ICBMS SM: A Smart Mediator for Mashup Service Development

Doyoung Lee*, Seyeon Jeong*, Taeyeol Jeong*, Jae-Hyoung Yoo†, and James Won-Ki Hong*

*Department of Computer Science and Engineering, POSTECH, Pohang, Korea
Email: {dylee90, jsy0906, dreamerty, jwkhong}@postech.ac.kr
†Ministry of Science, ICT and Future Planning, Korea
Email: jhyoo@iitp.kr

***Abstract*—With the advancement of the Internet technologies, cloud services and various open data, there have been many active attempts to develop mashup services. To develop a mashup service, many data sources and service platforms should be interlinked. Therefore, efficient ways of interconnecting various service platforms and managing data are essential to provide easy service development environment. However, current mashup service development environments do not provide any of them, which forces each developer to fully understand every platforms, interfaces and data format to develop a mahsup service. So, it gives a big burden to mashup service developers and also hinders growth of mashup service industry. In this paper, we propose ICBMS Smart Mediator which connects IoT, cloud, big data, mobile, and security platforms and helps developers to easily utilize them with less overhead and provides easy access to data. We have designed and implemented the proposed ICBMS Smart Mediator and created a new mahsup service with the ICBMS Smart Mediator.**

I. Introduction

The advancement of the Internet has provided an environment where people can easily access and use data. Nowadays, public institutions and companies open their data so the data can be used to produce new value. The data is called open data. For that reason, developers can create new services simply by using the open data. A new service created by using data from various sources and existing services is called a mashup service [1].

Generally, open data can be used through OpenAPI defined by institution or company which provide the data. However, OpenAPIs depend on data providers. Therefore, they do not have unified OpenAPI format, which requires inconvenient procedures to use the OpenAPIs such as getting access token, renewing token, and so on. Moreover, if developers want to use data provided by various institutions, they have to develop different types of interfaces to use OpenAPIs from different data sources. For these reasons, developing a mashup service requires developers to understand a lot of platforms, services and data each time they develop a mashup service. That is, the current mashup service development environment gives too much overhead to developers. In this paper, we propose ICBMS SM (IoT, Cloud, BigData, Mobile, Security Smart Mediator) to remedy the overhead. A main feature of the ICBMS SM is to provide an environment which interlinks various platforms and data sources through an unified OpenAPI. Because mashup services require many interlinked platforms and services, ICBMS SM provides an efficient way of interconnecting multiple platforms and services to satisfy the mashup service requirements. Consequentially, ICBMS SM can encourage developers to make new mashup services by providing an efficient development framework. To achieve the goal, ICBMS SM provides both access functions for various platforms and management methods which can mediate and interlink different types of data. The ICBMS SM's management methods include traffic control, service chain management, and workflow management. Furthermore, the ICBMS SM grants new value to data generated by platforms and opens the data so that they can be used by other mashup services (Fig.1). The contributions of this paper are as follows:

- We propose interfaces to connect various IoT, cloud, big data, mobile and security platforms. Mashup service developers do not need to consider and develop the interfaces required for their service because they are already provided through the SM.
- The proposed SM provides an unified OpenAPI to use the functions of the SM and the interlinked platforms through a web portal. Developers only need to know the OpenAPIs provided by the web instead of studying each platforms OpenAPIs.
- The proposed SM provides additional control functions such as service chaining, workflow engine, and traffic control. Developers can interlink platforms and implement high-quality functions easily.
- We construct a convenient and easy-to-use framework for new mashup service which encourages people to develop a new mashup service.

The rest of this paper is organized as follows: Section II introduces related work. Section III and Section IV present ICBMS architecture and implementation, respectively. Finally, conclusion and future work are discussed in SectionV.

