Hindawi
Mobile Information Systems
Volume 2021, Article ID 5590605, 19 pages
https://doi.org/10.1155/2021/5590605

Research Article

Integrated Service Composition Approach Based on Transparent Access to Heterogeneous IoT Networks Using Multiple Service Providers

Wenquan Jin,[1] **Rongxu Xu**,[2] **Sunhwan Lim**,[3] **Dong-Hwan Park**,[3] **Chanwon Park**,[3] **and Dohyeun Kim**[2]

[1]*Big Data Research Center, Jeju National University, Jeju 63243, Republic of Korea*
[2]*Department of Computer Engineering, Jeju National University, Jeju 63243, Republic of Korea*
[3]*Autonomous IoT Research Section, Intelligent Convergence Research Laboratory, Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea*

Correspondence should be addressed to Dohyeun Kim; kimdh@jejunu.ac.kr

Received 16 January 2021; Accepted 11 May 2021; Published 28 May 2021

Academic Editor: Paolo Bellavista

The Internet of Things (IoT) enables the number of connected devices to be increased rapidly based on heterogeneous technologies such as platforms, frameworks, libraries, protocols, and standard specifications. Based on the connected devices, various applications can be developed by integrating domain-specific contents using the service composition for providing improved services. The management of the information including devices, contents, and composite objects is necessary to represent the physical objects on the Internet for accessing the IoT services transparently. In this paper, we propose an integrated service composition approach based on multiple service providers to provide improved IoT services by combining various service objects in heterogeneous IoT networks. In the proposed IoT architecture, each service provider provides web services based on Representational State Transfer (REST) Application Programming Interface (API) that delivers information to the clients as well as other providers for integrating the information to provide new services. Through the REST APIs, the integration management provider combines the service result of the IoT service provider to other contents to provide improved services. Moreover, the interworking proxy is proposed to bridge heterogeneous IoT networks for enabling transparent access in the integrated services through proving protocol translating on the entry of the device networks. Therefore, the interworking proxy is deployed between the IoT service provider and device networks to enable clients to access heterogeneous IoT devices through the composited services transparently.

1. Introduction

Internet of Things (IoT) is an emerging paradigm to connect a massive number of devices to provide intelligent and autonomous services in heterogeneous network environments for various industrial domains such as healthcare, factory, transportation, and building [1–3]. The traditional web services are provided by web servers through the Internet. However, most of the sensing actuating services are provided by constrained IoT devices that are deployed in the network edge and equipped with limited hardware resources such as battery-based power supply, low-performance computing ability, and limited network communication range [4–6]. The IoT device provides IoT services to expose the functions of sensors and actuators based on the IoT resources that are hosted in the device for providing access to the physical resources [7]. The IoT resource is included in the application of the IoT device that provides a corresponding Application Programming Interfaces (APIs) for accessing the handler of the resource [8]. For providing IoT

services in a constrained environment, the service-oriented architecture (SOA) can be a solution to support the simple and extendable developmental architecture to develop the application of the IoT device to expose the sensing and actuating functions to the Internet [9]. In the SOA-based implementations, the Representational State Transfer (REST) enables web clients to access the APIs transparently which is adopted by various IoT protocols such as Constrained Application Protocol (CoAP) and the Hypertext Transfer Protocol (HTTP) to support reliable and efficient communication [10–12]. Therefore, providing IoT services based on REST APIs enables interoperability in heterogeneous IoT entities.

For discovering the APIs to access sensors and actuators, many IoT platforms, frameworks, and libraries are proposed to enable the IoT devices are available and accessible by web clients through exposing the IoT services to the Internet [13–16]. The IoT services deliver the environmental sensing data to the clients and control commands to the actuators which are the fundamental functionalities of an IoT network. For representing the IoT devices on the Internet, the management mechanism is required to provide registration and discovery services to the web clients [17, 18]. The registered information of IoT devices enables the context of the IoT devices to be accessed for understating the environment. The management of IoT devices is comprised of several management functionalities such as management of device, fault, configuration, network, and firmware that enables the IoT devices to provide services stably without human touch in the constrained environment [19, 20]. The management platform provides a set of services that are associated with the IoT resources and the environment where the IoT devices are deployed. Therefore, REST APIs enable the management services to interact with IoT resources as well as other services based on delivering information through simple interfaces.

Through the management of IoT devices, the functions of sensors and actuators are available on the Internet and accessed based on the REST APIs. Although, the heterogeneity of IoT devices requires different clients to the IoT resources. However, the interworking proxy enables transparent access to heterogeneous IoT devices in different network environments without considering the underlying protocols and platforms [21, 22]. With the interworking proxy, the IoT services can be delivered by the consistent interface from the heterogeneous IoT networks which enables the IoT services can be used by various service consumers to provide improved services for various industrial domains. Leveraging the service composition mechanism, the domain-specific service objects can be combined with the IoT services to enhance the systemic ability through integrating the components [23–26]. The IoT device management provides transparent access to the IoT devices to service consumers through REST APIs based on the interworking proxy. Then, the contents of the domain-specific service providers are enabled to integrate with the IoT data to provide improved information to users.

In this paper, we propose a service composition approach in the IoT architecture that combines various services from multiple service providers for providing improved IoT services based on the integrated service composition. The providers provide web services through REST APIs that deliver information to the clients and other providers which enables the service composition to present integrated services for various industrial domains. For combining the service contents with the IoT services, the integrated service composition provider is proposed to integrate the contents of REST APIs with the resources of sensors and actuators. Moreover, the IoT and location service providers are proposed to provide the management of IoT resources and location-based content for exposing the REST APIs on the Internet. Also, the interworking proxy is proposed to bridge the IoT service provider and heterogeneous IoT networks to enable transparent access by the composited services. For experimenting with the proposed IoT architecture, the integrated service composition, IoT service, and location service providers are implemented based on web server applications and deployed on the Internet to provide web services that are consumed by web clients as well as by service providers. The interworking proxy is implemented to serve in the entry of IoT networks to enable transparent access to the IoT devices that provide sensing and control services in constrained IoT networks such as CoAP and IoTivity.

The rest of the paper is structured as follows. Section 2 introduces the various IoT systems to review the management characteristics and IoT service composition approaches. Section 3 introduces the integrated IoT service architecture for the proposed service composition in heterogeneous IoT networks. Section 4 presents the integrated service composition approach based on multiple service providers. Section 5 presents the development results and performances of the proposed IoT scenario for integrated service composition based on IoT and location service providers. Section 6 presents a comparison with the existing solutions for emphasizing the significance of the proposed IoT architecture. Finally, we conclude this paper and introduce our future directions in Section 7.

2. Related Works

Recently, many IoT solutions are proposed with specified system architectures for specific industrial domains or general usages. For providing a generic IoT architecture, European researchers proposed the IoT-A to derive a concrete IoT architecture using the architectural reference such as models, views, perspectives, and best practices [27, 28]. The European Telecommunication Standard Institute (ETSI) proposed a standard IoT architecture for developing a high-level IoT service platform that is comprised of domains of application, network, and device [29, 30]. Ren et al. [31] proposed a scalable IoT architecture based on transparent computing to enable decoupling the software from the various devices for making the service provisioning to be transparent by users. Lloret et al. [32] proposed an integrated IoT architecture for serving the smart cities based on various system elements such as communication, data gathering procedure, and the decision

system. Desai et al. [33] proposed an IoT architecture to provide interfaces based on communication and data standards using a semantic gateway for supporting interoperability between various service providers. Datta et al. [34] proposed a gateway-centric IoT architecture that provides several fundamental functions for IoT systems including device discovery, connections with IoT devices, and metadata definition for sensing and actuating. These IoT architectures enable interoperability with the devices and other services based on the web services. Adopting the REST APIs in the proposed interworking proxy enables applications to consume the web services for providing customized content to the users.

In the comprehensive management system for IoT networks, monitoring, fault detection, network configuration, and device management are included to cover overall functional requirements [16, 35, 36]. For configuring the IoT devices to expose the IoT services on the Internet, device management provides interfaces to create, retrieve, update, and delete the virtual entities of IoT devices in the cyber world [37–39]. Also, the management enables the components of the IoT network to combine the content to provide improved services [40]. Kim et al. [41] proposed an IoT device management protocol based on the self-configuration process in the device to configure the device in the visible light communication network. Maloney et al. [42] proposed an IoT device management architecture based on the agent to deliver the updates to the IoT devices for synchronizing the configuration between the server and devices. Bera et al. [43] proposed device management based on a software-defined networking approach to manage device resources through activating services using the software-defined controllers. For secure IoT device management, the blockchain mechanism enables the synchronization of the IoT device information in an IoT network to provide the identification policy [44]. Ferreira et al. [45] proposed a device management solution based on ontology to register IoT devices for semantic discovery. The oneM2M is a large-scale IoT standard that is implemented by OCEAN through Mobius and CUBE modules [46–48]. These management solutions enable the various devices are available on the Internet to be accessed.

However, for overcoming the heterogeneity of the IoT networks, the interworking entity is important to translate the different protocols [49]. The gateways can be deployed in the entry of networks to manage the IoT devices that are deployed in the networks to provide IoT services. ThingsSpeak is an IoT middleware platform that is deployed between clients and devices to manage the device and data through APIs. The APIs of the ThingsSpeak platform are provided to IoT devices as well as clients which enables receiving real-time sensing data in the platform [50, 51]. For real-time data collection, the KAA platform is deployed in the cloud to serve the IoT system as a middleware that provides end-to-end services to IoT devices as well as clients [52]. The gateway-based IoT platforms are easy to integrate multiple IoT networks through the REST APIs between applications and IoT devices. The Open Mobile Alliance (OMA) proposed a device management specification for managing devices using a server-client architecture that is implemented by Lightweight M2M (LwM2M) to provide communications of constrained IoT devices through CoAP [53–55]. The Open Connectivity Foundation (OCF) proposed an IoT standard based on REST APIs that are implemented in IoTivity that is a lightweight framework to develop the IoT devices [56–59]. The LwM2M and IoTivity support the CoAP as the main protocol that provides the interworking approach based on the CoAP proxy through mapping the properties [60]. This mechanism is adopted in the proposed interworking proxy to translate different protocols.

REST APIs are provided by service providers that expose services to the Internet for enabling web clients to access the data. Based on REST APIs, service composition provides the interoperability between heterogeneous contents in IoT networks [61]. Berrani et al. [62] proposed an IoT service composition approach based on deploying multiple agents to provide different types of services and integrating the services in the IoT application. Akasiadis et al. [63] proposed a framework to develop complex IoT applications using the SYNAISTHISI platform that enables extracting the information from the IoT services based on ontologies and using the information in the service composition [64]. Mohammadi et al. [65] proposed a smart city solution that integrates the smart object to the location services based on the composition of location, deep learning, and IoT services. Sun et al. [66] proposed a dynamic service composition approach based on optimization approaches to find the optimal set of IoT services for reducing the energy consumption in IoT networks. Wang et al. [67] proposed an IoT architecture that balances the traffic load and enables a longer lifetime of the whole system based on the interaction using REST APIs of constrained web resources for energy efficiency. Most of the service composition approaches in IoT networks integrate domain-specific services to the IoT services for providing intelligent, autonomous, and rich information. The REST APIs provide the links to service consumers for combining and modifying the contents to provide the enhanced information to clients [68–70]. However, the heterogeneity of the IoT networks causes the consistent interfaces to be not provided by the REST APIs.

