

IoT Federation Service Provider based on Multi-Platforms using RESTful API for Integrated Device Control in Heterogeneous Actuator Networks

Sehrish Malik¹ and Do-Hyeun Kim^{2*}

^{1,2*}Dept. Of Computer Engineering, Jeju National University,
690-756 Jeju-si, South Korea
¹serrym29@gmail.com, ^{2*}kimdh@jejunu.ac.kr

Abstract

The integration of *IoT*(Internet of Things) and Web is known as *Web of Things*, which is getting a significant attention nowadays. *WoT* intends to enhance the applications of *IoT* with the provision of scalable and flexible solutions for smart things. In this paper, we present *IoT* federation service provider using *RESTful API* for integrated device control in heterogeneous actuator networks. The proposed federation service provider enables multi-platform and middleware for actuator networks to be connected with it independently, making the development of *IoT* environments easier. Additional, we implement *IoT* federation service provider based on web services and present performance analysis of *IoT* federation service using *RESTful API*.

Keywords: *WoT* (Web of Things), *IoT* (Internet of Things), federation service, actuator control

1. Introduction

The amalgamation of *IoT* in the web converts web into *Web of Things* (*WoT*). *Web of Things* aims to provide flexible and scalable ways for building *Internet of Things* using the *Web* as an application layer. A web thing is basically digitalization of any real-world things that can be a door, a room, building, an *Arduino* board *etc.* *Web of Things*' design facilitates maximum interoperability for *Internet of Things*. This web thing is made accessible to the users via web services. Two popular web services are *SOAP* and *RESTful* web services.

Nowadays, *RESTful* web services are preferred over *SOAP* and considered better due to being light-weight. *REST* architectural style is considered one of the best approach for the deployment of the *Web of Things* for actuator and sensor networks. Applications developed using *REST* guidelines are called *Restful* applications. Use of *URI* (*Unique Resource Identifier*) for resources, giving a choice to select most suitable resource representation via *MIME*-types making language aware websites and use of *HTTP* commands as *PUT*, *POST*, *GET* and *DELETE* leads to the creation of *Restful* applications. *Restful* provides more flexibility and less overhead resulting in a better solution for majority of the implementations [8].

A typical *Web of Things* architecture consists of five layers as shown in the Figure 1. The layers are named as network things layer, accessibility layer, findability layer, sharing layer and composition layer. The bottom layer, networked things layer, contains actuator and sensors networks which are counted as the real-world things. The accessibility layer ensures the provision of internet connection via proxies, gateways and protocols offered along with the *RESTful APIs* for exposing services. The findability layer provides *Web*

Received (October 10, 2017), Review Result (December 19, 2017), Accepted (January 16, 2018)

* Corresponding Author

thing model which is discovery of things at Web, use of semantic Web and linked data. The sharing layer deals with authorizations, Web of Things mashups and social Web of Things, basically the idea of pushing the data generated by the things to the web in a secure and efficient manner so that intelligent decision can be made using the generated data. The top layer, composition layer, provides the result in the form of simpler web of things applications and services, by integrating the data generated by the things and services via web tools and web mashups [8-9].