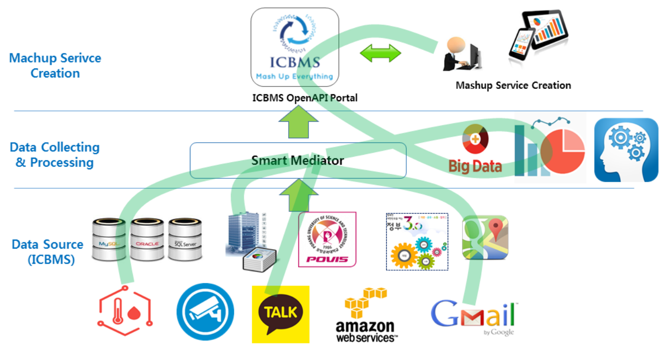

Fig. 1. ICBMS SM Overview

II. Related Work

There are many examples to generate services by combining various platforms. In [2], IoTMaaS, a cloud based IoT mashup service model was proposed to overcome difficulties of connecting and data processing among heterogeneous devices. IoTMaaS satisfies various mashup requirements such as different device vendors and users. In [3], P2P-based architecture for mobile mashup service was introduced. Compared to the Internet-based mashups, mobile mashup services have different features. One of the features is that mobile device connects to the network through mobile telecommunication and mobile device can move without network disconnection. On the other hand, a mobile device has weaknesses including limited hardware environment and network bandwidth. The proposed architecture considered these features and limitations concurrently then applied P2P protocol to resolve and overcome the challenges. In [4], a mashup service recommendation approach was proposed. The approach monitors users interests from their mashup usage history and gives recommendation to the users through the social network by using web APIs and tags. In [5], REST2SOAP was proposed as a framework to integrate RESTful services into BPEL (Business Process execution Language) by transforming a RESTful service into SOAP services semi-automatically. Although RESTful web services are mainstream nowadays, a service-oriented architecture (SOA) [6] is also considered as a successful concept and it is exploited in both industry and academia. However, it is costly or time-consuming to utilize RESTful services in an SOA-based application. For that reason, they tried to solve the problem by using the proposed framework, REST2SOAP. In [7], they proposed data mashup framework to access various heterogeneous IoT data and efficiently generate values from the data. Furthermore, LAMEC (Lightweight Architecture for Mobile web content access over Enterprise Cloud mashup) was proposed in [8]. It is a method which provides mobile web content by utilizing mobile devices and cloud computing resources to minimize size of data delivered when mobile web contents are used. In [9], IoTMAP, IoT mashup application platform for the flexible IoT ecosystem, was proposed. IoTMAP is a platform providing flexible interaction with surrounding smart devices. The platform enables easier developing an IoT mashup application by providing various library and environment for development. EFESTO was proposed in [10] as a platform for the end-user development of interactive workspaces for data exploration. It satisfies an increasing demand by end users. EFESTO makes end users to access, integrate, and use flexibly multiple resources available online.

Those related work [2]-[10], make new mashup services by interlinking IoT, cloud, big data, mobile and security platforms and solve the challenges appeared during the mashup service development process in terms of architectural point of view. However, the number of types of interlinked platforms supported in [2]-[10] is quite small and the purposes of mashup services are very limited. On the other hand, the goal of the ICBMS SM is to provide a general mashup service development environment and to give efficient methods to interlink various platforms.

III. ICBMS SM Architecture

In our proposed ICBMS SM, we have categorized platforms into five types: IoT, Cloud, Big Data, Mobile, and Security. To mediate services and data provided by these platforms, a mediator interlinking the platforms is needed to be located in the middle. Moreover, interfaces have to be developed to make connections between the mediator and each platform. To satisfy the requirements, we implement Smart Mediator (SM) that handles service interlinking and data processing. ICBMS SM, framework for developing mashup service, which is proposed in this paper consists of 1) Message Router, 2) ICBMS adapters, 3) Linked Data, 4) OpenAPI, and 5) developed mashup service (Fig.2).