3. Integrated IoT Service Architecture

The proposed integrated service composition approach is comprised of IoT services and domain-specific services that are provided by service providers through the Internet. The service providers are deployed in the server machines to provide services to their clients. We propose the overall IoT architecture including the interactions and entities such as users, clients, servers, and devices. The entities are presented in Figure 1 through five layers to illustrate the IoT service architecture.

The IoT service layer includes the IoT service provider that is comprised of management and connectivity functions to provide service to its clients and other providers. The IoT service provider enables the IoT devices are available on the Internet by representing the physical entities in the cyber

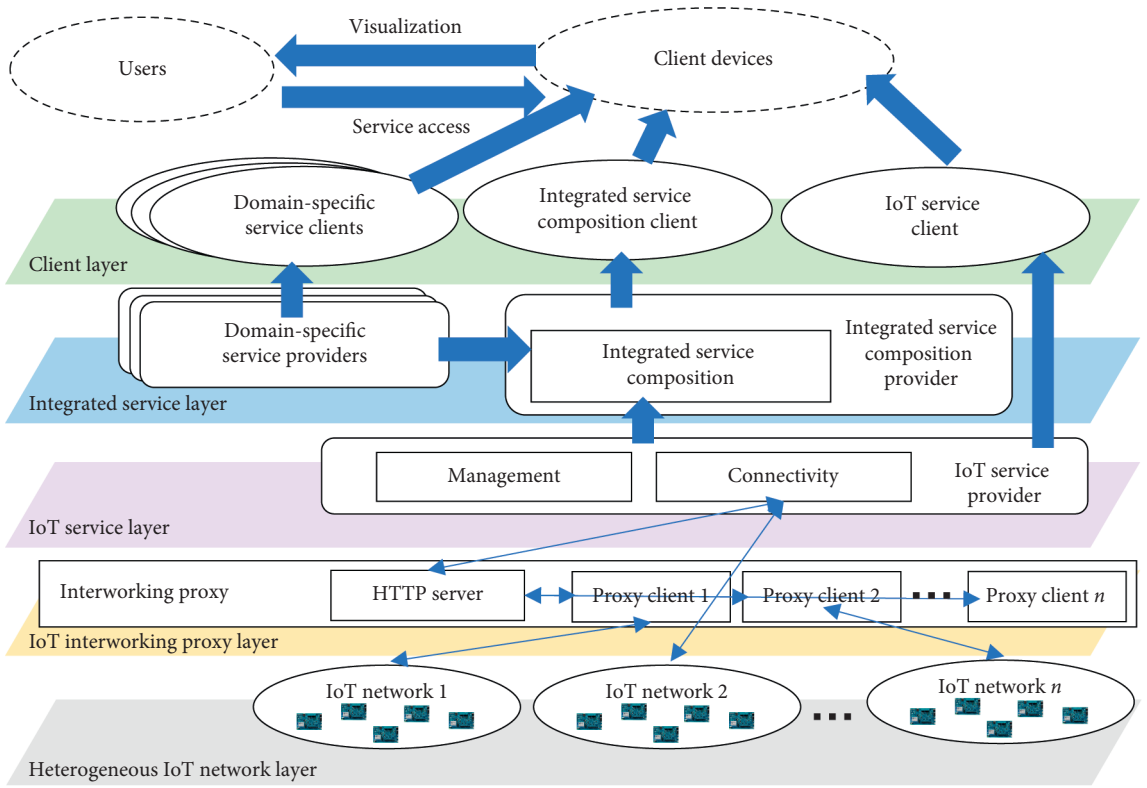

Figure 1: Hierarchical integrated IoT service architecture.

world. The management service provides discovery and registration of IoT devices through saving and retrieving the information of IoT devices using the database. The connectivity function provides services to access the IoT devices which can be provided through the interworking proxy and directly. The provided services of the IoT server provider are consumed by the entities of integrated service and client layers based on the client functionality. The integrated service layer includes multiple service providers that provide domain-specific services such as transportation, healthcare, buildings, farm, and other industries. Also, the integrated service composition provider is included in this later to integrate the domain-specific services with IoT services for providing integrated services. The domain-specific services and IoT services deliver the information to the integrated service composition which creates new information based on the service composition. Each service provider has its service client that is used for presenting the User Interfaces (UIs) to interact with users.

The service client of the integrated service provides the UIs for creating, updating, retrieving, deleting, and accessing the integrated services. In the client layer, we include various service clients that provide the UIs to the users through the client devices. For the domain-specific service providers, the UIs are developed for presenting the information and functions that are provided by the service provider. The service client can be an application that is implemented for a platform such as Windows, Android, and Linux. Also, the application can be a web client that is delivered from the service provider as UI services. For providing the integrated services using the service composition, the domain-specific information is required through the domain-specific services to generate. Then, the integrated service composition client requests the domain-specific information through the service composition provider to integrate into the IoT resource. In this interaction model, the services are provided through the REST APIs that deliver the information between clients and servers to perform creating, retrieving, updating, and deleting for the improved services based on service composition.

In the proposed IoT architecture, the service providers, interworking proxy, and IoT devices include a server to provide services to the service clients such as web clients, servers, and devices that include the REST API client to request the servers. The client and server architecture brings the relationship of cooperating programs in web applications. The server application provides a set of functions or services to one or many clients that send the request message through the networks.

As shown in Figure 2, the proposed IoT architecture is comprised of multiple service providers to deliver the information using the client-server model which enables the integrated IoT services. The service clients request the service providers that are deployed in the server machine to provide the REST APIs through the Internet. In the proposed IoT architecture, integrated service composition, domain-specific service, and IoT service providers are proposed to provide services. The domain-specific service and IoT service providers are accessed by its service client and service composition provider. In the integrated services, the IoT

resources are accessed through the services that are provided by the IoT service provider. For accessing the constrained IoT devices, the interworking proxy is deployed between the IoT service provider and IoT devices to translating the request and response messages. We assume some IoT devices are configured with sufficient hardware to run general libraries and frameworks. The HTTP-based IoT device can be a typical device to use a general communication protocol to consume more network resources.

4. Integrated Service Composition Approach Based on Multiple Service Providers

4.1. Integrated Service Composition Scenario for Improved IoT Service. The proposed service composition is used for providing integrated services based on integrating IoT services and domain-specific services. The IoT service provider provides information regarding IoT resources, and the domain-specific service provider provides the information of domain-specific information such as location information.

In Figure 3, the integrated service composition flow is presented where the location service provider is considered as the domain-specific service provider to provide location-related information. The provided information includes IoT and location that are integrated based on the service composition and provided as new services.

Figure 4 shows the IoT device management scenarios using the IoT service provider that provides services to provide device discovery, registration, update, delete, and access. A user can access the index page that is provided by the IoT service client. The client provides the index page with the registered device list that is provided by the service provider. Using the registration page, the user can fill in the information of the IoT device and submit the data to the service provider. Registered IoT devices can be retrieved by the service client and modified by the user to update the information of the IoT device. The user gets the sensing data and sends the control command through the service client. For accessing the IoT device, direct access and proxy-based access are provided in the IoT architecture. Based on the IoT service provider, the IoT devices are configured and provided to the Internet through REST APIs. Using the exposed REST APIs, the integrated service composition provider integrates the IoT and location information to generate new services.

Figure 5 shows the location management scenario using the location service provider to configure the outdoor and indoor location information. The service provider provides services to create the outdoor location based on the map service. The provided UIs are presented by the service client with outdoor map information. Firstly, the user creates an outdoor location by making the location on the map and submitting the location information to the service provider. Then, the user uploads the indoor map plan and submits the extra information for describing the indoor location information. Using the location services, buildings can be presented to users for understating the structure of user spaces which enables the things of spaces to be visualized on outdoor and indoor maps.

Figure 6 shows the sequence that depicts the integrated service composition scenario based on multiple service providers including IoT service provider, location service provider, and integrated service composition provider. Through the IoT service provider and location service provider, the information of IoT and location is created and provided based on REST APIs. For integrating the IoT resource to the location, the location information is retrieved and combining the IoT resources to the location as nodes in the indoor environment.

The integrated service composition provider provides the interface to the UIs for presenting the created location information through interacting with the location service provider. The locations are retrieved and displayed in the composition client. The location types are outdoor location and indoor location, respectively. Firstly, the outdoor locations are retrieved based on the map service. Then, through accessing an outdoor location to retrieve the indoor locations based on a plan to visualize a floor of the building. The combine function is provided by the composition provider which combines the IoT resource to the indoor location. In the indoor map, the user can mark a node on the map and combine an IoT resource to the node. In this step, the IoT service provider provides the IoT device information to the service composition client. Once the node is created, the mapping information is saved in the composition provider. Then, using the composition client, the user can get the sensing data based on the IoT resource that is visualized in the indoor map.

4.2. Functional Architecture Based on IoT and Location Service Providers. The proposed service providers are deployed in the web servers that provide web services to the web client. The web services are provided for presenting the contents that are categorized by functions and interfaces. The functions are exposed on the Internet through REST APIs. For service composition, the REST APIs are important to combine the information in the integrated service composition provider. The interfaces are used for presenting the information in the service clients based on UIs. Each provider provides an index page to access the functions and interfaces.

Figure 7 shows the functional architecture of the IoT service provider. The provider provides functions of creating, retrieving, updating, and deleting to manage the IoT device information. Through the index page, the registered devices are presented as a list. Each item of the list provides the links to access the pages of detail, updating, and creating. Also, the deleting function is provided. In the detail page, the service client accesses the IoT networks. The IoT networks are heterogeneous due to the various frameworks and platforms. The page provides the detail information of the IoT device that includes REST APIs of the resources. In the updating page, the detail information of the IoT device is included for updating the device through the updating function. For creating information of IoT device, the form is provided through the creating page. The user can fill the form and submit the data to the provider for registering the

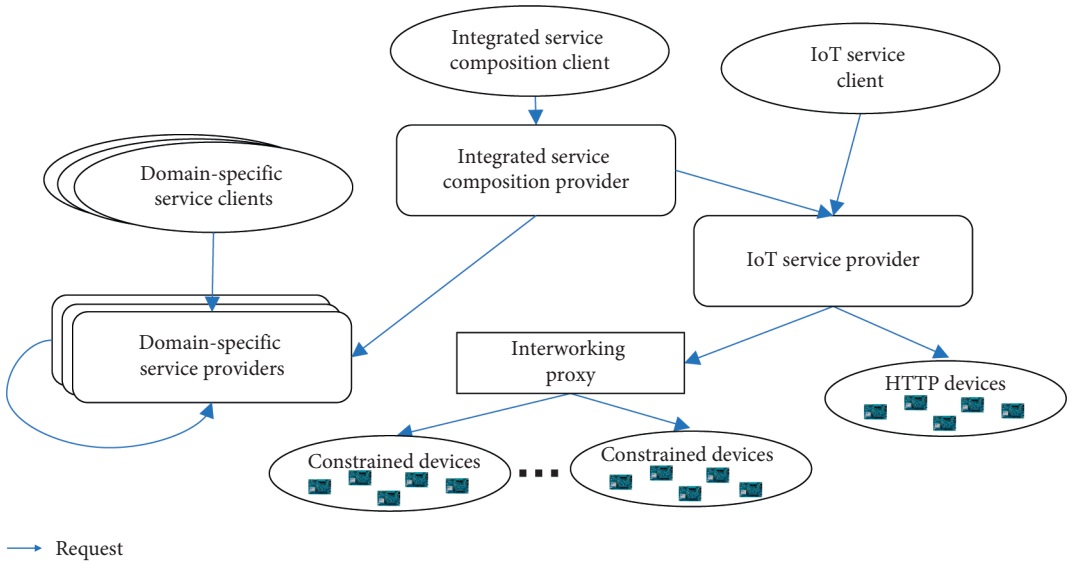

Figure 2: Integrated IoT service configuration based on multiple service providers.