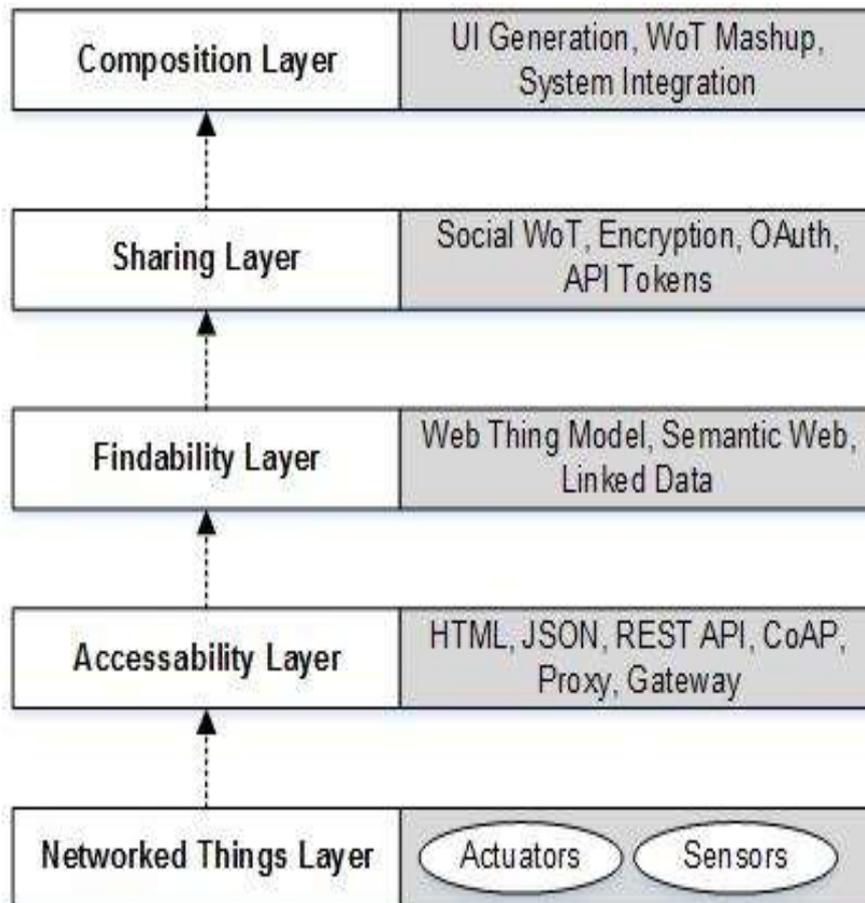


Figure 1. Web of Things Architecture

In this paper we have proposed a federation service provider for controlling things in actuator networks deployed using RESTful API. It aims to provide solution for developing web of things with ease and a generic approach that can be adopted in different scenarios without having too many dependent factors. We have implemented an Actuator Control, which enables user to control actuator in an indoor IoT environment.

The remainder of this paper is divided as follows. Section 2 presents the related literature. Section 3 proposes the design and architecture of the IoT federation service provider. Section 4 presents implementation of the IoT federation service provider along with the performance analysis and section 5 concludes the paper.

2. Related Work

In Web of Things, each resource is accessible through a URL. In design, JSON is considered better as it makes parsing easier as compared to XML but XML provides better integration. As for keeping a uniform interface, HTTP is sufficient providing GET, PUT,

DELETE and POST. Data being sent can viewed in HTTP headers while data in the HPPT bodies [1].

The work presented in [2], proposes to make real-world data and functionality an integral part of the Web to make development of smart things easier. After exploring Web of Things architectures and best-practices based on RESTful, a practical framework is provided enabling users to build their own Web of Things applications. Prototypes for connecting environmental sensor nodes and energy monitoring system are proposed. Main focus is to development of highly interactive and Web-compliant physical mashups.

The work presented in [3] proposes a contribution towards digitalization of things a home, building it to a smart home through the reuse of core principles of Web architecture for building a smart home services. The application framework facilitates multiple occupants in the smart home using web as an application layer and simplifying home automation. The proposed work covers devices' discovery to the services offered and uniform interactions among them and Web.

DiscoWot is a semantic discovery service for Web-enabled resources which depends on the application of multiple Discovery Strategies. Its RESTful implementation enables users to create and update strategies at runtime, aiming the provision of semantical discovery of the functionalities of web-enabled devices by both humans machine users along with device integration and interaction [4]. The work presented in [5], explores potential emerging approaches for executing distributed queries through diverse neuroscience data sources. A variation of semantic web methodologies have been explored to describe, map, and dynamically query created datasets which are health care and life science, aiming to provide support to dynamic query federation. Analysis on the wide range of semantic web technologies is provided, identifying both the strengths and weaknesses, for facilitating a neuroscience query federation scenario.

SUNRISE GATE provides an interface to users for accessing resources offered by the testbeds and run their experiments via SUNRISE federation. The federation considers a variety of marine environments and provides applications services of port monitoring, marine park protection, search and rescue operation and fisheries surveying [6]. Many cloud based federations are also being proposed to simplify the implementations of systems based on multi-platform with data originating from many sources. The work presented in [16], proposes a cloud federation for dealing with problems of processing and storing heterogeneous information. The authors propose a two-layer architecture that provides the client with cloud based Platform as a Service (PaaS). This federation approach covers big data storage approaches with the provision of non-SQL distributed storage system.

The work proposed in [17], discusses the need of provision of services for the real-world via smart objects by federating them into digital world. More attention is needed to the integration of smart objects and ensuring the provision of a unified access to the smart objects data on the Web. The authors then propose a sematic modelling approach and linked data approach for an IoT framework, which enables publication of the data.

3. Architecture of the IoT Federation Service Based on Multi-Platforms using RESTful API

In this section we present the IoT federation service provider design and the overall components. Our main focus is the federation service provider and its role in the Actuator Control. Federation ideally connects to a sensor network platform and an actuator network platform. The proposed architecture enables the joining of multi-platforms with federation and provide services to the client. In this paper, though our main focus is actuator networks but we will also be briefly explaining the actuator network side, for the purpose of better understanding of the whole developed IoT federation service provider.

In the proposed architecture, the federation service provider connects multi-actuator platform, multi-actuator middleware, and service registry in heterogeneous actuator networks. Actuator middleware and actuator network are below actuator platform and connected via actuator platform. Client pass control commands to actuator platform via federation.