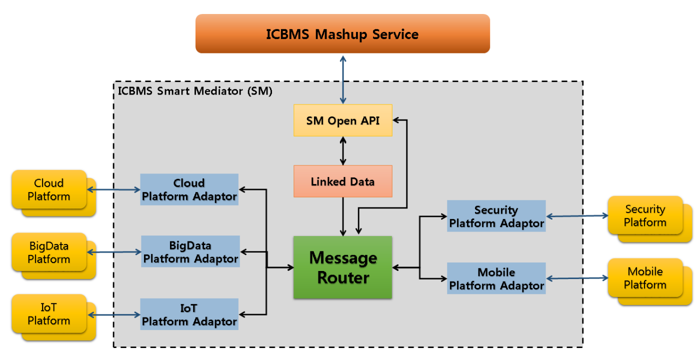

Fig. 2. ICBMS SM Architecture

A. Message Router

One of the main features of SM is to provide a framework for developing mashup services by interlinking various data and services from commercial ICBMS platforms. For that, SM has to have a component to receive and analyze users requests and make sub-requests to deliver them to ICBMS platforms. Moreover, it needs to collect responses from ICBMS platforms and give the result to users. The component is called Message Router. The message router carries out the functions described above in terms of functional side. It also has traffic control function for the framework itself. Furthermore, it provides scaling functions and GUI in terms of management side. The architecture of the proposed message router is described in Fig.3.

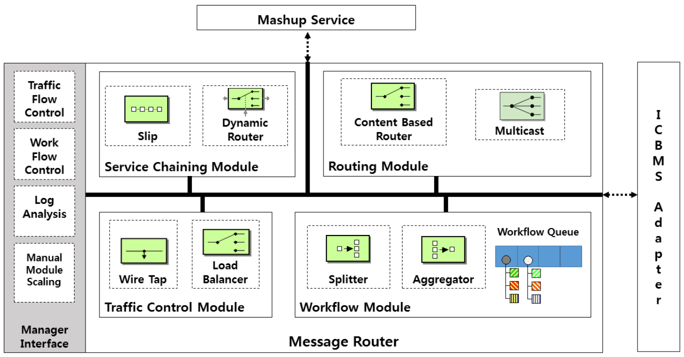

Fig. 3. Message Router Architecture

The proposed message router can be considered as one of EAI (Enterprise Application Integration) frameworks based on EIP (Enterprise Integration Pattern) [11]. EAI framework supports message-based communication between message router and various ICBMS platform adapters to provide linkage. EIP defines message processing patterns and terms which are frequently used in application integration using EAI, so that can be possible to process application integration efficiently by using EIP. For that reason, modules composing the message router are designed based on EIP. They provide message routing function between each internal module as well as each ICBMS adapter.

1) Routing Module: This module receives a mashup service request by an user and analyzes it to define routing path. A destination of the routing path can be another internal module, an ICMBS adapter or a combination of several destinations. A service request by an user is converted into a message containing routing information then the message is delivered to a next destination. To derive routing information from service request message, Content Based Router EIP is applied.

2) Service Chaining Module: Some services are specified on a sequential path which passes through several destinations while the routing module analyzes a mashup service request. For example, if a file stored in a cloud needs to be analyzed by a big data analytics service, it needs to have service chaining in the sequence of service request, ICBMS platform authentication, access to cloud platform and access to big data platform. The service chaining module is designed by using Slip and Dynamic Router EIP to specify a path on the service chaining.

3) Workflow Module: Workflow in the SM is represented as a flow of task in terms of time, that is, scheduling. In case of a mashup service providing real-time analytics of IoT device streaming data, a mashup service requires to deliver parallel queries to different IoT platforms, collect data, and deliver the data to big data platform. After that the Big Data platform analyzes the received data and then returns analysis result to the service. To make the process possible, the workflow module has functions such as splitting and aggregating service request messages, requesting real-time service and batching processing in terms of time. The module is designed based on Splitter and Aggregator EIPs.

4) Traffic Control Module: The message router has to process a huge amount of traffic because it mediates various ICBMS platforms and mashup services. Furthermore, if a specific mashup service uses network resources too much, fairness problem occurs among other mahsup services. For that reason, a tool to monitor, identify, and control the network traffic in the SM is needed. To achieve them, a traffic control module was designed as the tool. An administrator of the message router can monitor traffic volume of each mashup serivce and apply network management policies to the module. Moreover, the module provides functions such as dynamic scaling and traffic load balancing in module level to supply stable environment for developing mashup services. Additionally, Wire Tap and Load Balancer EIPs are used to design the traffic control module.

B. ICBMS Adapter

There are many ways of providing functions for each platform as much as the number of ICBMS platform types. In case of ICBMS platforms based on cloud service, services are provided as IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service). Moreover, data types of each ICBMS platform are variable (e.g., streaming data, Linked Data). ICBMS adapter is operated as an interface between each ICBMS platform and the message router.