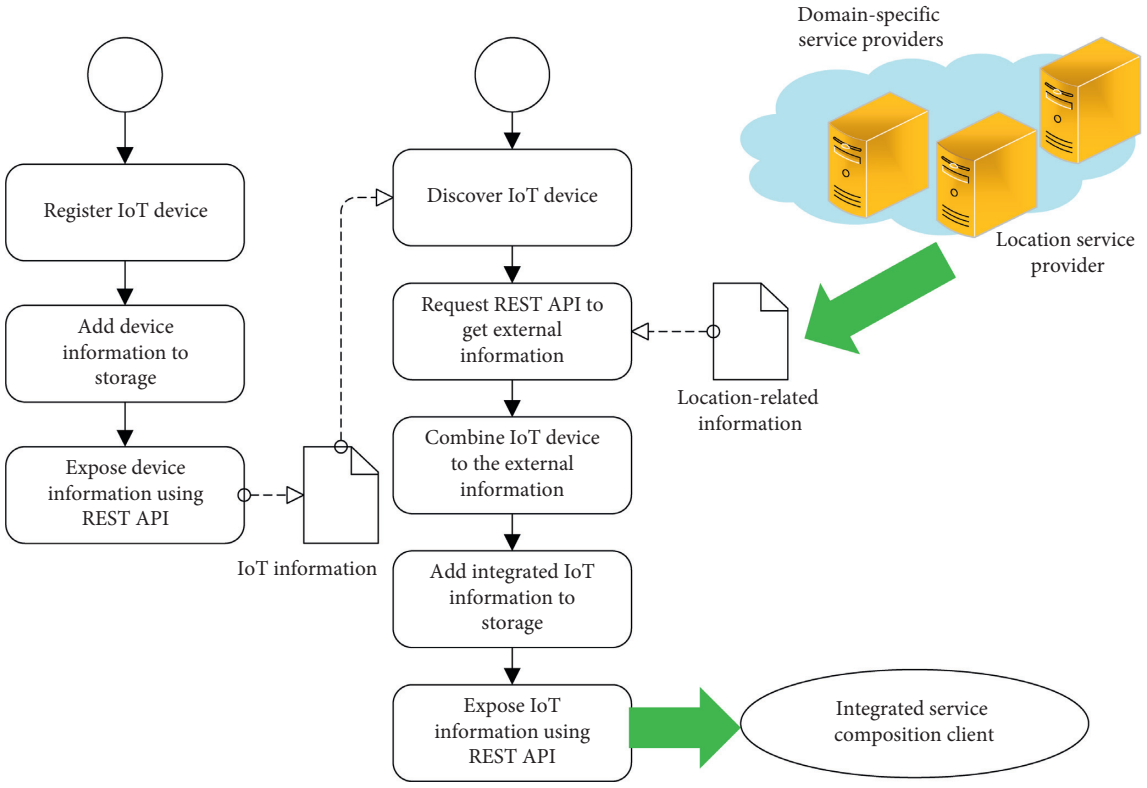

Figure 3: Integrated service composition flow.

IoT device. Through the REST APIs, users can manage IoT devices by other web clients.

Figure 8 shows the functional architecture of the location service provider. The location service provider provides created outdoor locations on the index page. The pages of creating and detail are provided in the index page to create a new outdoor location based on map APIs and a new indoor location by uploading new plans. The detail page provides the detail information of the outdoor location including the floors. Also, updating and deleting functions are provided to modify the outdoor location. For visualizing the IoT device in the outdoor and indoor map, the service provider provides layered maps for the spaces such as buildings where the IoT devices are deployed.

Figure 9 shows the functional architecture of the integrated service composition provider. For providing the

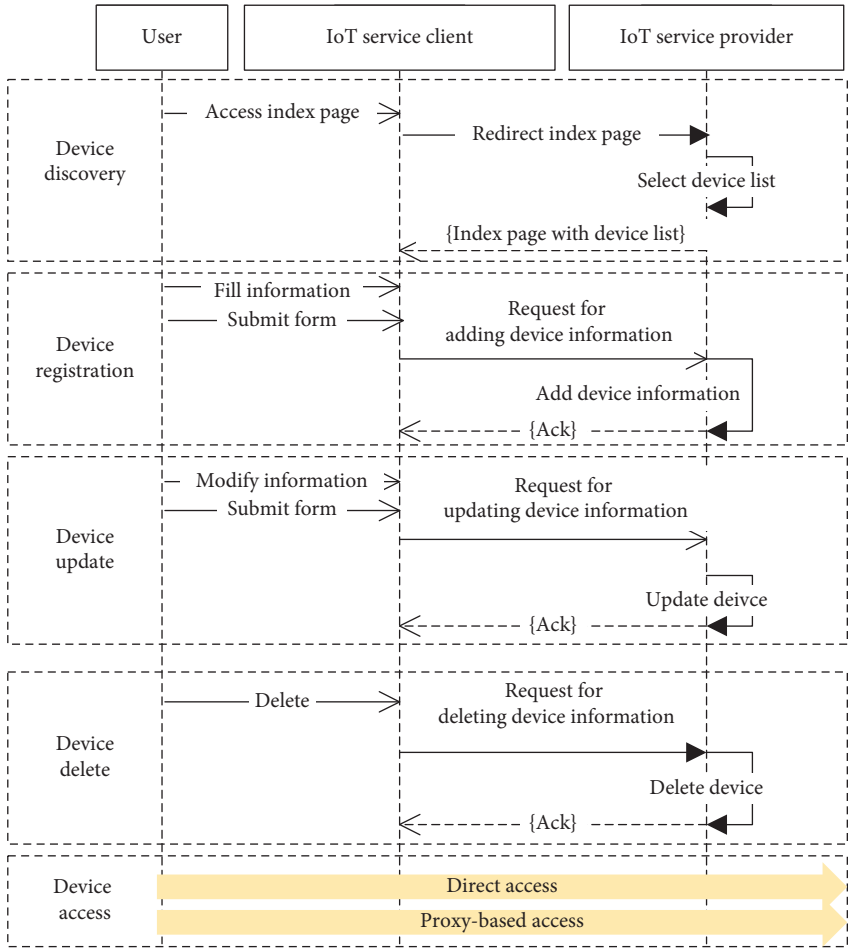

Figure 4: IoT device management scenario using IoT service provider.

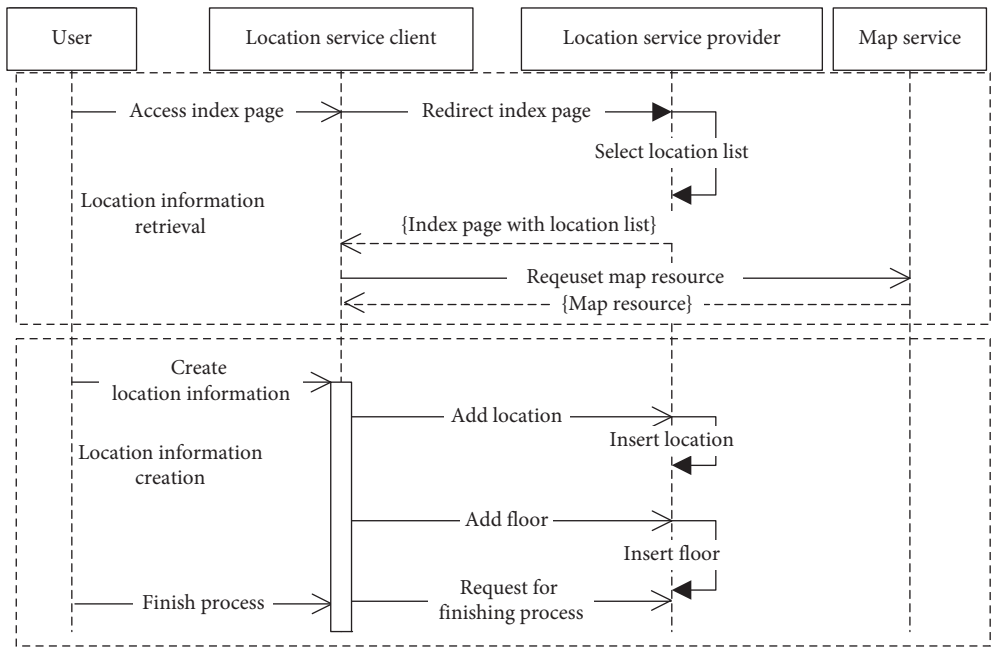

Figure 5: Outdoor/indoor location management scenario using the location service provider.

integrated services in the proposed IoT architecture, the integrated service composition provider is deployed in the server machine to provider services. The provider forwards the services that are provided by the IoT and location service providers. The index page presents outdoor locations on the map which are retrieved by the floor IDs. The IoT devices are deployed in the floors and combined to the information of floors in the cyber world. Using the node creation interface, the user can combine IoT devices to the locations. In the interface, the registered location list is provided based on the location service provider. The user selects an outdoor location to enter a floor interface with the floor information and links for accessing other floors. In the selected floor, the user can combine an IoT device by requesting the device information based on REST APIs that are provided by the IoT service provider. The IoT service provider provides services to retrieve the IoT devices and access the resource to get sensing data and control actuators. The selected IoT device can be combined through the function "Add Node" that provided the service composition provider through the REST API.

4.3. Interworking Proxy for Heterogeneous IoT Networks. For enabling transparent access by the services based on the integrated service composition provider, the interworking proxy is deployed in the entry of the IoT networks to provide the translation functionality. The heterogeneous IoT networks are configured using different development environments that require different interfaces to access these networks. The proposed interworking proxy enables to access heterogeneous IoT networks using a consistent interface.

As shown in Figure 10, the interwork proxy is deployed between IoT service provider and IoT device. The IoT services are provided to the Internet by the IoT service provider. For acceding the IoT devices using the deferent protocol, the proxy provides the interface to forward the messages to the IoT network where the devices are deployed. Based on the access mechanism in the IoT service provider, the provider selects an access scheme between direct and proxy-based access. For accessing the HTTP-based IoT device, provider forwards the request to the IoT device directly. However, non-HTTP devices cannot be accessed by the request directly. Then, the provider generates a request that includes the IoT device URI as the query in the request URI. In this process, the HTTP server in the proxy receives the request message from the provider and generates a corresponding request using the proxy client. The HTTP server has one or more handlers to translate the HTTP messages to other messages.

Figure 11 shows the architecture of the interworking proxy that enables transparent access for the service client. The IoT services are accessed by the IoT service client and integrated services through the integrated service composition client. The IoT service provider includes the HTTP client to forward the communication messages to the IoT devices. The HTTP-based device is accessed by the HTTP client directly. The other protocol-based devices such as CoAP and IoTivity require the interworking proxy to translate the communication messages.