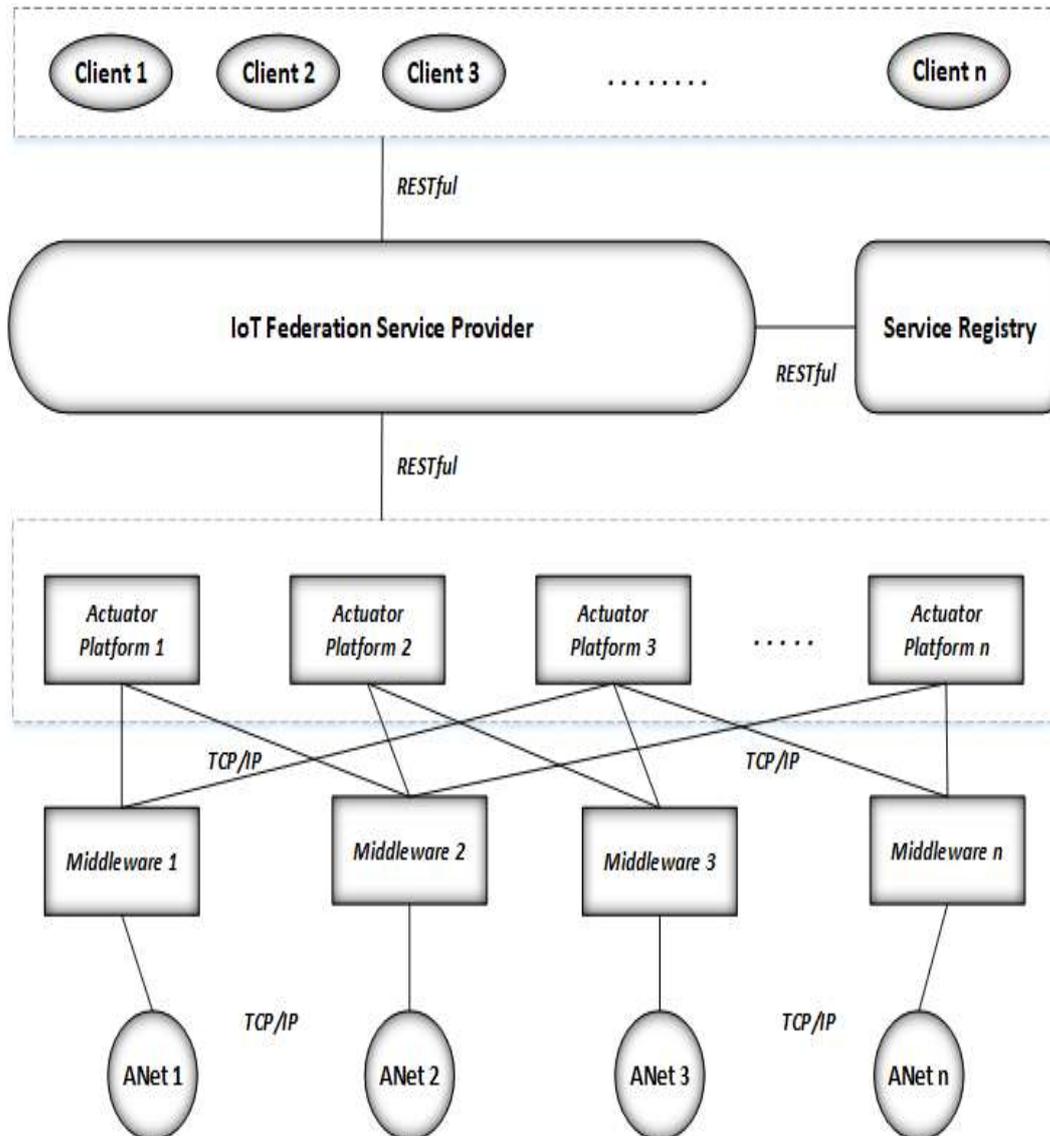


Figure 2. Architecture of IoT Federation Service based on Multi-Platforms using RESTful API for Integrated Actuator Control

3.1. Actuator Network

Actuator network consists of the actuator devices in the indoor environment. The actuator devices can be any electronic devices in the physical world. The system automates the control for actuator devices in order to maintain the indoor environment within set conditions. Our system has four actuator devices as fan, boiler, light and air-conditioner. We have implemented simulators for all four actuator devices and also we have connected light and fan modules with our actuator platform via Edison board.

3.2. Actuator Middleware and Platform

Actuator middleware acts as a gateway between the actuator networks and the actuator platform. Each actuator is connected to the platform by first mapping to a middleware. The middleware enables the actuator network and actuator platform to exchange the status and control commands. Actuator middleware passes on the connected actuators status information to the actuator platform.

Actuator platform provides connectivity to the actuator via actuator middleware. The services provided by the actuator platform are content service and provider service. It keeps a check on the actuator status and sends the control commands to the actuators when needed.

