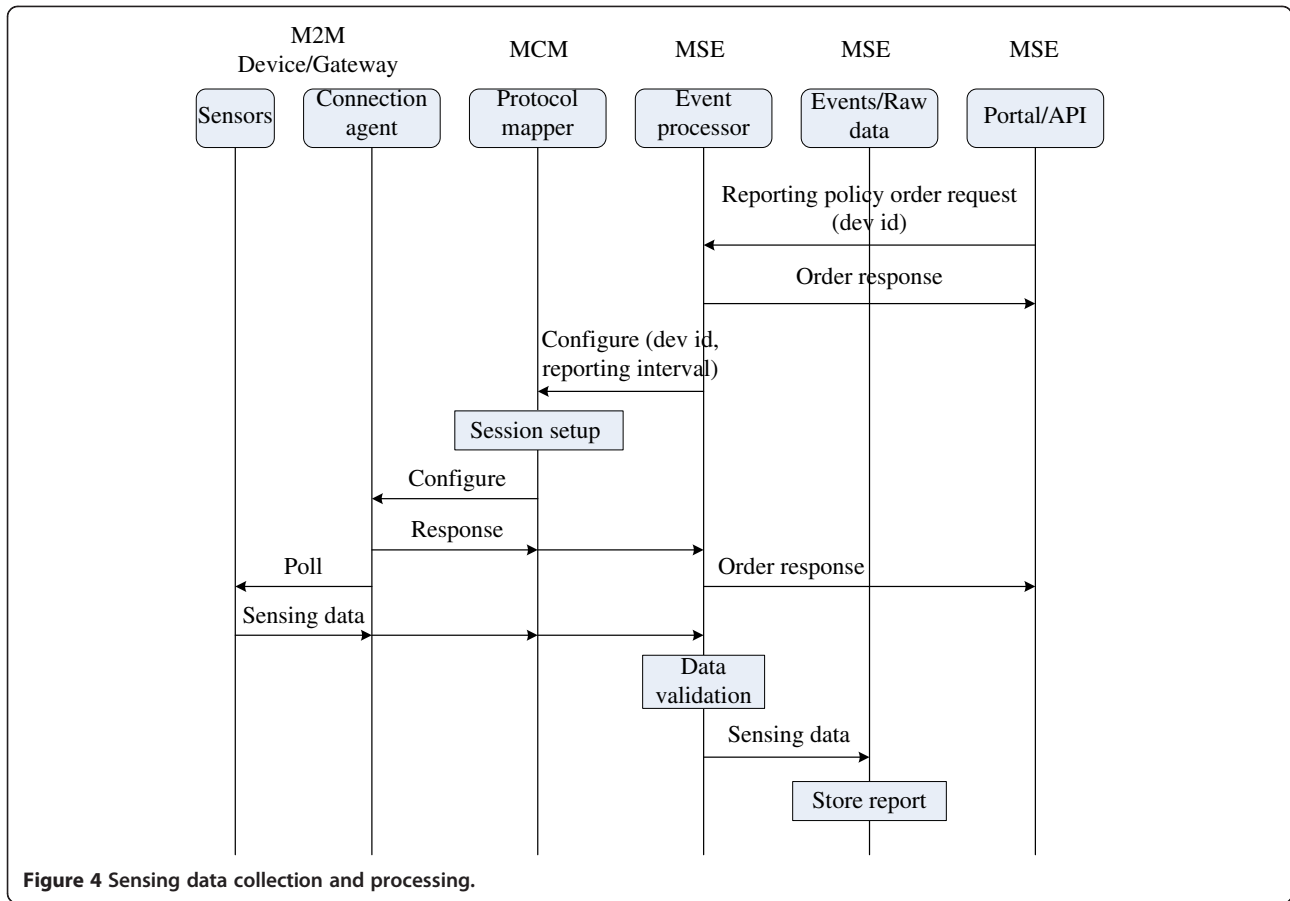


Figure 4 Sensing data collection and processing.

service platform. We configured the domains of the system that are presented in Figure 5, namely the M2M area network (including the M2M gateway and M2M devices beyond it). To set up the M2M service platform, we use three servers; the web application server (WAS), M2M platform server, and data base (DB) server. In our experiment, the WAS is responsible for the integration of the user/administrator terminal and M2M platform

and processes the business logic. The implementation of our M2M platform is based on the COOS, which is a general purpose, modular, pluggable and distributable, open source middleware platform written in Java. The detailed description for the COOS open source platform can be found in [13]. Figure 6 shows the software architecture for the implementation of the M2M service platform. In this figure, the M2M service platform includes the COOS framework that provides the connectivity