5. Development Results and Performances

The proposed IoT architecture is comprised of service providers, interworking proxy, and IoT device. The service providers are deployed in the server machine to provide web service through the Internet. The interworking proxy is deployed in the entry of the local network where the IoT device is deployed with sensor and actuator.

Table 1 presents the development environment to introduce the implementation details of the proposed entities. The service providers are web applications that are implemented based on Spring MVC framework to provide web services. For providing reading and writing functions in the server application, MyBatis is included to interact with the My SQL database. Based on the basic reading and writing functions, the providers are implemented to provide the management for services, locations, and IoT devices. Interworking proxy and IoT device are implemented for the small device that works on the constrained environment using limited resources. The hardware of the small device is Intel Edison Board, and the system is Android Things. For handling the request from the Internet, the interworking proxy is implemented based on the Jetty to provide HTTP services. The IoTivity and Californium frameworks are used for forwarding the HTTP messages to the IoT device. For experimenting the proposed IoT architecture, 3 types of IoT devices are developed including devices of IoTivity, CoAP, and HTTP.

As shown Figure 12, the HTTP-based IoT device is accessed by the IoT service provider directly and others are accessed through the interworking proxy which is developed based on CoAP and IoTivity. The IoT devices are registered to the IoT service provider that enables the IoT devices to be available in the Internet. Then, the discovery service provides the information to the client for accessing the environment through the IoT resources. In this implementation, the client can be the IoT service client and integrated service composition client that are used for presenting the environmental data to the users. The composition provider combines the services of location service provider and IoT services provider and provides the integrated service to the client. Through the composition client, the IoT data are presented based on outdoor and indoor maps.

Figure 13 shows the implementation result of IoT service provider that is used for providing the IoT device information to the clients for accessing the environment through the IoT resources. The provider provides services to deliver the data to the clients including IoT service client and integrated service composition client. However, the data are delivered by REST APIs which enables the presentations of data which are different in IoT service client and integrated service composition client. For accessing an IoT device, firstly, the user accesses the list page that is used for presenting the list of registered IoT devices. In this implementation, we implemented four IoT devices with different specifications such as protocols and equipped sensors. Once

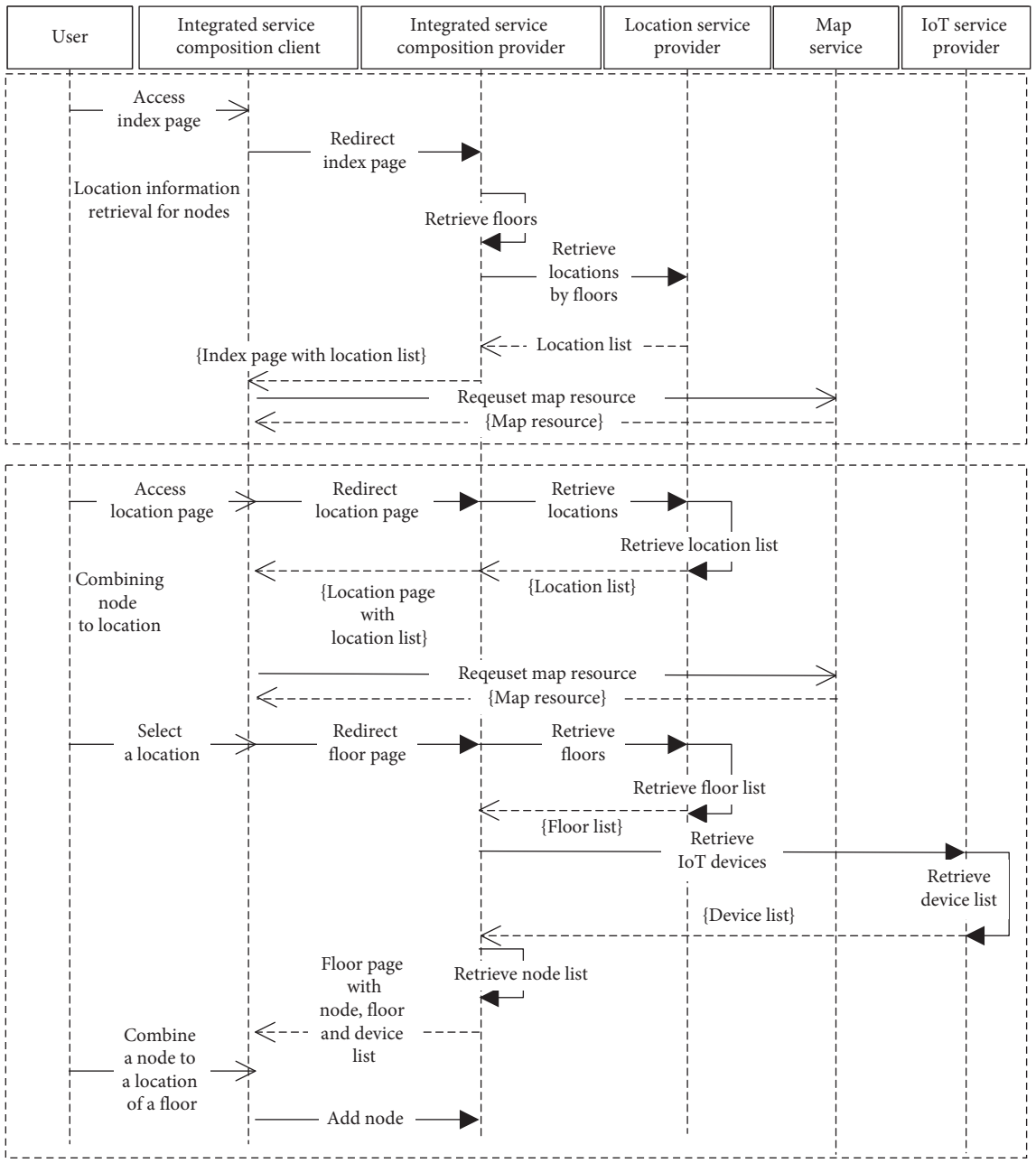

Figure 6: Integrated service composition scenario based on multiple service providers.

a device is selected, the service delivers the detail information of the device. In the detail page, the user can get the sensing data or control the actuator based on the presented information of the IoT device. The information also is provided to the integrated service based on the REST API.

For providing the location-based service to integrate IoT and outdoor/indoor maps, the geographical information and indoor location of IoT device are enabled to be registered and presented through the location service client as shown in Figure 14. The outdoor location represents a building in the outdoor map that is provided by Google Maps. The registered outdoor locations are presented by the client through a list as well as markers in the map. By selecting a location, the user can register an indoor maps through uploading the plan of floors. The registered indoor maps present the details of the building that is located in the outdoor map.

Through integrating the services from IoT and location service providers, the improved services are provided to the client. For integrating services, the service management is implemented to consume the services based on REST APIs. The client of composition provider provides the integration functions and integrated services as shown in Figure 15. The client presents the outdoor location of a building in the map. The user can access the indoor maps through the outdoor location and select a floor to mark the indoor location of an IoT device. The integrated object is a node that is comprised of location and IoT data. The user can discover the node

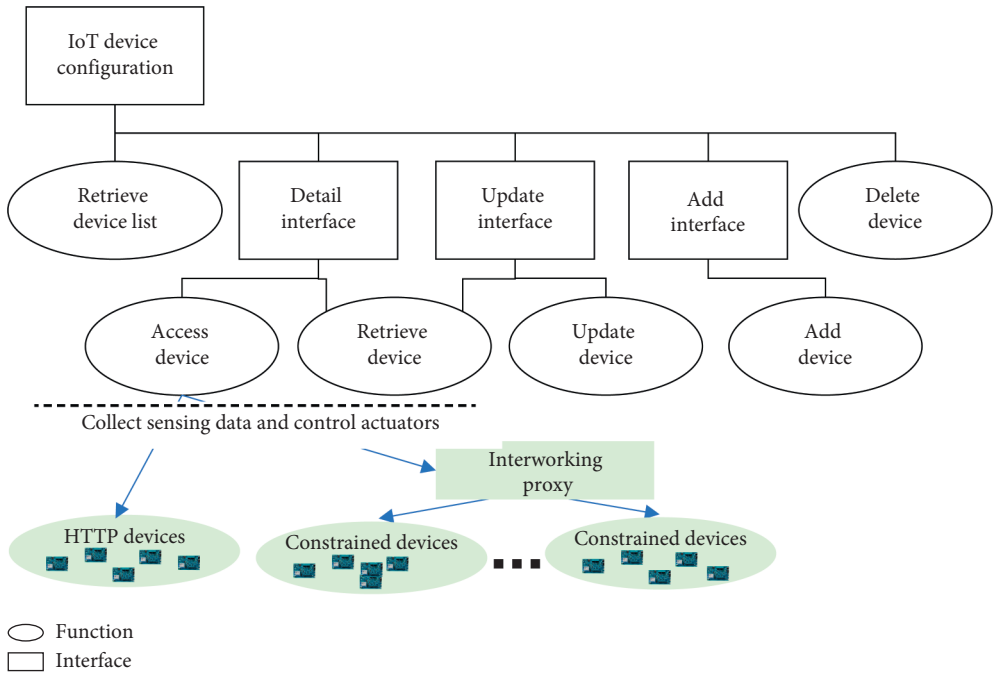

Figure 7: IoT service provider functional architecture.

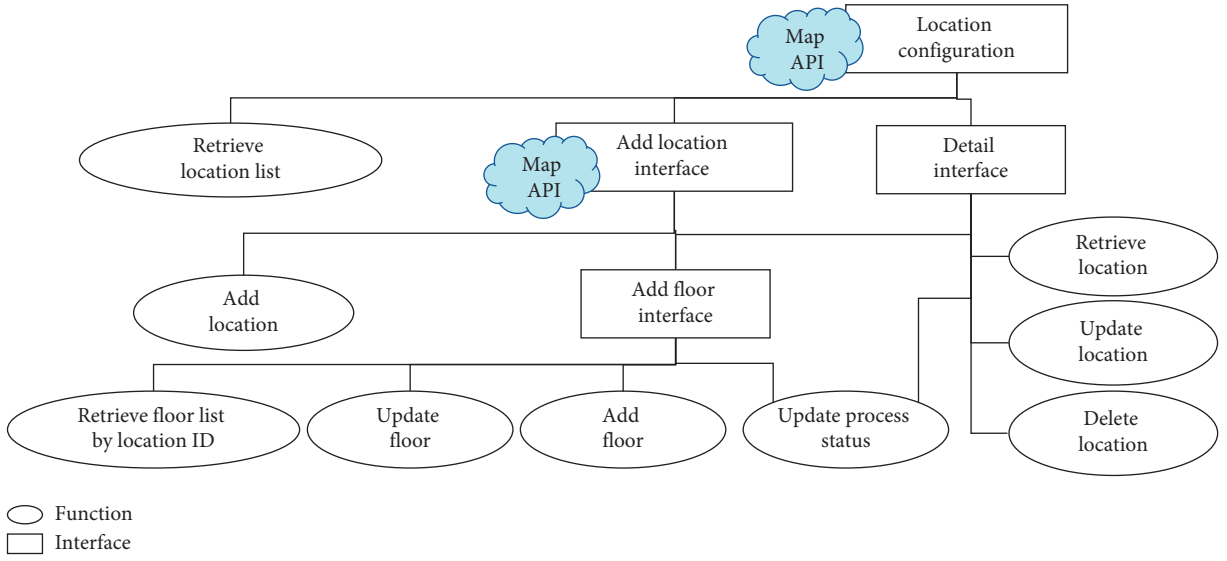

Figure 8: Location service provider functional architecture.

from the outdoor map that presents the outdoor locations. By selecting the location, the floors are listed to provide the indoor locations of the nodes. A floor is the indoor map that presents the registered nodes in the plan. The user can select a node to access the detail information that is delivered by the REST API of the IoT service provider. Through the information, the client accesses the IoT device. The request is delivered by the REST API of the IoT service provider to the IoT device. When the network of IoT devices is provided for IoT specific networks such as CoAP and IoTivity, the interworking proxy enables transparent access for the heterogeneity of IoT networks.