3.3. Service Registry

Service registry shows the available services with service number, service name, service type and service access Uri. Service registry has the information of devices in actuator networks. It also offers to manage and update the services information.

3.4. Federation Service Provider

The configurations of the proposed federation service provider based on the actuator web application platform are shown in Figure 3. Federation service provider is composed of provider service and content service.

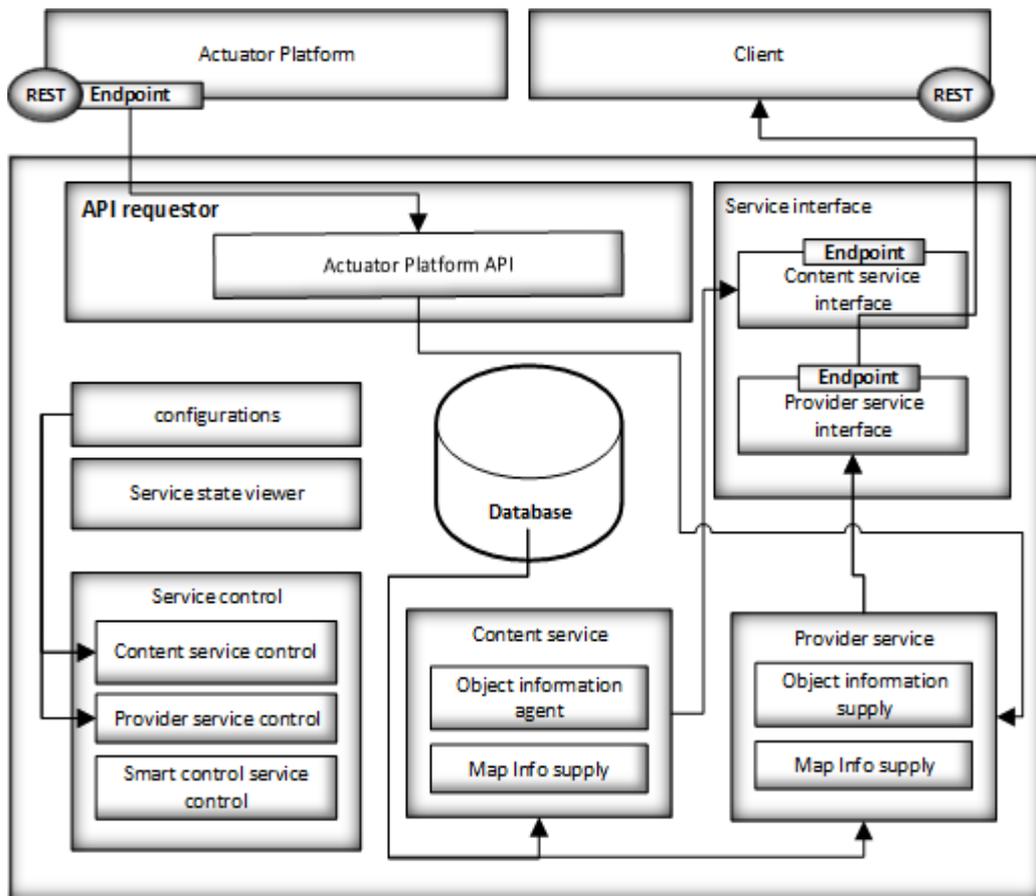


Figure 3. IoT Federation Service Provider Configurations

The provider service is responsible for object, actuator and map information supply. Object information supply provides actuator object information such as location information, type and provider service address to the client and actuator information supply offers actuator detailed information such as actuator name, actuator ID, actuator state and actuator movement attribute. Map information supply offers map's detailed information such as outdoor map, indoor map, building, floor and room information.

Content service consists of object information agent, information supply and map binding agent. Object information agent is like application server composite tool, providing actuator object location information management and map service binding management. Actuator information supply offers actuator detailed information and map information supply offers map's detailed information. Database saves map service binding information and object information.