1) IoT Adapter: IoT adapter mediates IoT platforms which performs as a hub to connect many IoT devices and the SM. Amazon Web Services (AWS) IoT [12], Azure IoT Suite [13], ThingWorx [14], Mobius [15] and IoTMakers [16] are representative IoT platforms. To use functions or services provided by an IoT platform for developing mashup service, it needs functions such as connection function assuring long-term connection, device profile management, streaming data processing, and access control of user devices.

2) BigData Adapter: Big Data adapter mediates various big data platforms such as Google BigQuery [17] provided as a PaaS, Amazon EMR (Elastic MapReduce) [18] as a IaaS and Apache Spark built on users local environment with the SM. To use functions or services provided by big data platforms for developing mashup service, it needs functions such as analytics function through interlinking various data sources, learning data and constructing data model for analysis (machine learning), and visualization function to represent analyzed data.

3) Cloud Adapter: Cloud adapter mediates services provided by various cloud platforms like Amazon EC2 and Google Compute Engine. The services are given as different types such as IaaS, PaaS and SaaS. To use functions and services provided by the platforms, several functions such as making cloud services, monitoring and managing resources provided by cloud platforms are needed for mashup service development.

4) Mobile Adapter: Mobile adapter accommodates mobile platforms such as Android and iOS. The adapter also accommodates various applications operated on the platforms. For example, Facebook, Instagram, Twitter, and Line. The mobile adapter delivers services or information provided by other ICBMS platforms to mobile platform users. Moreover, the adapter is used for a mashup service when data collected from end users is needed. For that, it needs to provide functions such as an unified authentication method for various mobile platforms and communication functions including multicasting, broadcasting, and periodical polling or non-periodical push.

5) Security Adapter: ICBM platforms themselves can be exposed to security threats and attacks while the platforms are interlinked. For that reason, SM requires an unified security system. The requirement can be satisfied by a security adapter. Various security platforms such as firewall and IDS are connected by the adapter so that mashup service can solve security issues by using it. Moreover, the adapter supports encryption and data anonymity for the data from ICBM platforms. It also provides scheduling function to handle security related requests and responses.

C. Linked Data

Linked Data [19] is a method which grants new value to data. It links the data and makes the data public by using hyperlink. The hyperlink is based on semantic web technique which helps machine to understand the meaning of data. Data can be interlinked with other related data and it makes reuse of data easier. The SM converts data collected from platforms to linked data in order to enable a new mashup service to easily utilize the data.

D. OpenAPI

SM provides it's functions through OpenAPIs. The OpenAPIs interact with interlinked platforms and it helps developers to access data and use services of the platforms. The OpenAPIs are provided through an OpenAPI web portal.

IV. Implementation

To validate the proposed SM functions and effectiveness for developing mashup service, we implemented a building management service prototype as an use case. Current commercial building management solutions have high cost to deploy it and they do not interact with each system. For example, air-conditioning and heating system, electric power management system, and access control system are general solutions to manage a building but these are constructed separately. Moreover, deploying conventional building management systems leads additional cost when a buillding does not have sufficient facilities or equipments for the system installation. Also, once those systems are deployed it is hard to dynamically modify or improve the systems to meet new requirements. For that reason, building management as a mashup service is a meaningful example which can overcome the problems and prove effectiveness of the SM.