Figure 16 shows the logging result of accessing IoT device through the proposed interworking proxy. The request is sent by the integrated service to the IoT service provider and delivered to the IoT device through the interworking proxy. Figure 16(a) shows the logging in the IoT service proxy that presents the URI of the request. The request URI is used for accessing the interworking proxy which includes the destination IoT device as the parameter. In the parameter, the prefix "coap" is used for identifying the destination protocol. The interworking proxy translates the HTTP request to the CoAP request based on the prefix.

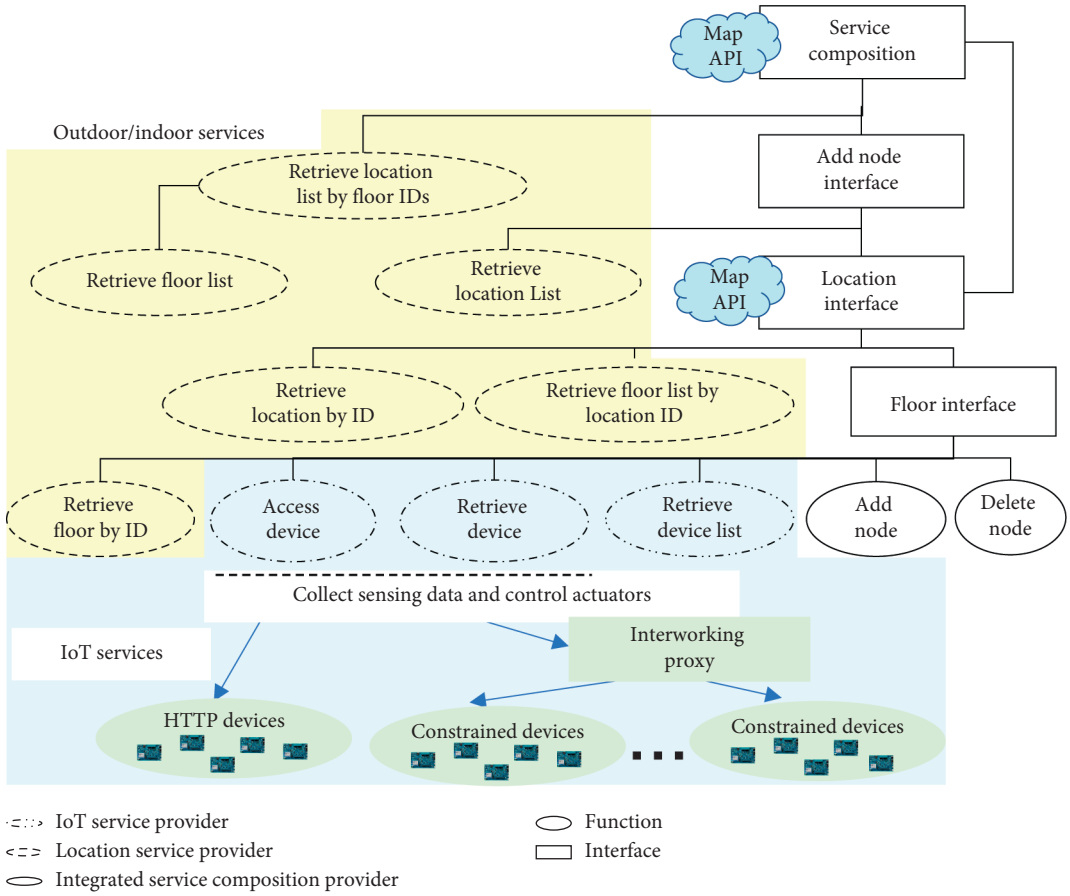

Figure 9: Integrated service composition provider functional architecture.

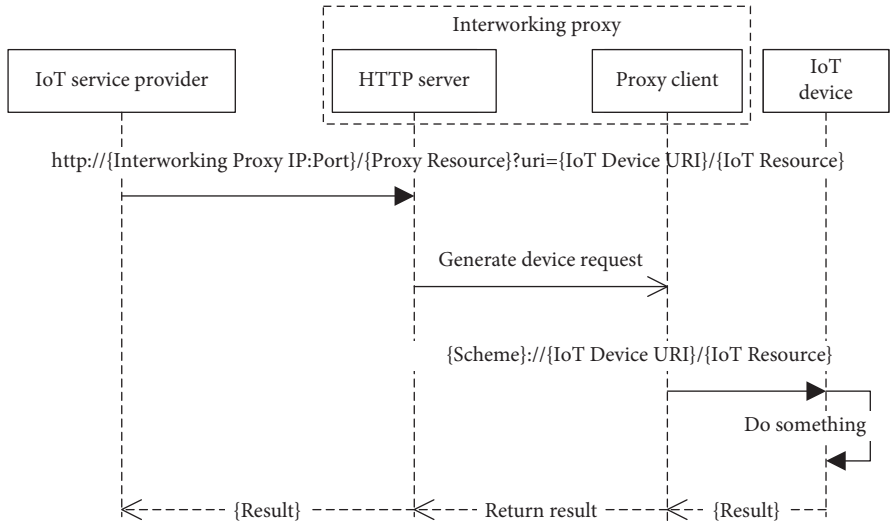

Figure 10: Request message translation sequence based on interworking proxy.

Once the request is delivered to the interworking proxy, the proxy interprets the URI and gets the information to create the corresponding request as shown in Figure 16(b). The destination URI and other variables are delivered through the parameters of the request. The value of "uri" is the URI to request the IoT device with the PUT method and parameter "level" and its value "1." The request accesses the resource/led of the IoT device that returns {"result":"1"} in this experiment. Figure 16(c) shows the loggings in the IoT device that presents the details of the request. Therefore,

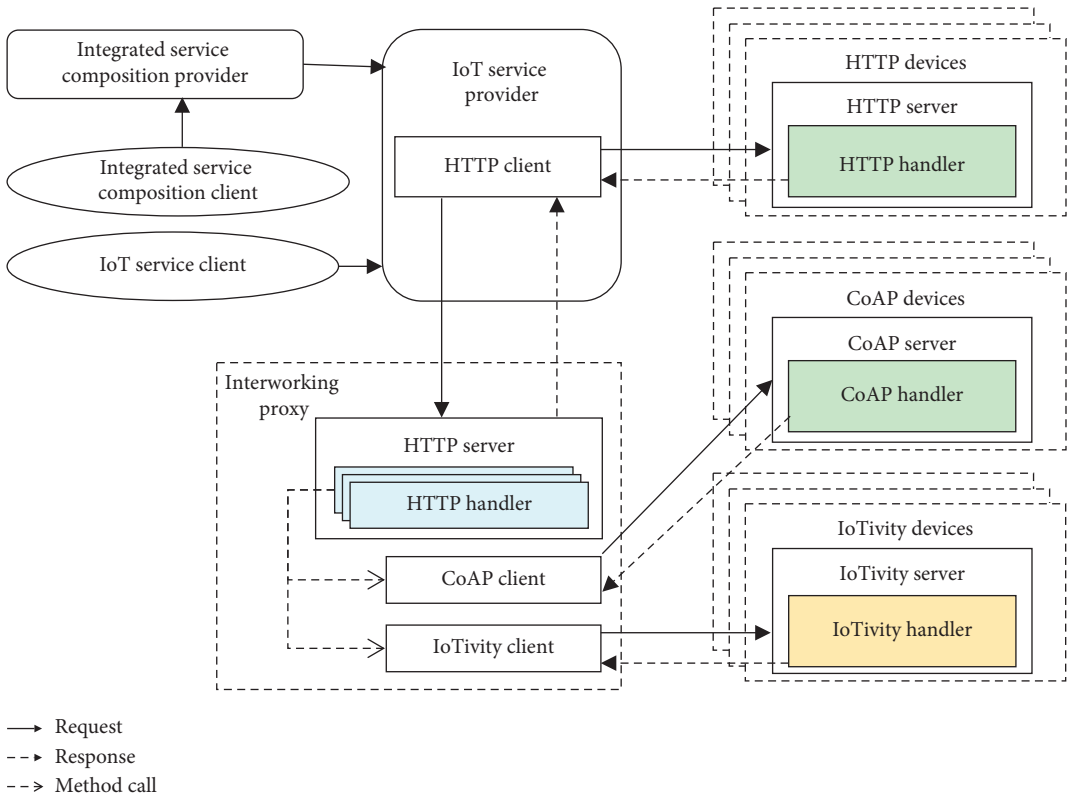

Figure 11: Proposed transparent IoT device access based on interworking proxy.

Table 1: Development environment.

Entity	Platform	Frameworks	Tool
Service providers	Windows 10 Pro 64 bit	Spring MVC framework 4.3.6.RELEASE, MyBatis	Spring Tool Suite 3.8.1.RELEASE, MySQL Workbench
Interworking proxy	Intel Edison Board with Android Things 0.2	IoTivity 1.2/Californium CoAP, Jetty 9.1	Android Studio 2.3
IoT device	Intel Edison Board with Android Things 0.2	IoTivity 1.2/Californium CoAP/Jetty 9.1	Android Studio 2.3

based on the interworking proxy, transparent access is enabled from the integrated service to the heterogeneous IoT devices.

For enabling transparent access to the heterogeneous IoT networks in the proposed integrated service composition approach, the interworking proxy translates the request to the destination protocols. Most of the IoT networks are configured for a constrained environment. The IoT devices are developed using the small size of machine that requires a limited power supply and computational ability. Therefore, the network packets shall be smaller. We experiment with the proposed IoT system using the IoT networks based on HTPP and CoAP to evaluate the performances.

Figure 17 presents the latency of requests to the IoT devices that are developed using HTTP and CoAP. The request is sent by the integrated service to the IoT devices. The Round-Trip Time (RTT) is the approach to collect the latency of requests. For the performances, the RTTs are collected for the requests from the IoT service provider to the IoT devices. For accessing the HTTP-based IoT network, the request can be directly delivered to the IoT device without the protocol translation by the interworking proxy. For accessing the CoAP-based IoT network, the request needs to be delivered by the interworking proxy through the protocol translation. As expected, the average latency of accessing the HTTP-based IoT device is smaller than the CoAP-based accessing. Nevertheless, the result can be referred to develop the proxy-based IoT system.