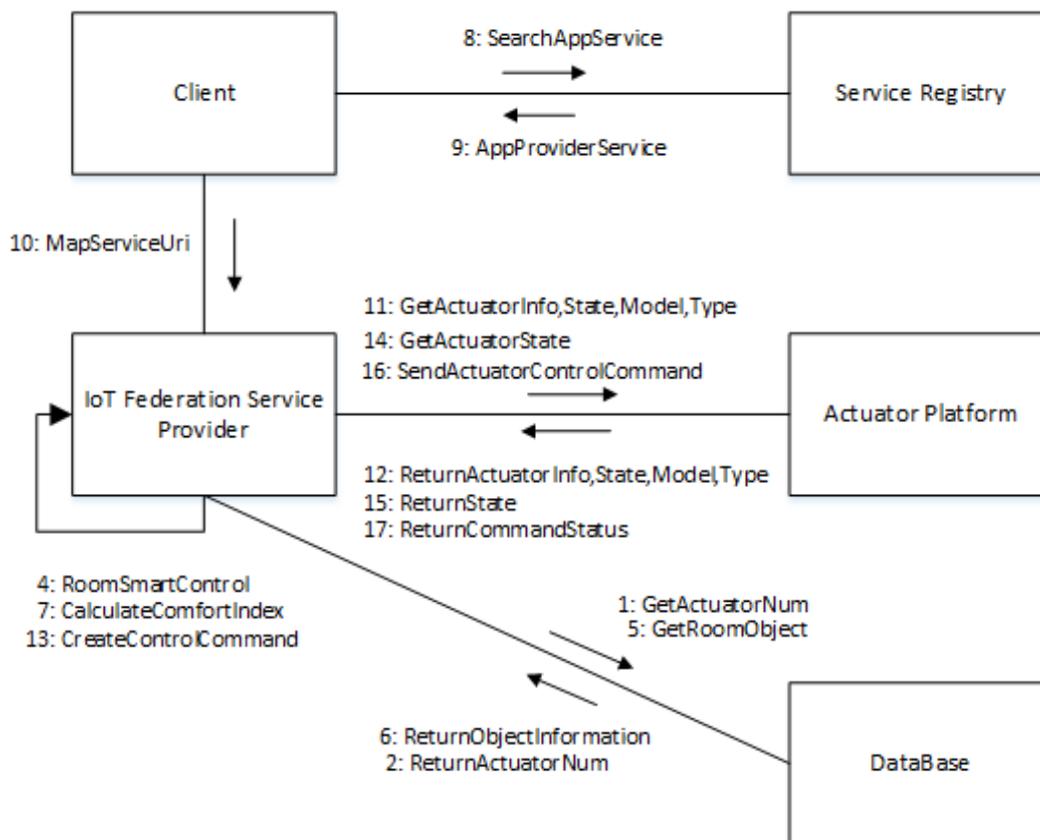


Figure 4. Collaboration Diagram of IoT Federation Service Provider

Figure 4 shows the collaboration diagram of IoT federation service provider. First step for federation is to query the status of all the actuators in the IoT federation service provider from the database, opens the provider service and runs the room smart control function. Room smart control gets the details of the sensors on specific selected room and calculates the comfort index of the room using the values read from sensors. Next it gets the current actuators in the room and their status, it matches the actuator's status with the required status in context to the user's comfort level. If the status of actuator doesn't match to users' requirements, then it creates the control command for the actuators and changes the state of the actuators.

Figure 5 shows the sequence diagram of IoT federation service provider for the interactions of the federation service provider to other modules. Federation service provider gets the total number of actuators in the IoT federation service provider from

database first to display and then starts the provider service, content service and the smart control. When the room smart control is initialized, it gets the room objects information from database and calculates the comfort index of the room. Then it gets the actuator information including actuator id, model, model information, type, and type information and actuator state from the actuator platform. After getting all the information of actuator, it compares the room comfort index with the actuator state and creates a smart control command with accordance to the comfort index, which is then sent to the actuator platform which further forwards it to the actuator in order to execute it.