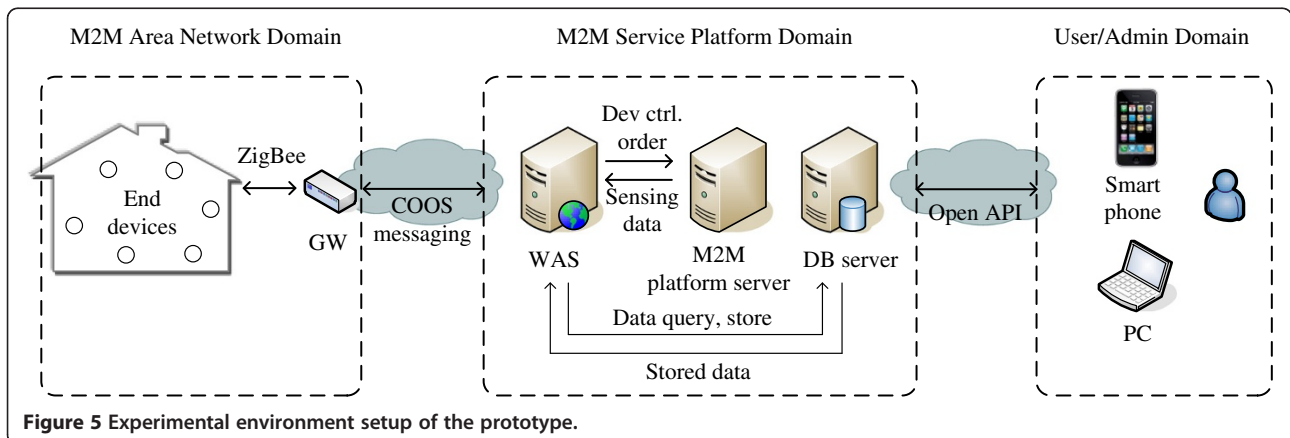


Figure 5 Experimental environment setup of the prototype.