However, the packet size in the CoAP-based IoT network is significantly smaller than the HTTP-based IoT network as shown in Figure 18. The integrated service sends the same request message to the IoT devices. For the CoAP-based IoT device, the packet size is reduced due to the structure of the protocol that enables the communications in the IoT networks using small packets. Also, most IoT devices are developed for constrained environments using constrained

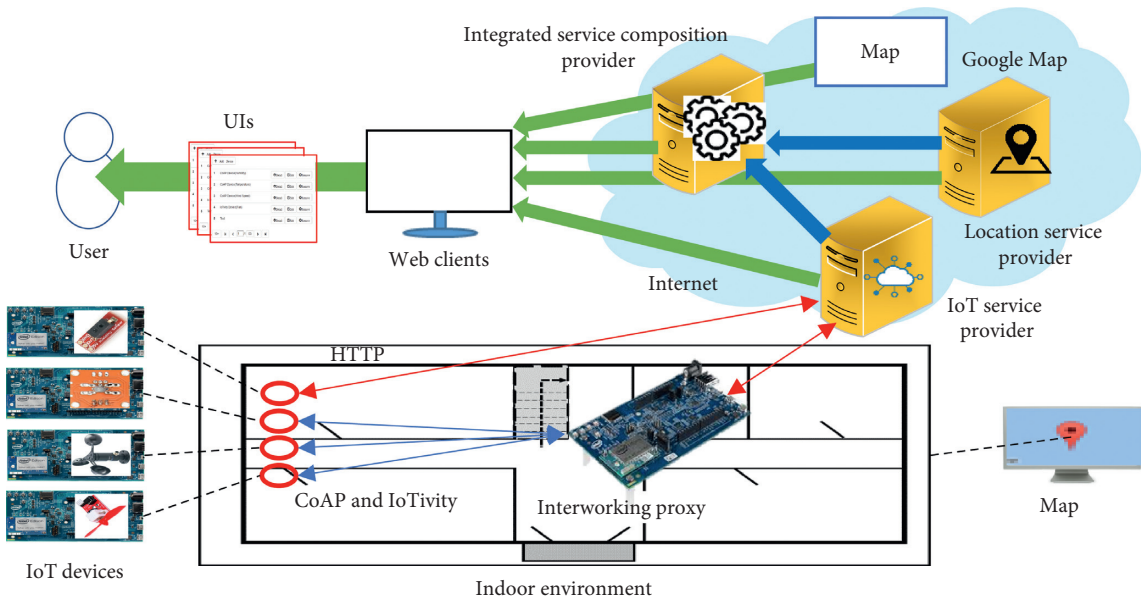

Figure 12: Proposed IoT scenario implementation for integrated service composition based on IoT and location service providers.

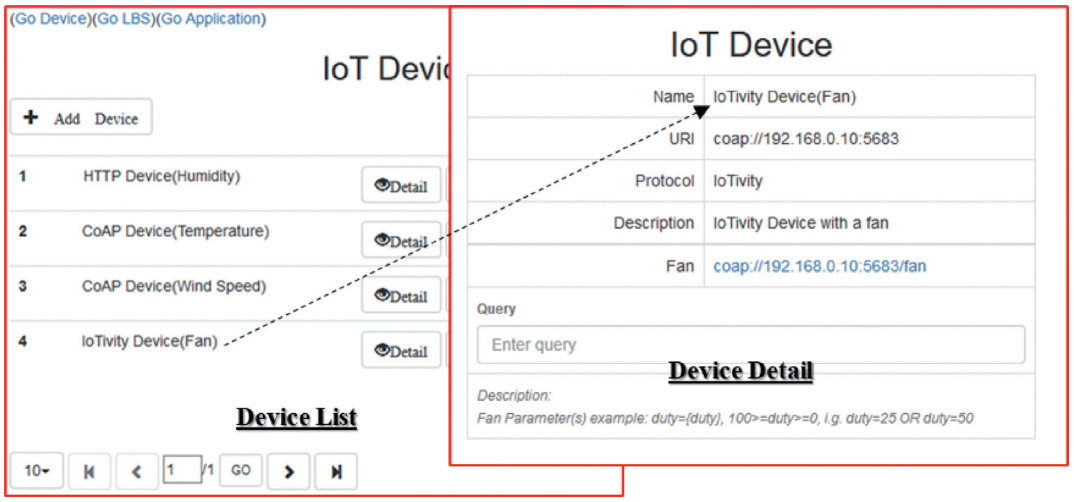

Figure 13: Implementation result of IoT service provider.

solutions such as CoAP. Therefore, the interworking proxy enables the integrated service in the Internet to access the IoT device by converting the request message for the constrained environment.

6. Comparison and Significance

For emphasizing the significance of the proposed IoT architecture, the comparisons with the existing solutions are presented based on the brief introductions of IoTivity, OCEAN, LWM2M, ThingSpeak, and KAA that are comprehensive IoT solutions to provide various features and extensions. Table 2 presents the advantages and weaknesses of the proposed IoT architecture by comparing the main characteristics with other IoT platforms and frameworks that are introduced in the section of related works. For providing the proposed integrated service composition, the REST API architecture enables the service providers to provide services by representing the data in the web. The service providers provide REST APIs for various services by delivering the data to the client-side including other service providers [71]. For providing transparent access by the integrated service, the interworking proxy can be deployed in the entry of IoT networks to translate the network packets for the destination protocols.

KAA is an IoT middleware platform to develop smart IoT solutions. The platform provides management solutions to the IoT device network for delivering the data to the cloud. For accessing the IoT devices through KAA, MQTT is provided which enables communication based on subscription and notification. Also, the visualization service is provided by the platform. KAA provides the services

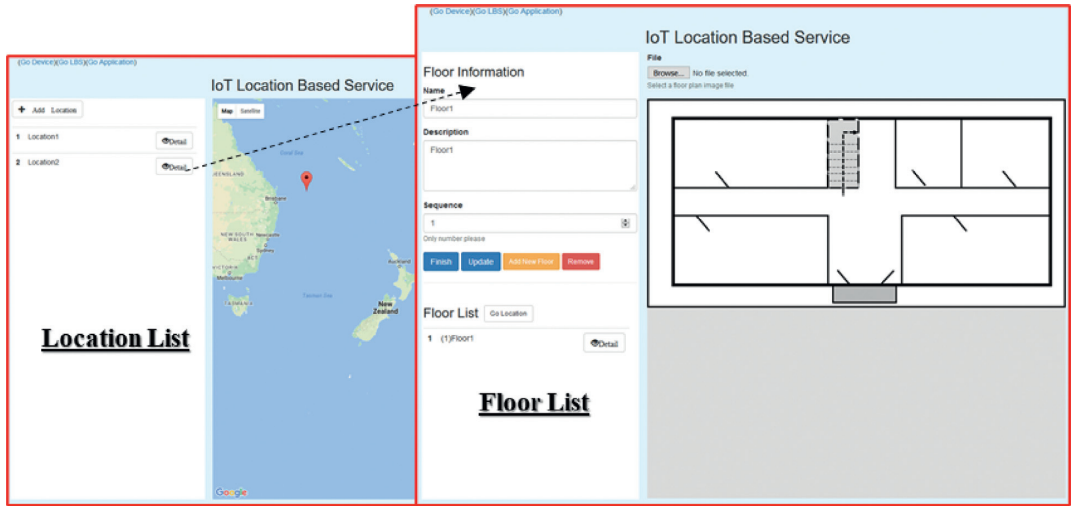

Figure 14: Implementation result of location service provider.

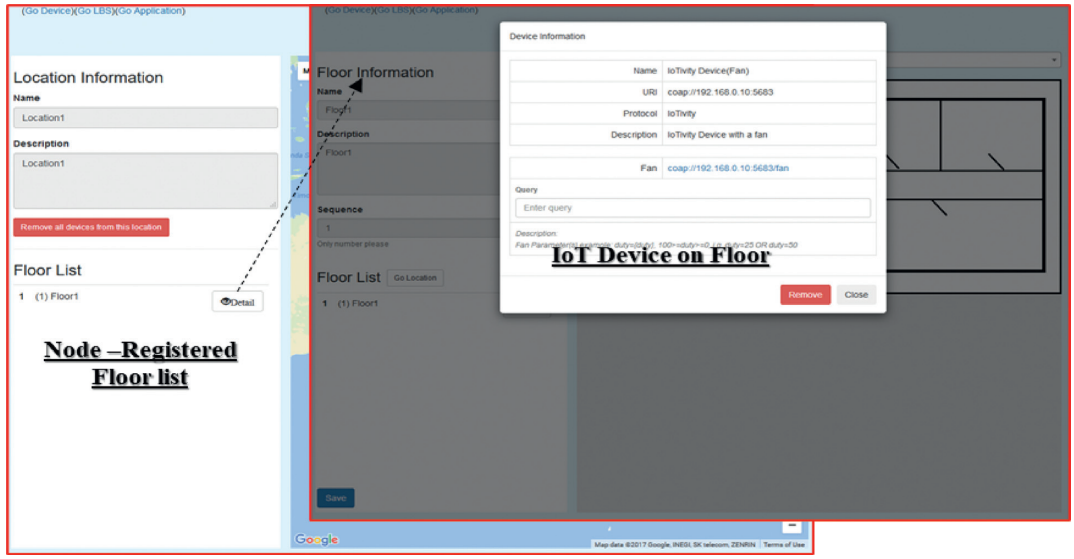

Figure 15: Implementation result of integrated service composition provider.

through the REST APIs that enable the integration with other services in the cloud. However, for the heterogeneous IoT devices, the interworking mechanism is not provided. ThingSpeak is an IoT platform that provides comprehensive services including management, visualization, and integration with various third-party platforms such as Twilio, Twitter, ThingHTTP, and MATLAB. The integration approach is enabled by the REST APIs that are provided by the cloud server. The platform supports a less number of device connectivity simultaneously [72]. Also, the heterogeneity of IoT networks is not supported due to the communication architecture.

For providing the IoT standard architectures, several specifications are proposed such as OMA, OCF, and oneM2M. The OMA proposes the IoT management architecture for providing services to manage IoT devices. The services are developed based on the request and response transaction model through the client and server. The implementation of the OMA specification is LWM2M that provides services using CoAP for constrained devices. The CoAP-based architecture enables the services which can be accessed through the REST APIs. Also, the CoAP extensions enable the interworking with other protocols through the proxy [73]. However, the LWM2M addresses to provide services for the constrained devices. Therefore, the visualization function is not necessary. The IoTivity is the implementation of OCF that includes the functionality of messaging, discovery, monitoring, and maintenance based on fundamental communication ability. Nevertheless, the visualization function is not provided due to provide services for constrained devices. Different from OMA and OCF, the oneM2M addresses to large-scale industrial solutions for logistics, factories, and cities. The OCEAN is the implementation that is comprised of Mobius and CUBE. The