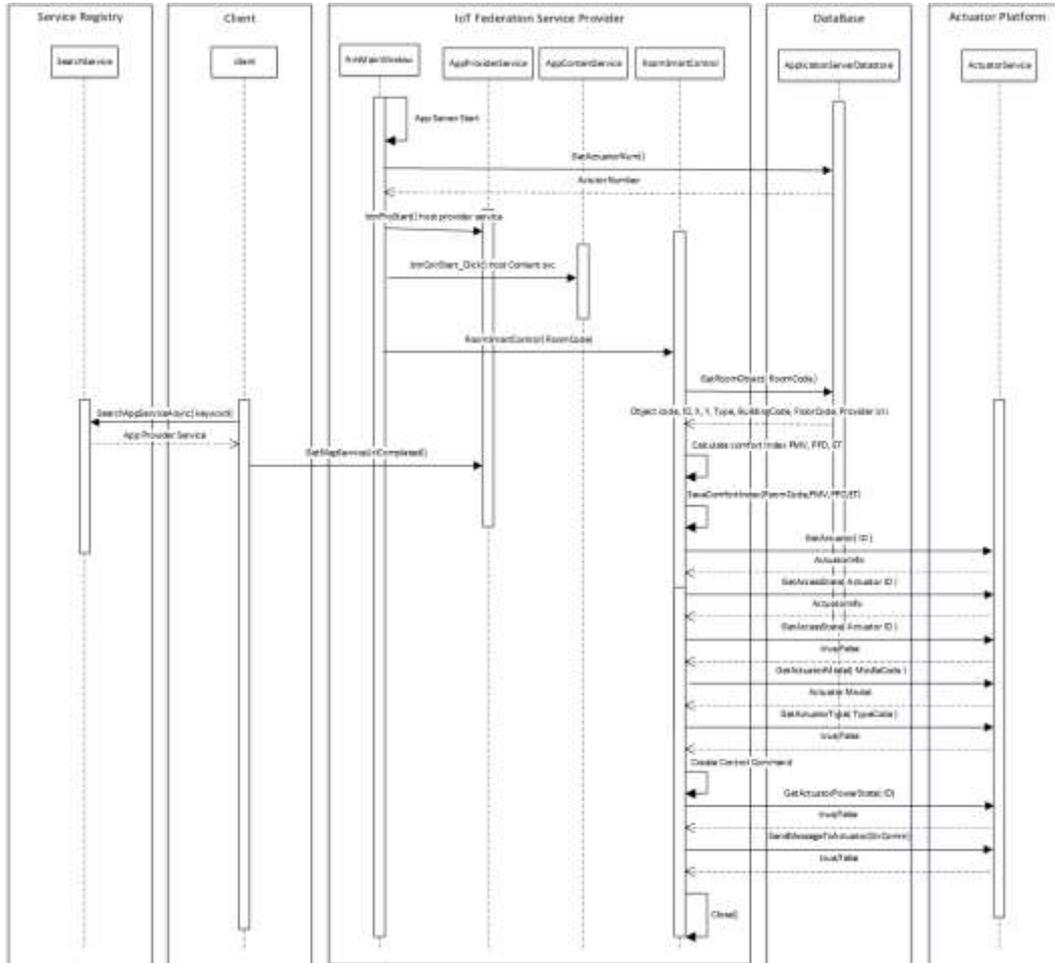


Figure 5. Sequence Diagram of IoT Federation Service Provider

4. Implementation and Performance Analysis

In this section we present the implementation environment and output results for our federation service provider deployed for actuator networks. Table 1 below shows the implementation environment.

Table 1. Implementation Environment

Components	Version
Operation System	Microsoft Windows 10 pro (x64)
Microsoft Visual Studio	2015
Microsoft SQL Server Management Studio	2016

Table 2 shows the Restful APIs for the service provider and content provider of the federation.

Figure 6 shows the federation service provider's execution screen. Actuator and GIS platforms can be connected with this federation by simply putting the URI for each. It allows independent platforms to be connected together and interact in order to facilitating the client with integrated actuator control. It shows the information saved in federation server's database such as object state, actuator count, taken from connected actuator platforms. An actuator count is the total number of actuator objects that are bound in database. The services provided by the federation as discussed earlier are provider service, content service and smart control. Provider service deals with the provision of object, actuator and map information while content service supplies the object information agent, and map binding agent. Services are run and shown real-time. Services have two states as service start and stop. Service start state is the operational state of a service while service stop is the non-operational state of the service.

Table 2. RESTful APIs for Actuator Control

Module Name	RESTful API Name
Federation Server	IAppServiceProvider_REST
	IAppContentProvider_REST