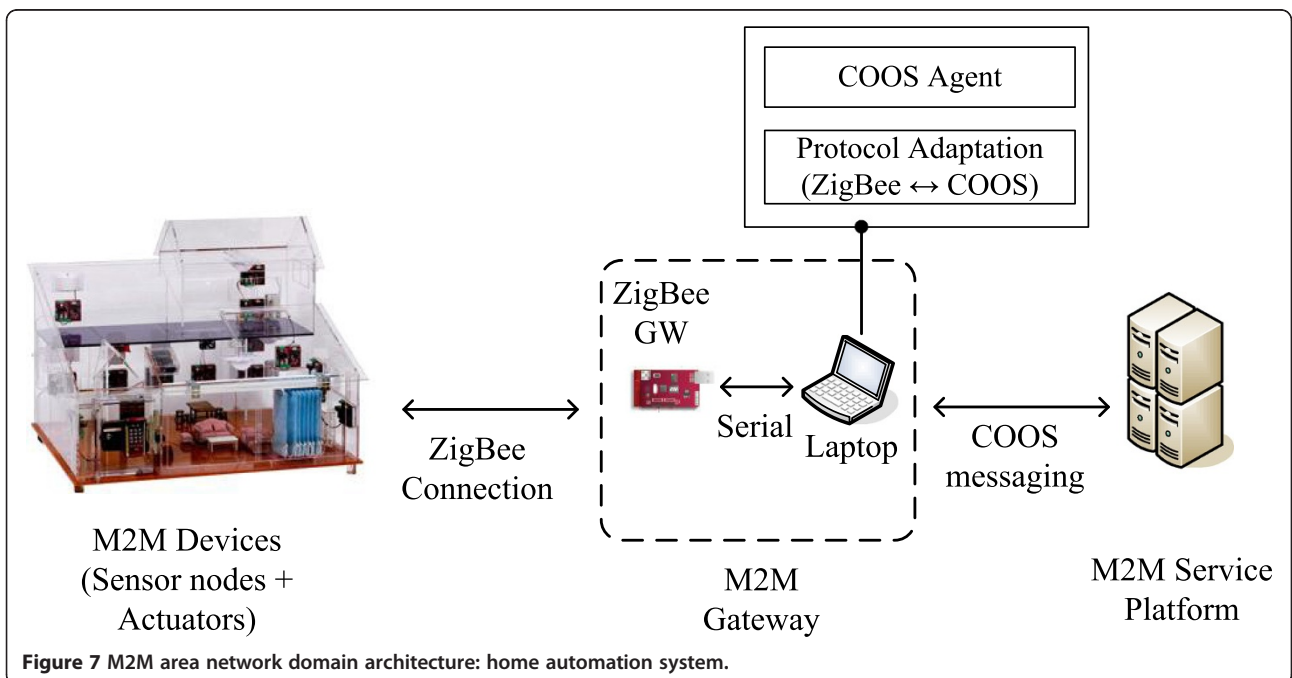
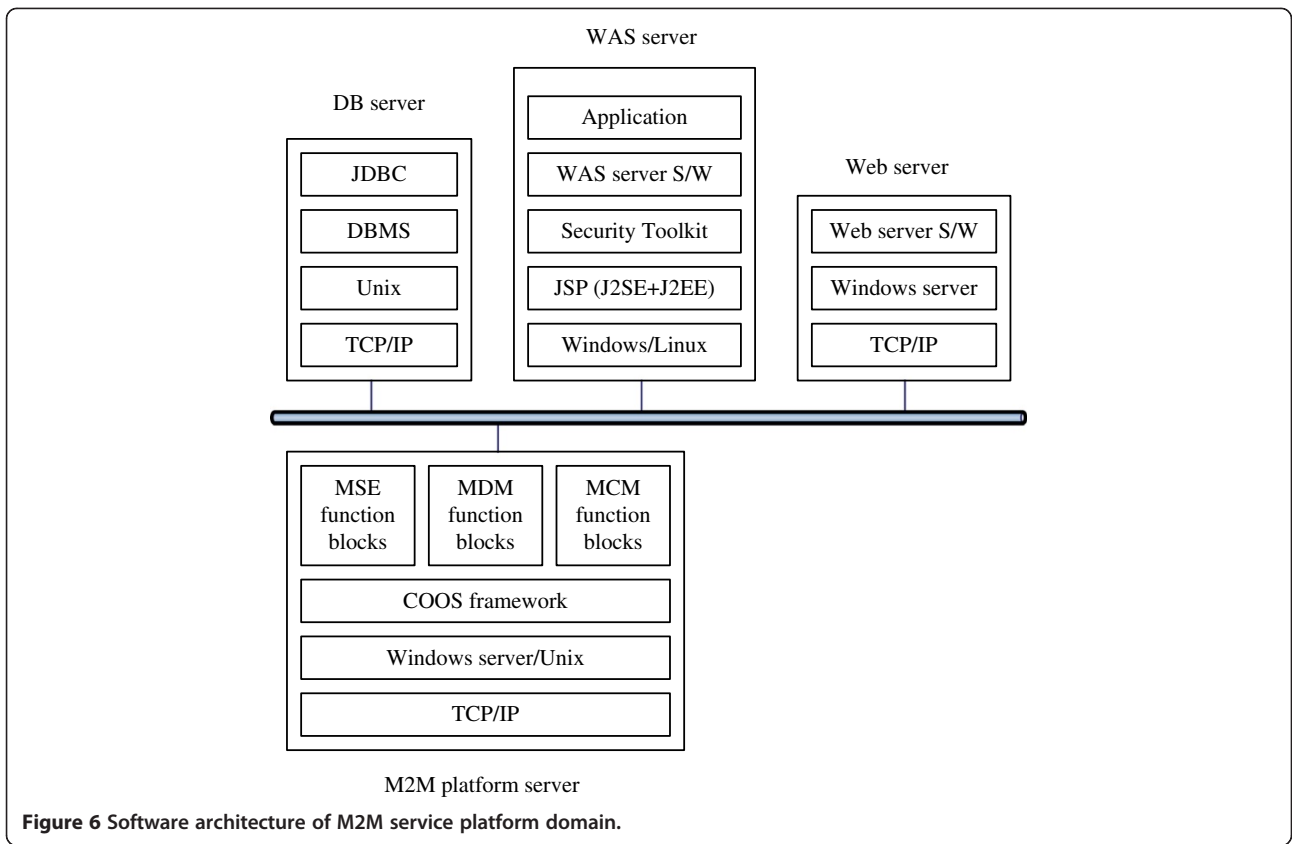




Figure 8 User interface to application: smart phone application interface. (a) Remote control. (b) Sensing information monitoring. (c) Statistical view of accumulated data.

with the end devices via COOS-oriented messaging instead of existing device interfaces such as SMS and HTTP. This partial component of the COOS framework (called the COOS agent) resides locally on a device, in order to provide the connectivity with the M2M platform.