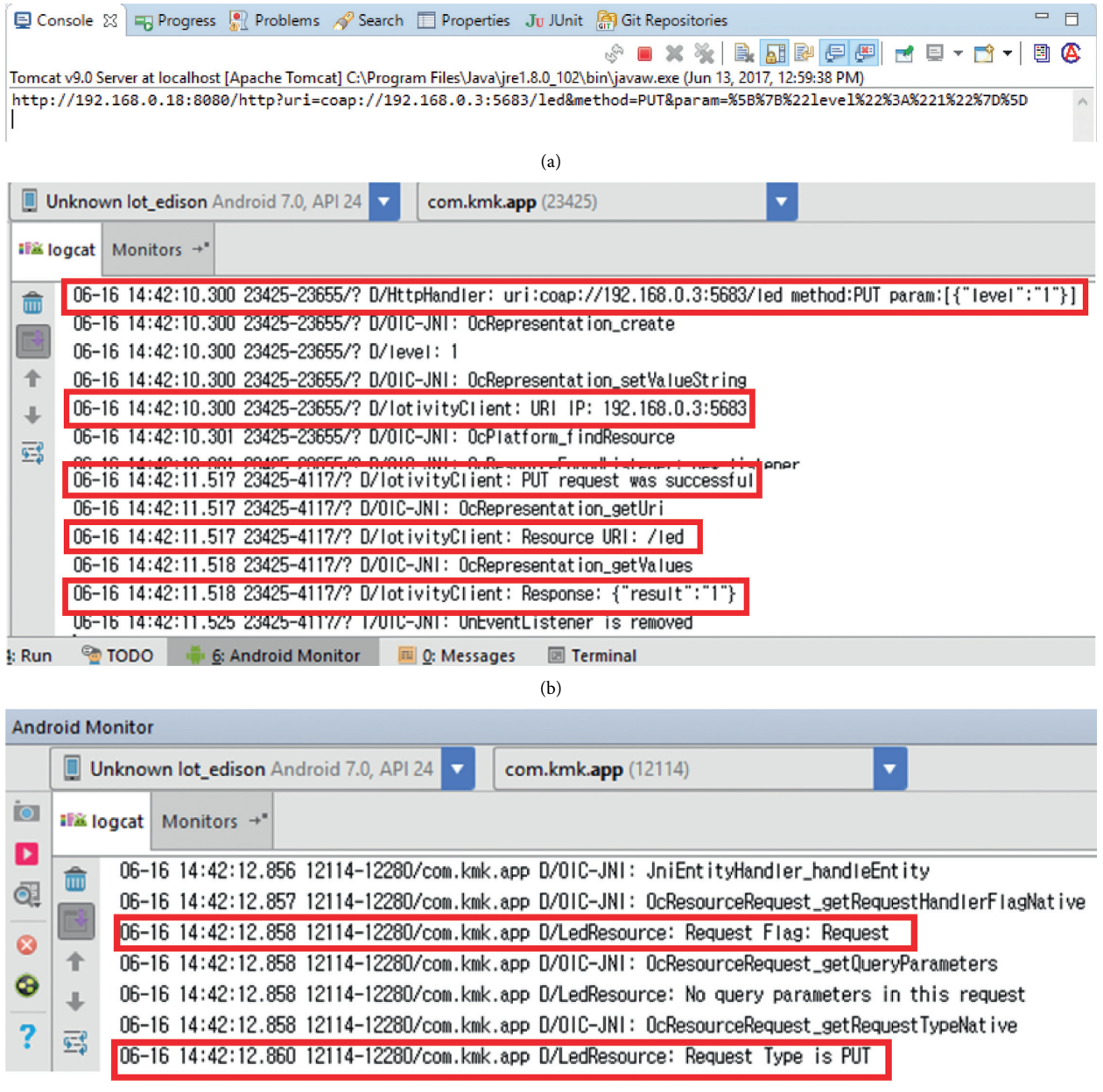

Figure 16: Logging result of accessing IoT device through interworking proxy. (a) Logging in IoT service provider. (b) Logging in interworking proxy. (c) Logging in IoT device.

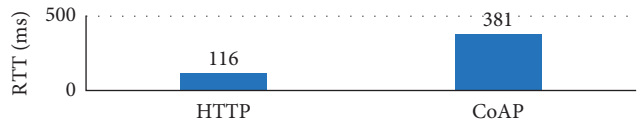

Figure 17: IoT device request latency comparison based on HTTP and CoAP.

Mobius can be deployed in the cloud server to provide management and UI. The MQTT is the main protocol to communicate with the IoT devices that is implemented using CUBE. The services of Mobius are provided through REST APIs that enable the management services to be accessed by other service clients for integrating with other services to provide improved services. The oneM2M considers the heterogeneity management based on the interworking proxy

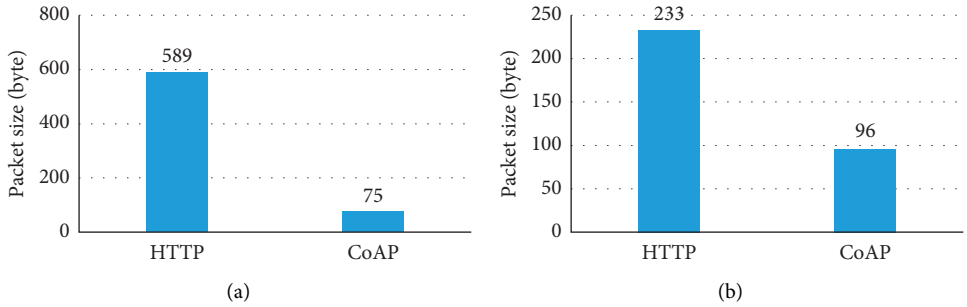

Figure 18: IoT device request packet size comparison. (a) Request message. (b) Response message.

Table 2: Comparison of IoT implementations.

Characteristics	Proposed IoT architecture	IoTivity	OCEAN	LWM2M	ThingSpeak	KAA
Device management	Yes	Yes	Yes	Yes	Yes	Yes
Protocol support	HTTP, CoAP	CoAP	MQTT	CoAP	HTTP, MQTT	MQTT
Visualization	Yes	No	Yes	No	Yes	Yes
Integration	REST API	REST API	REST API	REST API	REST API	REST API
Heterogeneity management	Yes	Yes	No	Yes	No	No

entity to handle the interworking between the oneM2M system and external systems. However, the interworking proxy is used for bridging the other systems to the oneM2M.

Compared with other platforms and frameworks, the proposed IoT architecture includes all the proposed characteristics that provide the integrated service composition that is enabled by the REST API service architecture. Moreover, the transparent access to the heterogeneous IoT networks is supported by deploying the interworking proxy in the entry of IoT networks.

7. Conclusions and Future Directions

We proposed a service composition approach in the IoT architecture based on multiple service providers including IoT service, location service, and integrated service composition providers to integrate the IoT services to the location information for providing improved IoT services. Moreover, the interworking proxy is deployed in the entry of IoT networks to translate the request and response messages which enables transparent access to heterogeneous IoT networks for integrated services. Each provider provides web services through REST APIs that deliver information to the clients as well as other providers to integrate the information for providing improved services. For integrating IoT service to location-based service, the APIs of IoT and location service providers deliver the information of IoT devices and location to the integrated service composition provider. Then, based on the APIs of the integrated service composition provider, the registered IoT devices are integrated into the map and represented to the Internet. The interworking proxy provides the interface to the Internet for bridging the HTTP clients to the constrained IoT devices. Therefore, the size of network packets is reduced in the IoT network which can support the device to be in low-energy communication. According to the performance of the proposed architecture, the packet size is greatly reduced to compare through the interworking proxy to access the IoT network. However, the latency is increased compared with direct access through HTTP. Nevertheless, the proposed IoT architecture enables the service clients do not need to consider the heterogeneity of IoT networks while using the integrated services.

As future directions, we will develop an improved interworking proxy to bridge multiple communication solutions such as protocols based on BLE and WiFi. The implemented platform provides BLE communication based on IoTivity. Therefore, the interworking proxy can be upgraded to deliver messages to the IoT device through BLE. Moreover, we will integrate an intelligent service provider to the proposed IoT architecture for providing intelligent services based on IoT devices. The deep learning models can be trained with sensing and actuating data to derive the inference models that are used for providing intelligent services in the intelligent provider. The proposed IoT architecture can adopt intelligent services to operate the IoT devices.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Wenquan Jin, Rongxu Xu, Sunhwan Lim, Dong-Hwan Park, Chanwon Park, and Dohyeun Kim designed the overall system. Wenquan Jin implemented the overall system and performed experiments. Wenquan Jin and Dohyeun Kim wrote this paper.

Acknowledgments

This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean government under Grant NRF-2019R1I1A1A01062456 and in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2020-0-00048, Development of 5G-IoT Trustworthy AI-Data Commons Framework, 50%).

References

[1] Y. Hajjaji, W. Boulila, I. R. Farah, I. Romdhani, and A. Hussain, "Big data and IoT-based applications in smart environments: a systematic review," *Computer Science Review*, vol. 39, Article ID 100318, 2021.

[2] W. Jin, R. Xu, S. Lim, D.-H. Park, C. Park, and D. Kim, "Dynamic inference approach based on rules engine in intelligent edge computing for building environment control," *Sensors*, vol. 21, no. 2, p. 630, 2021.

[3] V. Kamath, J. Morgan, and M. I. Ali, "Industrial IoT and Digital Twins for a Smart Factory: an open source toolkit for application design and benchmarking," in *Proceedings of the 2020 Global Internet of Things Summit (GIoTS)*, pp. 1–6, Dublin, Ireland, June 2020.

[4] I. Hedi, I. Špeh, and A. Šarabok, "Iot network protocols comparison for the purpose of iot constrained networks," in *Proceedings of the 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 501–505, Opatija, Croatia, May 2017.

[5] W. Jin and D. Kim, "A sleep-awake scheme based on CoAP for energy-efficiency in internet of things," *International Journal on Informatics Visualization*, vol. 1, no. 4, pp. 110–114, 2017.

[6] Z. Zhai, K. Xiang, L. Zhao, B. Cheng, J. Qian, and J. Wu, "IoT-RECSM-resource-constrained smart service migration framework for IoT edge computing environment," *Sensors*, vol. 20, no. 8, p. 2294, 2020.

[7] B. B. Gupta and Q. Megha, "An overview of Internet of Things (IoT): architectural aspects, challenges, and protocols," *Concurrency and Computation: Practice and Experience*, vol. 32, Article ID e4946, 21 pages, 2020.

[8] V. Adat and B. B. Gupta, "Security in Internet of Things: issues, challenges, taxonomy, and architecture," *Telecommunication Systems*, vol. 67, no. 3, pp. 423–441, 2018.

[9] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the soa-based internet of things: discovery, query, selection, and on-demand provisioning of web services," *IEEE Transactions on Services Computing*, vol. 3, no. 3, pp. 223–235, 2010.

[10] C. Pautasso, "Restful web services: principles, patterns, emerging technologies," in *Web Services Foundations*-Springer, Berlin, Germany, 2014.

[11] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: a survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[12] D. Guinard, V. Trifa, and E. Wilde, "A resource oriented architecture for the web of things," in *Proceedings of the 2010 Internet of Things (IOT)*, pp. 1–8, Beijing, China, June 2010.

[13] P. Sethi and S. R. Sarangi, "Internet of things: architectures, protocols, and applications," *Journal of Electrical and Computer Engineering*, vol. 2017, Article ID 9324035, 2017.

[14] S. Balaji, K. Nathani, and R. Santhakumar, "Iot technology, applications and challenges: a contemporary survey," *Wireless Personal Communications*, vol. 108, no. 1, pp. 363–388, 2019.

[15] H. Hejazi, H. Rajab, T. Cinkler, and L. Lengyel, "Survey of platforms for massive iot," in *Proceedings of the 2018 IEEE International Conference on Future IoT Technologies*, pp. 1–8, Eger, Hungary, January 2018.

[16] S. Sinche, D. Raposo, N. Armando et al., "A survey of iot management protocols and frameworks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1168–1190, 2019.

[17] W. Jin and D. Kim, "Resource management based on ocf for device self-registration and status detection in iot networks," *Electronics*, vol. 8, no. 3, p. 311, 2019.