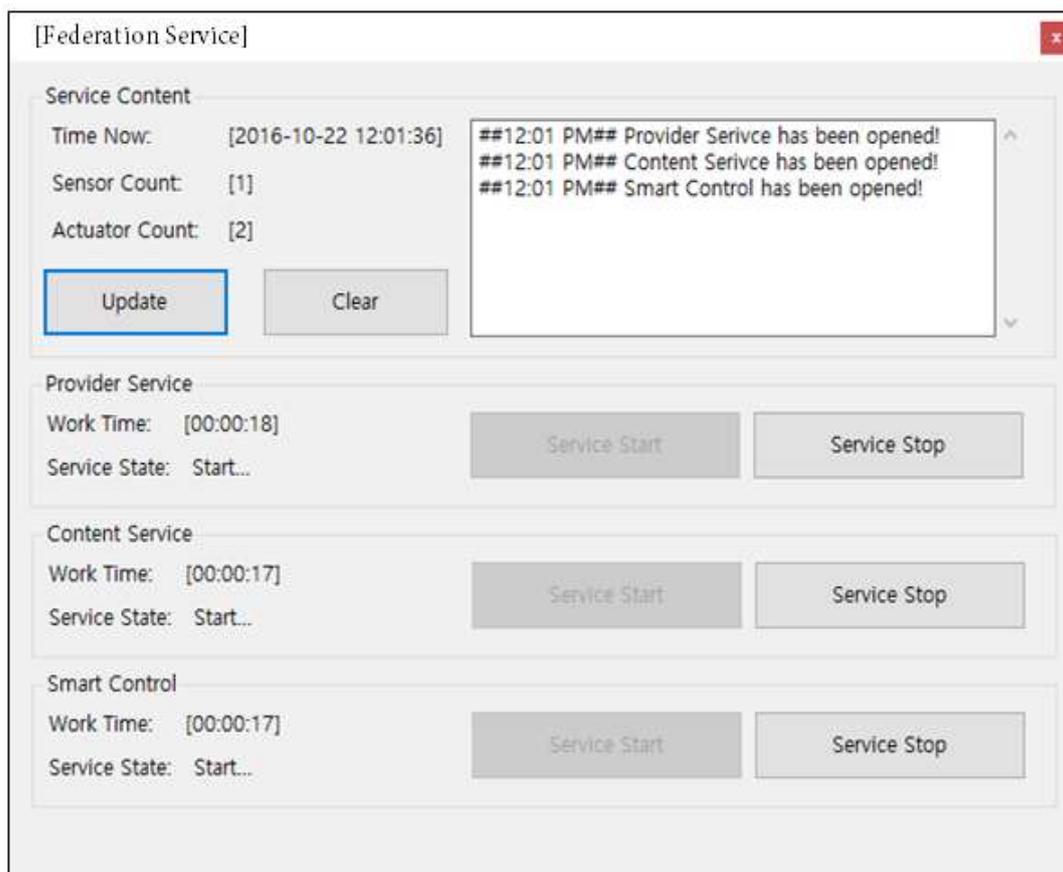


Figure 6. Federation Service Provider

In order to evaluate the performance of our implemented federation service provider, we have the service initiation and data access times with the functionalities implemented in RESTful web service. Figure 7, shows the performance of federation web services implemented using RESTful web services. It shows the execution times taken in ms for actuator's data access query initiated by federation service provider for actuator platform and client. The results clearly demonstrate that execution times for RESTful based web services and query requests take less than a milisecond on average.

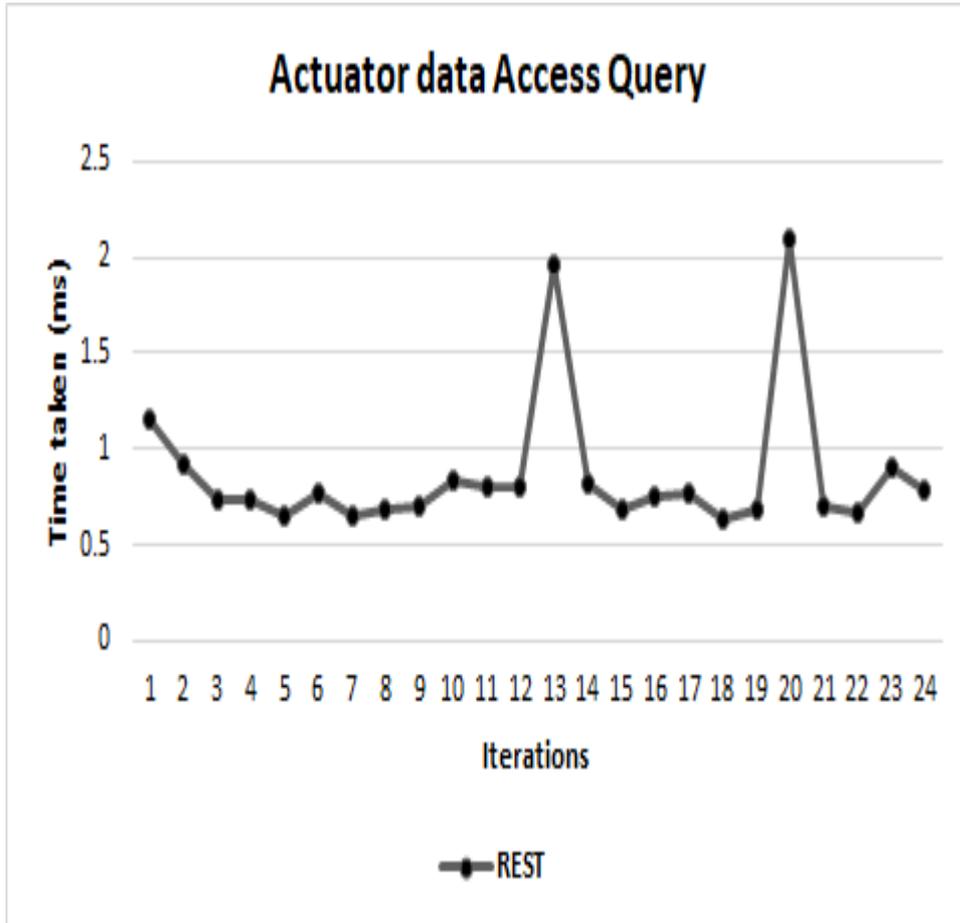


Figure 7. Performance of IoT Federation Service Based on Restful

6. Conclusion

In this work, we have presented an architecture for federation service provider, which acts as an application layer in the Web of Things based implementations. Our proposed architecture provides an independent federation layer that can be connected to multiple platforms, hence facilitating the implementation of web of things. Federation enables multiple platforms to get connected and connect client with indoor actuator environment, enabling the unified control for the actuator network. We have implemented the federation service provider based on RESTful APIs. We also have taken out the performance of IoT federation service Based on Restful for actuator control.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2017-0-00756,

Development of interoperability and management technology of IoT system with heterogeneous ID mechanism), and this research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-2014-0-00743) supervised by the IITP (Institute for Information & communications Technology Promotion), Any correspondence related to this paper should be addressed to DoHyeun Kim.