Figure 7 shows the implementation architecture of the M2M area network domain. Note that, in this article, we

consider the “smart home” application as a representative M2M service. To set up the M2M area network for our home automation system, we used the home miniature test bed, which consists of 13 sensor nodes. Each sensor node is equipped with various types of sensors (e.g., temperature, humidity, illumination, and magnetic sensors) and an actuator such as a light, digital door lock, digital gas valve,

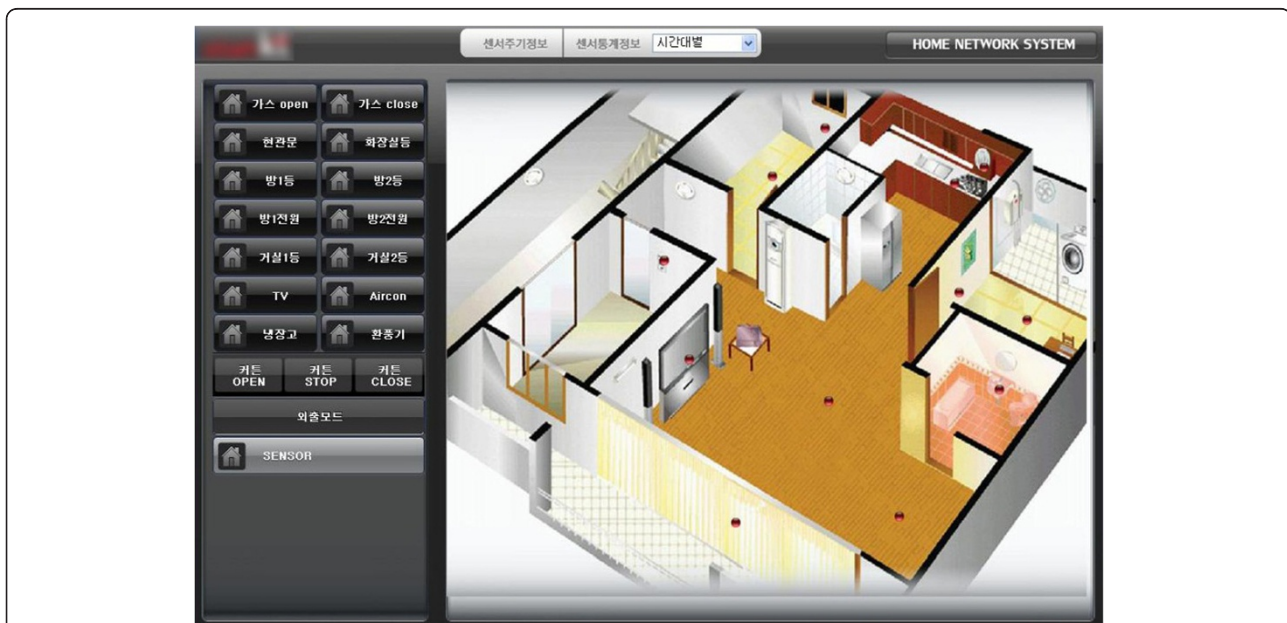


Figure 9 User interface to application: web portal interface (in Korean).

or electronic curtain. The sensor nodes operate by means of a simple sensor network software stack running NesC and TinyOS 2.0, and act as the M2M end devices. These M2M devices exchange data packets with the M2M gateway via a ZigBee connection. The M2M gateway consists of a USB type of sensor node and a laptop PC. On the laptop, COOS agent and protocol adaptation layers are installed to communicate with the M2M platform server and translate the data structure between the ZigBee protocol and COOS message formats, respectively. Note that this configuration enables the M2M service platform to manage appropriately the M2M end devices through the M2M gateway.

The users or administrator can use a smart phone or standard PC as their terminal. We developed both a smart phone application and web page for monitoring and controlling the M2M devices within the miniature house. Figure 8a shows the user interface employed for the remote control of the smart phone application. The user is able to select the menu to monitor and change remotely the status of the actuators (i.e., home appliances) located at home. The detailed procedure is described in the sequence diagram, “remote configuration of M2M device/gateway” in Figure 3. When the configuration command is passed to the M2M gateway, the gateway converts the data type of the command to the ZigBee data type and then transmits it to the corresponding sensor node in order to control the connected actuator.