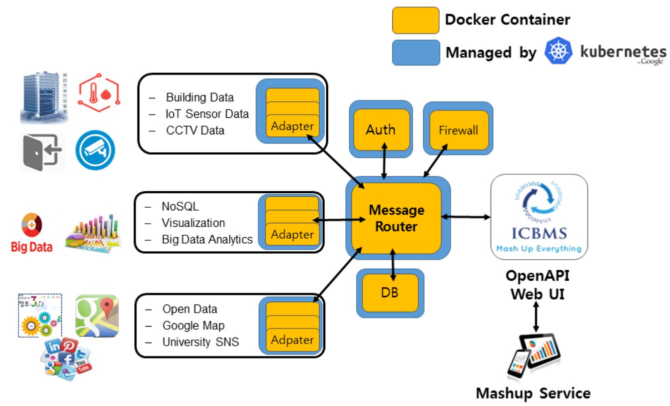

Fig. 4. Smeart Mediator Architecture

A. SM

SM mainly consists of two types of components. One is a message router which interlinks SM components and delivers messages between them. The other is adapters which perform as interfaces for mediating between ICBMS platforms and the SM. Additionally, authentication and firewall functions were included for SM security and stability. Only permitted user can access the SM by authentication function and abnormal traffic can be blocked by the firewall (e.g., blocking specific IP address ranges). Also, relational database is constructed and connected with the message router directly. One of SM functions is to generate new data as linked data format. To store the data, the DB component is added and it generates and stores linked data whenever new data passes through the message router. These components are implemented on each Docker container [20] and they are managed through Kubernetes [21] providing load balancing, scale-in/out and auto-healing functions when SM suffers from overheads or failures (Fig.2).

B. OpenAPI Portal

OpenAPI portal provides web-based interface to help developers to create new mashup services conveniently. The portal supports GUI-based mashup service development for simple mashup services. General people or novice developers can create their new mashup service by drag-and-drop based approach. They just need to select multiple platforms, data sources, job scheduling, visualization and analyzing functions step by step depending on their mashup service through the GUI. These decisions are converted to OpenAPIs which are defined in advance then delivered to the SM. After that, they are interlinked and processed by each ICBMS platform. As a result of the process, web-based mashup service is created (Fig.5). On the other hand, the OpenAPI portal also provides REST API for professional mashup service developers. By using the REST API, developers can utilize the full functions of SM function more dynamically.