References

- [1] D. Guinard and V. Trifa, "Towards the web of things: Web mashups for embedded devices", Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences), Madrid, Spain, vol. 15, (2009).
- [2] G. Dominique, V. Trifa and E. Wilde, "A resource oriented architecture for the web of things", Internet of Things (IOT), 2010, IEEE, (2010).
- [3] A. Kamilaris, A. Pitsillides and V. Trifa, "The smart home meets the web of things", International Journal of Ad Hoc and Ubiquitous Computing, vol. 7, no. 3, (2011), pp. 145-154.
- [4] S. Mayer and D. Guinard, "An extensible discovery service for smart things", Proceedings of the Second International Workshop on Web of Things. ACM, (2011).
- [5] K.-H. Cheung, "A journey to Semantic Web query federation in the life sciences", BMC bioinformatics 10.10 (2009), p. S10.
- [6] C. Petrioli, "The sunrise gate: Accessing the sunrise federation of facilities to test solutions for the internet of underwater things", Underwater Communications and Networking (UComms), 2014. IEEE, (2014).
- [7] V. Trifa, "Design and implementation of a gateway for web-based interaction and management of embedded devices", Submitted to DCOSS, (2009), pp. 1-14.
- [8] D. Uckelmann, M. Harrison and F. Michahelles, "An architectural approach towards the future internet of things", Architecting the internet of things. Springer Berlin Heidelberg, (2011), pp. 1-24.
- [9] M. Zorzi, "From today's intranet of things to a future internet of things: a wireless-and mobility-related view", IEEE Wireless Communications, vol. 17, no. 6, (2010).
- [10] D. Guinard, "From the internet of things to the web of things: Resource-oriented architecture and best practices" Architecting the Internet of things, (2011), pp. 97-129.
- [11] R. Alarcón and E. Wilde, "RESTler: crawling RESTful services", Proceedings of the 19th international conference on World wide web. ACM, (2010).
- [12] C. Rodrigues, J. Afonso and P. Tomé, "Mobile application webservice performance analysis: Restful services with json and xml", International Conference on ENTERprise Information Systems. Springer Berlin Heidelberg, (2011).
- [13] C. Pautasso, "RESTful Web service composition with BPEL for REST", Data & Knowledge Engineering, vol. 68, no. 9, (2009), pp. 851-866.
- [14] H. Zhao and P. Doshi, "Towards automated restful web service composition", Web Services, 2009. ICWS 2009. IEEE International Conference on. IEEE, (2009).
- [15] F. Belqasmi, R. Glitho and C. Fu, "RESTful web services for service provisioning in next-generation networks: a survey", IEEE Communications Magazine, vol. 49, no. 12, (2011), pp. 66-73.
- [16] M. Fazio, "The need of a hybrid storage approach for IoT in PaaS cloud federation", Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on. IEEE, (2014).
- [17] S. De, "An internet of things platform for real-world and digital objects", Scalable Computing: Practice and Experience, vol. 13, no. 1, (2012), pp. 45-58.

Authors



Sehrish Malik, she received her B.S. in Computer Science from National University of Computer and Emerging Sciences (NUCES-FAST), Pakistan. In September 2014, she moved to Republic of Korea for M.S. studies and started working in the Protocol Engineering Laboratory (PEL), Sangmyung University (Cheonan-si Campus). After completing her M.S. in 2016, she moved to Jeju-do in September 2016, and started working as a Ph.D. research fellow in the Mobile Computing Laboratory (MCL), Jeju National University. Research interests include IoT, sensor networks, actuator networks, privacy, trust, and communication systems.



Do-Hyeun Kim, he received the B.S. degree in electronics engineering from the Kyungpook National University, Korea, in 1988, and the M.S. and Ph.D. degrees in information telecommunication the Kyungpook National University, Korea, in 1990 and 2000, respectively. He joined the Agency of Defense Development (ADD), from March 1990 to April 1995. Since 2004, he has been with the Jeju National University, Korea, where he is currently a Professor of Department of Computer Engineering. From 2008 to 2009, he has been at the Queensland University of Technology, Australia, as a visiting researcher. His research interests include sensor networks, M2M/IOT, energy optimization and prediction, intelligent service and mobile computing.