[18] J. Wu, I. Bisio, C. Gniady, E. Hossain, M. Valla, and H. Li, "Context-aware networking and communications: Part 1 [guest editorial]," *IEEE Communications Magazine*, vol. 52, no. 6, pp. 14-15, 2014.

[19] W. Jin and D.-H. Kim, "Iot device management architecture based on proxy," in *Proceedings of the 2017 6th International Conference on Computer Science and Network Technology (ICCSNT)*, pp. 84–87, Dalian, China, October 2017.

[20] W. Jin and D. Kim, "A sleep scheme based on mq broker using subscribe/publish in iot network," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 8, no. 2, pp. 539–545, 2018.

[21] W. Jin and D. Kim, "Development of virtual resource based iot proxy for bridging heterogeneous web services in iot networks," *Sensors*, vol. 18, no. 6, p. 1721, 2018.

[22] W. Jin and D. Kim, "Interworking proxy based on ocf for connecting web services and iot networks," *Journal of Communications*, vol. 15, no. 2, 2020.

[23] P. Asghari, A. M. Rahmani, and H. H. S. Javadi, "Service composition approaches in iot: a systematic review," *Journal of Network and Computer Applications*, vol. 120, pp. 61–77, 2018.

[24] M. Hamzei and N. Jafari Navimipour, "Toward efficient service composition techniques in the internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3774–3787, 2018.

[25] S. N. Han, I. Khan, G. M. Lee, N. Crespi, and R. H. Glitho, "Service composition for ip smart object using realtime web protocols: concept and research challenges," *Computer Standards & Interfaces*, vol. 43, pp. 79–90, 2016.

[26] D. Arellanes and K.-K. Lau, "Evaluating iot service composition mechanisms for the scalability of iot systems," *Future Generation Computer Systems*, vol. 6, no. 1, 2020.

[27] S. Krco, B. Pokri, and F. Carrez, "Designing iot architecture (s): a ′ european perspective," in *Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT)*, pp. 79–84, Seoul, Korea, March 2014.

[28] F. Carrez, "Iot-a deliverable D1. 5–final architectural reference model for the Iot V3. 0," 2017.

[29] F. J. Lin, Y. Ren, and E. Cerritos, "A feasibility study on developing iot/m2m applications over etsi m2m architecture," in *Proceedings of the 2013 International Conference on Parallel and Distributed Systems*, pp. 558–563, Seoul, Korea, December 2013.

[30] L. A. Grieco, M. B. Alaya, T. Monteil, and K. Drira, "Architecting information centric etsi-m2m systems," in *Proceedings of the 2014 IEEE International Conference on Pervasive Computing and Communication Workshops*

(PERCOM WORKSHOPS), pp. 211–214, Budapest, Hungary, March 2014.

[31] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A survey on end-edgecloud orchestrated network computing paradigms: transparent computing, mobile edge computing, fog computing, and cloudlet," *ACM Computing Surveys (CSUR)*, vol. 52, no. 6, pp. 1–36, 2019.

[32] J. Lloret, J. Tomas, A. Canovas, and L. Parra, "An integrated iot architecture for smart metering," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 50–57, 2016.

[33] P. Desai, A. Sheth, and P. Anantharam, "Semantic gateway as a service architecture for iot interoperability," in *Proceedings of the 2015 IEEE International Conference on Mobile Services*, pp. 313–319, New York, NY, USA, May 2015.

[34] S. K. Datta, C. Bonnet, and N. Nikaein, "An iot gateway centric architecture to provide novel m2m services," in *Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT)*, pp. 514–519, Seoul, Korea, March 2014.

[35] N. Armando, J. Fernandes, S. Sinche, D. Raposo, J. S. Silva, and F. Boavida, "A unified solution for iot device management," in *Proceedings of the 2019 22nd International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pp. 1–6, Lisbon, Portugal, November 2019.

[36] K. Košťál, P. Helebrandt, M. Belluš, M. Ries, and I. Kotuliak, "Management and monitoring of iot devices using blockchain," *Sensors*, vol. 19, no. 4, p. 856, 2019.

[37] B. Agyemang, Y. Xu, N. Sulemana, and M. Liu, "Resource-oriented architecture toward efficient device management and service enablement," in *Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2561–2566, Banff, AB, Canada, October 2017.

[38] S. Sinche, J. S. Silva, D. Raposo, A. Rodrigues, V. Pereira, and F. Boavida, "Towards effective iot management," in *Proceedings of the 2018 IEEE SENSORS*, pp. 1–4, New Delhi, India, October 2018.

[39] P. Hu, H. Ning, L. Chen, and M. Daneshmand, "An open internet of things system architecture based on software-defined device," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2583–2592, 2018.

[40] C. L. Stergiou, K. E. Psannis, and B. B. Gupta., "IoT-based big data secure management in the fog over a 6G wireless network," *IEEE Internet of Things Journal*, vol. 31, 2020.

[41] C.-M. Kim, S.-I. Choi, and S.-J. Koh, "Idmp-vlc: iot device management protocol in visible light communication networks," in *Proceedings of the 2017 19th Inter national Conference on Advanced Communication Technology (ICACT)*, pp. 578–583, Pyeongchang, South Korea, March 2017.

[42] M. Maloney, E. Reilly, M. Siegel, and G. Falco, "Cyber physical iot device management using a lightweight agent," in *Proceedings of the 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 1009–1014, Rhodes Island, Greece, November 2019.

[43] S. Bera, S. Misra, S. K. Roy, and M. S. Obaidat, "Soft-wsn: softwaredefined wsn management system for iot applications," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2074–2081, 2016.

[44] C. Esposito, M. Ficco, and B. B. Gupta, "Blockchain-based authentication and authorization for smart city applications," *Information Processing & Management*, vol. 58, no. 2, Article ID 102468, 2021.

[45] J. Ferreira, J. N. Soares, R. Jardim-Goncalves, and C. Agostinho, "Management of Iot devices in a physical network," in *Proceedings of the 2017 21st International Conference on Control Systems and Computer Science (CSCS)*, pp. 485–492, Bucharest, Romania, May 2017.

[46] H. Park, H. Kim, H. Joo, and J. Song, "Recent advancements in the internet-of-things related standards: a onem2m perspective," *Ict Express*, vol. 2, no. 3, pp. 126–129, 2016.

[47] J. Swetina, G. Lu, P. Jacobs, F. Ennesser, and J. Song, "Toward a standardized common M2M service layer platform: introduction to oneM2M," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 20–26, 2014.

[48] V. C. Andrianto, J. Lam, R. N. Soenjoto Widodo, S.-G. Lee, H.-J. Lee, and H.-T. Lim, "Toward implementation of onem2m based iot platform," *Journal of Theoretical & Applied Information Technology*, vol. 96, no. 2, 2018.

[49] J. Hwang, J. An, A. Aziz, J. Kim, S. Jeong, and J. Song, "Interworking models of smart city with heterogeneous internet of things standards," *IEEE Communications Magazine*, vol. 57, no. 6, pp. 74–79, 2019.

[50] M. A. G. Maureira, D. Oldenhof, and L. Teernstra, "Thingspeak–an api and web service for the internet of things," 2011.

[51] S. Pasha, "Thingspeak based sensing and monitoring system for iot with matlab analysis," *International Journal of New Technology and Research*, vol. 2, no. 6, 2016.

[52] M. U. H. Al Rasyid, M. H. Mubarrok, and J. A. Nur Hasim, "Implementation of environmental monitoring based on kaa iot platform," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 6, pp. 2578–2587, 2020.

[53] "Oma," http://www.openmobilealliance.org.

[54] S. Rao, D. Chendanda, C. Deshpande, and V. Lakkundi, "Implementing lwm2m in constrained iot devices," in *Proceedings of the 2015 IEEE Conference on Wireless Sensors (ICWiSe)*, pp. 52–57, Melaka, Malaysia, August 2015.

[55] C. A. L. Putera and F. J. Lin, "Incorporating oma lightweight m2m protocol in iot/m2m standard architecture," in *Proceedings of the 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pp. 559–564, Milan, Italy, December 2015.

[56] "Ocf," https://openconnectivity.org/.

[57] S. Park, "A new open iot consortium," in *Proceedings of the 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 356–359, Taipei, Taiwan, March 2017.

[58] J.-C. Lee, J.-H. Jeon, and S.-H. Kim, "Design and implementation of healthcare resource model on iotivity platform," in *Proceedings of the 2016 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 887–891, Jeju Island, Korea, May 2016.

[59] "Iotivity," https://iotivity.org.

[60] M. Iglesias-Urkia, A. Orive, and A. Urbieta, "Analysis of CoAP implementations for industrial internet of things: a survey," *Procedia Computer Science*, vol. 109, pp. 188–195, 2017.

[61] A. I. A. Ahmed, A. Gani, S. H. Ab Hamid et al., "Service management for iot: requirements, taxonomy, recent advances and open research challenges," *IEEE Access*, vol. 7, pp. 472–155, 2019.

[62] S. Berrani, A. Yachir, B. Djemaa, and M. Aissani, "Extended multiagent system based service composition in the internet of things," in *Proceedings of the 2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, pp. 1–8, Tebessa, Algeria, May 2010.

[63] C. Akasiadis, G. Tzortzis, E. Spyrou, and C. Spyropoulos, "Developing complex services in an iot ecosystem," in *in Proceedings of the 2015 IEEE 2nd World Forum on Internet of*

Things (WF-IoT), pp. 52–56, IEEE, Milan, Italy, December 2015.

[64] G. Pierris, D. Kothris, E. Spyrou, and C. Spyropoulos, "An enabling platform for the current internet of things ecosystem," in *Proceedings of the 19th Panhellenic Conference on Informatics*, pp. 438–444, Athens, Greece, October 2015.

[65] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J.-S. Oh, "Semisupervised deep reinforcement learning in support of iot and smart city services," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 624–635, 2017.

[66] M. Sun, Z. Shi, S. Chen, Z. Zhou, and Y. Duan, "Energy-efficient composition of configurable internet of things services," *IEEE Access*, vol. 5, pp. 609–625, 2017.

[67] K. Wang, Y. Wang, Y. Sun, S. Guo, and J. Wu, "Green industrial Internet of Things architecture: an energy-efficient perspective," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 48–54, 2016.

[68] K. Sangsanit, W. Kurutach, and S. Phoomvuthisarn, "REST web service composition: a survey of automation and techniques," in *Proceedings of the 2018 International Conference on Information Networking (ICOIN)*, pp. 116–121, Busan, Korea, February 2018.

[69] F. Haupt, M. Fischer, D. Karastoyanova, L. Frank, and K. Vukojevic-Haupt, "Service composition for REST," in *Proceedings of the 2014 IEEE 18th International Enterprise Distributed Object Computing Conference*, pp. 110–119, Ulm, Germany, September 2014.

[70] F. Dai, Q. Mo, Z. Qiang, B. Huang, W. Kou, and H. Yang, "A choreography analysis approach for microservice composition in cyber-physical-social systems," *IEEE Access*, vol. 8, pp. 53215–53222, 2020.

[71] W. Jin, R. Xu, T. You, Y.-G. Hong, and D. Kim, "Secure edge computing management based on independent microservices providers for gateway-centric IoT networks," *IEEE Access*, vol. 8, pp. 187975–187990, 2020.

[72] P. P. Ray, "A survey of iot cloud platforms," *Future Computing and Informatics Journal*, vol. 1, no. 1-2, pp. 35–46, 2016.

[73] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (CoAP)," 2014.