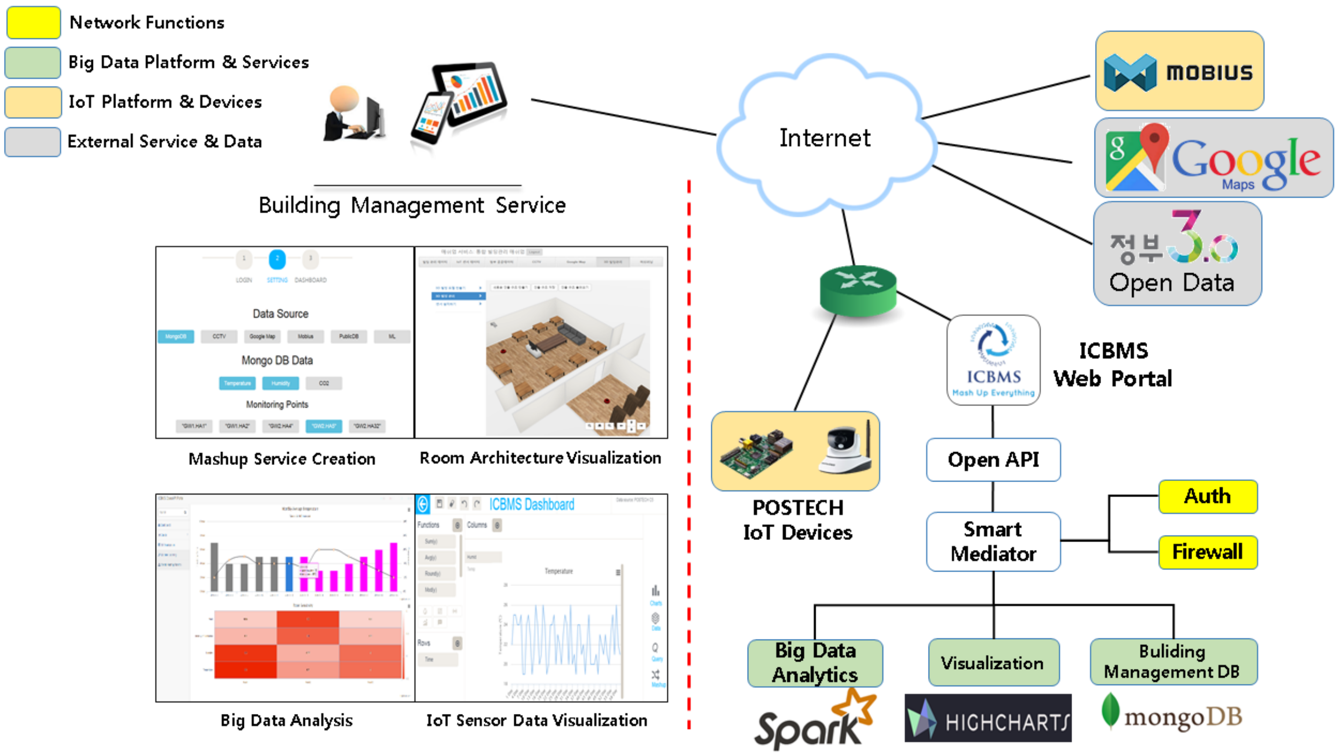

Fig. 5. Building Management Service Implementation

C. Mashup Service

Building management mashup service developed by SM mainly uses services of IoT and big data platforms. The IoT adapter collects building data and the big data adapter analyzes the data collected and stored on a DB then provides meaningful information as a result. To make the meaningful information, several services and data sources were used.

The implemented IoT adapter interlinks IoT platforms such as IoTMakers provided by Korea Telecom, and Mobius provided by SK Telecom. The platforms periodically collect data including temperature, humidity, and light intensity from IoT sensors installed in a building. The IoT adapter connects SM to the IoT platforms and gets sensor data. The collected data is then delivered and stored onto the relational data base, MySQL, which is located in the SM. At the same time, the data is also delivered to the big data adapter to store the data as building information. The data is converted to linked data by D2RQ [22], an open source framework to generate linked data, to be used later. Moreover, the IoT data can be visualized as graphs or charts in time unit according to the requirements of a mashup service.

The big data adapter is mediated with NoSQL platform to use accumulated building data stored in MongoDB [23] and to store some data into the database. Furthermore, Apache Spark [24] was installed in our local environment and a related interface was developed for interlinking. The big data adapter gets data from the MongoDB and automatically converts the data format to make the data format readable by Apache Spark directly. After the data format conversion, data is analyzed by the Apache Spark and the result is returned to the adapter. Big data adapter can deliver the result to other platforms for visualization.

The building management mashup service created by SM provides building information by collecting the data from several data sources. However, SM needs, not just to connect data and service, but to generate new values or insights from them. To achieve this goal, our mashup service provides functions as follows:

- Prediction of future duration of air-conditioning by combining IoT sensor data (temperature), previous duration of air-conditioning, weather forecast, and big data analysis.
- Analysis of each rooms heating efficiency and sensitivity by using building information (temperature, humidity, and light), outdoor information (temperature, humidity, wind velocity), and big data analytics service.
- Air-conditioning and heating based on discomfort index by combining IoT sensor data, air-conditioning and heating system, and discomfort index provided as an open data.
- Monitoring environment by using building information, IoT sensor data, CCTV, Google Maps [25], and 3D visualization function.

The building management mashup service created by the proposed SM, provides not only the core features of conventional building management systems, but also new values or insights for building management. The new values cannot be generated by the existing separated platforms and services. Moreover, the created building management mashup service was implemented in a very low cost compared to the conventional building management system. Also, it can be modified dynamically and extended easily according to new service requirements.

V. Concluding Remarks

We have presented the concept and architecture of ICBMS SM. The ICBMS SM provides development environment for mashup services by efficiently interlinking various IoT, cloud, big data, mobile and security platforms efficiently. The environment encourages developers to develop a mashup service by providing various management functions such as traffic control, service chain management, and workflow management. To validate the proposed SM and its functions, we have implemented the ICBMS SM and developed a building management service as a mashup service use case by only using the proposed SM itself. ICBMS SM enable developers to create new mashup services easily and it can smoothly interlink platforms needed for mashup services. Furthermore, it grants new values to data then store and open the data to developers so that they can utilize them for new mashup services. Finally, SM provides its functions through an OpenAPI portal. Novice developers or even anyone can develop a simple mashup service by drag-and-drop based web portal GUI. Professional developers can fully utilize OpenAPIs by using REST API for high-quality mashup services.