The M2M end devices (sensor nodes) periodically send their sensing data to the M2M service platform via the M2M gateway based on the reporting policy pre-configured by the user. All of the sensing data are stored in the events and raw data repositories of the DB server according to the procedure described in the sequence in Figure 4. The user can check the current sensing data, such as the temperature and humidity (Figure 8b) and their history (Figure 8c), by means of the user interface on his or her smart phone application. Figure 9 shows the web page version of the user interface for our prototype M2M system, by means of which the user can access the M2M service platform via the Internet and use the same service in the smart phone application.

Our ultimate goal is to design a general purpose M2M service platform rather than a home automation-specific one. In this article, we prototyped the home M2M system to provide insight into the design of the M2M platform architecture. Our future work includes deploying various kinds of M2M applications using our M2M service platform.

5. Conclusion

This article presents a design for an M2M service platform architecture and a prototype implementation of a home automation system based on the designed M2M

platform. In the design of the platform architecture, there exist three collaborating subsystems, namely the MSE, MDM, and MCM subsystems, in which we define the essential function blocks needed to horizontally accommodate various M2M applications. The feasibility of the proposed M2M platform architecture is demonstrated through a full “smart home automation” system, including the M2M service platform, M2M area network, and user/administrator domains.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Advanced Institute of Technology, KT Corporation, Umyeon-dong, Seocho-gu, Seoul 137-792, South Korea. ²Telecommunication Systems Division, Samsung Electronics Co., Ltd., Maetan 3-dong, Yeongtong-gu, Suwon, Gyeonggi-do 443-742, South Korea.

Received: 31 December 2012 Accepted: 20 February 2013
Published: 20 March 2013

References

1. D Niyato, L Xiao, P Wang, Machine-to-machine communications for home energy management system in smart grid. *IEEE Commun. Mag.* **49**(4), 53–59 (2011)
2. R Lu, X Li, X Liang, X Shen, X Lin, GRS: the green, reliability, and security of emerging machine to machine communications. *IEEE Commun. Mag.* **49**(4), 28–35 (2011)
3. Y Zhang, R Yu, S Xie, W Yao, Y Xiao, M Guizani, Home M2M networks: architectures, standards, and QoS improvement. *IEEE Commun. Mag.* **49**(4), 44–52 (2011)
4. ZM Fadlullah, MM Fouda, N Kato, A Takeuchi, N Iwasaki, Y Nozaki, Toward intelligent machine-to-machine communications in smart grid. *IEEE Commun. Mag.* **49**(4), 60–65 (2011)
5. G Wu, S Talwar, K Johnson, N Himayat, KD Johnson, M2M: from mobile to embedded internet. *IEEE Commun. Mag.* **49**(4), 36–43 (2011)
6. SY Lien, KC Chen, Y Lin, Toward ubiquitous massive accesses in 3GPP machine-to-machine communications. *IEEE Commun. Mag.* **49**(4), 66–74 (2011)
7. S Gilani, The promise of M2M: how pervasive connected machines are fueling the next wireless revolution, in *Embedded Systems White Paper*, 2009. <http://www.mentor.com>
8. K Kouji, N Yoshitaro, S Tadashi, M2M service platform to support carrier cloud. *NEC Tech. J.* **5**(2), 116–121 (2010)
9. K Chang, A Soong, M Tseng, Z Xiang, Global wireless machine-to-machine standardization. *IEEE Internet Comput.* **15**(2), 64–69 (2011)
10. Y Katsuhiko, S Hirohisa, Trends in M2M standardization and NEC's activities to promote the standardization of remote management technologies. *NEC Tech. J.* **6**(4), 28–32 (2011)
11. ETSI TS 102 690 v2.0.6, ETSI technical specification, machine-to-machine communications (M2M); functional architecture. (2012). <http://portal.etsi.org>
12. GSM Association (GSMA) Embedded Mobile. <http://www.gsma.com>. Accessed 31 Aug 2012
13. J Audestad, I Gronbak, S Svaet, Connected objects platform specification version 1, R&I Research Report, 1-66. (2009)

doi:10.1186/1687-1499-2013-79

Cite this article as: Kim and Youm: Machine-to-machine platform architecture for horizontal service integration. *EURASIP Journal on Wireless Communications and Networking* 2013 **2013**:79.