Our future work is to consider many mashup service scenarios to increase the number of platforms interlinked with the ICBMS SM and to build interfaces systematically for better mediation. Additionally, we will work on efficient service chaining and workflow control mechanisms for providing effective service functions. We also plan to devise a method which automatically converts from data generated by different platforms to linked data to utilize the data for new mashup services. Furthermore, OpenAPI format and data type between each adapter and message router will be defined systematically because they are essential for integrating various platforms.

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korean government(MSIP) (R0126-15-1009, Development of Smart Mediator for Mashup Service and Information Sharing among ICBMS Platform).

References

[1] X. Liu, Y. Hui, W. Sun, and H. Liang, "Towards service composition based on mashup," in *Services, 2007 IEEE Congress on*. IEEE, 2007, pp. 332–339.

[2] J. Im, S. Kim, and D. Kim, "Iot mashup as a service: cloud-based mashup service for the internet of things," in *Services Computing (SCC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 462–469.

[3] X. Zhang, M. Song, K. Xu, and J. Song, "Mobile mashup based on p2p: Architecture, design and realization," in *IT in Medicine & Education, 2009. ITIME'09. IEEE International Symposium on*, vol. 1. IEEE, 2009, pp. 416–420.

[4] B. Cao, J. Liu, M. Tang, Z. Zheng, and G. Wang, "Mashup service recommendation based on user interest and social network," in *Web Services (ICWS), 2013 IEEE 20th International Conference on*. IEEE, 2013, pp. 99–106.

[5] Y.-Y. Peng, S.-P. Ma, and J. Lee, "Rest2soap: A framework to integrate soap services and restful services," in *Service-Oriented Computing and Applications (SOCA), 2009 IEEE International Conference on*. IEEE, 2009, pp. 1–4.

[6] E. Newcomer and G. Lomow, *Understanding SOA with web services (independent technology guides)*. Addison-Wesley Professional, 2004.

[7] D. Zhiquan, Y. Nan, C. Bo, and C. Junliang, "Data mashup in the internet of things," in *Computer Science and Network Technology (ICCSNT), 2011 International Conference on*, vol. 2. IEEE, 2011, pp. 948–952.

[8] S. K. Guirguis, A. A. El-Zoghabi, and M. A. Hassan, "Lightweight architecture for mobile web content access over enterprise cloud mashup," in *Computer Applications & Research (WSCAR), 2014 World Symposium on*. IEEE, 2014, pp. 1–8.

[9] S. Heo, S. Woo, J. Im, and D. Kim, "Iot-map: Iot mashup application platform for the flexible iot ecosystem," in *Internet of Things (IOT), 2015 5th International Conference on the*. IEEE, 2015, pp. 163–170.

[10] G. Desolda, C. Ardito, and M. Matera, "Efesto: A platform for the end-user development of interactive workspaces for data exploration," in *Rapid Mashup Development Tools*. Springer, 2016, pp. 63–81.

[11] Enterprise integration patterns. [Online]. Available: http://www.enterpriseintegrationpatterns.com/

[12] Amazon web services (aws) iot. [Online]. Available: https://aws.amazon.com/iot/

[13] Azure iot suite. [Online]. Available: https://www.microsoft.com/en-us/server-cloud/internet-of-things/azure-iot-suite.aspx

[14] Thingworx. [Online]. Available: https://www.thingworx.com/

[15] Mobius. [Online]. Available: http://iotmobius.com/

[16] Iotmakers. [Online]. Available: http://iotmakers.olleh.com/

[17] Google bigquery. [Online]. Available: https://cloud.google.com/bigquery/

[18] Amazon emr (elastic mapreduce). [Online]. Available: https://aws.amazon.com/elasticmapreduce/

[19] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data-the story so far."

[20] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment," *Linux Journal*, vol. 2014, no. 239, p. 2, 2014.

[21] Kubernetes. [Online]. Available: http://kubernetes.io/

[22] C. Bizer and A. Seaborne, "D2rq-treating non-rdf databases as virtual rdf graphs," in *Proceedings of the 3rd international semantic web conference (ISWC2004)*, vol. 2004. Citeseer Hiroshima, 2004.

[23] Mongodb. [Online]. Available: https://www.mongodb.com/

[24] Apache spark. [Online]. Available: http://spark.apache.org/

[25] Google maps. [Online]. Available: https://www.google.co.kr/maps