Proposal and Evaluation of Automatic Registration Method of Service Information by Distribution from Deployment System to Maintenance System

Ryota Katayanagi
NTT Network Service Systems Laboratories
NTT Corporation
Tokyo, Japan
ryota.katayanagi.wt@hco.ntt.co.jp

Kensuke Takahashi
NTT Network Service Systems Laboratories
NTT Corporation
Tokyo, Japan
kensuke.takahashi.gm@hco.ntt.co.jp

Satoshi Kondoh
NTT Network Service Systems Laboratories
NTT Corporation
Tokyo, Japan
satoshi.kondou.hm@hco.ntt.co.jp

***Abstract*— With the popularization of B2B2X (Business to Business to X, where "X" represents any end user) business, the software for supporting both deployment and maintenance of collaborative services, which combine multiple services, is required. We previously investigated technology for enabling both deployment and maintenance of collaborative services in one stop and achieved automation in every process such as deployment and maintenance. However, to complete a series of processes in one stop without operators, it is necessary to automatically distribute service information (configuration of systems that provide collaborative services) between processes. In this study, we developed a method for automatically distributing service information between a deployment system and a maintenance system, and we evaluated its feasibility. For the evaluation, we implemented the proposed method with current technologies and applied it to the service-maintenance process in a virtual environment.**

Keywords—Automatic Registration, Network Management, Service Discovery

I. INTRODUCTION

Information Technology Infrastructure Library (ITIL) [1] makes the purpose of the service operations to be the service offering at the level agreed with the customer. In a service using a network and cloud, monitoring, analysis, and restoration from failure are done from decisions on the basis of knowledge of the operator and use of the system and tools.

However, services combining network functions, such as service function chaining (SFC) [2] and network functions virtualization (NFV) [3], and business models, such as Business to Business to X (B2B2X, where "X" represents any end user) have spread, and the management process of services have become complicated (Fig. 1).

With the spread of B2B2X business, there is a growing demand for software that supports operations to provide a collaborative service that combines multiple services. We previously proposed orchestration technology that reduces manual operations by building and setting up collaborative services in a one-stop manner [4] [5]. We have also proposed a workflow-less autonomous system architecture in which maintenance work is expressed in microservices (called workers) that autonomously perform maintenance work to reduce manual operations in collaborative-service maintenance work [6] [7] [8] [9] [10].

Through these studies, the technology for supporting deployment and maintenance operations was developed to reduce manual operations, but to achieve a series of operations from deployment to maintenance in one stop, it is necessary for the maintenance system to refers to service information, which is the configuration of systems that provide a collaborative service.

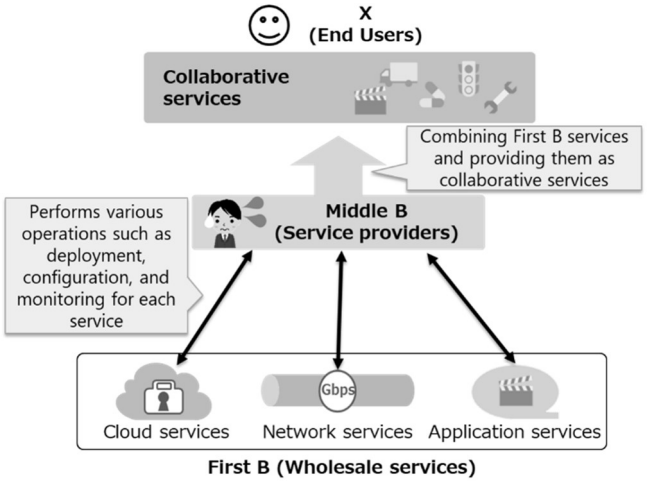

Fig. 1 Burden of Middle B with B2B2X model

In Section 2, we describe our orchestration technology and workflow-less autonomous system architecture, and in Section 3, we describe the challenges to achieving one-stop operation of a collaborative service by using current technologies. In Section 4, we present our proposed automatic distribution method with which a deployment system with our orchestration technology cooperates with a maintenance system with our workflow-less autonomous system architecture to automatically distribute service information of deployed collaborative services. In Section 5, we present the results of an implementation and evaluation of the proposed method. Finally, we conclude the paper and mention future work in Section 6.

II. CONVENTIONAL WORK

In this section, we describe the technology for supporting each process for one-stop operation.

A. Orchestration technology

In B2B2X business, service providers (Middle B) provide collaborative services to end users (X). To provide collaborative services, it is necessary to carry out deployment

©Copyright IEICE - APNOMS 2021

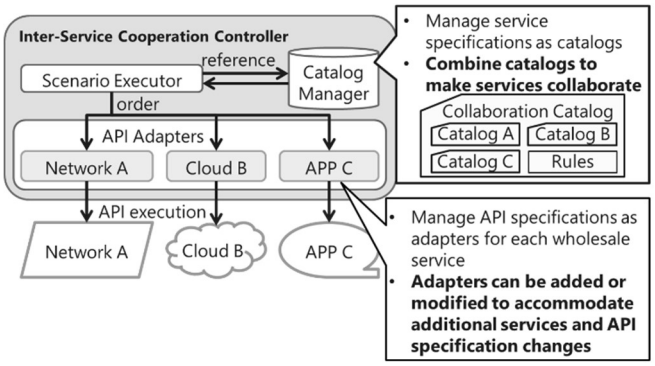

Fig. 2 Overview of orchestration technology

operations, such as application, construction, and configuration of each service provided by the wholesale service provider (First B). As a result, many operations are necessary, making it difficult to enter the business.

We previously proposed orchestration technology for enabling the construction and configure a collaborative service in one stop by defining and managing the configuration and construction procedure of the service as catalogs and the service API execution function as adapters to support the construction of collaborative services.

Fig. 2 shows a system with our orchestration technology. The specifications of each wholesale service are defined as a catalog, and the specifications of the API are defined as an adapter. To construct a collaborative service, it is necessary to create a collaboration catalog with collaboration rules, which defines the execution order of the API and correlation of parameters among catalogs. Using these catalogs and adapters, the collaborative service can be constructed and configured in one stop.

B. *Workflow-less autonomous system architecture*

Since a collaborative service is composed of multiple wholesale services, it is complicated as a maintenance object.

To reduce manual operations, workflow engines, such as StackStorm [11] and Jenkins [12], have been proposed to automate operations. However, verification of the entire workflow is required when the service specifications and equipment change. Since the verification requires much time, it is difficult to make the workflow follow the changes in service specifications.

In our workflow-less autonomous system architecture, the functional components of an operation are defined as workers. Workers autonomously cooperate, judge, and act. By processing a series of operations due to cooperation among workers, defining the workflow is not unnecessary, and it becomes possible to flexibly handle the changes in the situation such as those in service specifications and equipment.

Fig. 3 shows the concept of our workflow-less autonomous system architecture. Maintenance functions, such as information collection, information processing, information analysis, testing, and configuration change, are defined as workers. Information required for operations is shared among workers via a message bus. When an input (message) arrives from another worker, the worker autonomously performs an operation on the maintenance object on the basis of the judgment function implemented in the worker. For addition, change and deletion of an operation

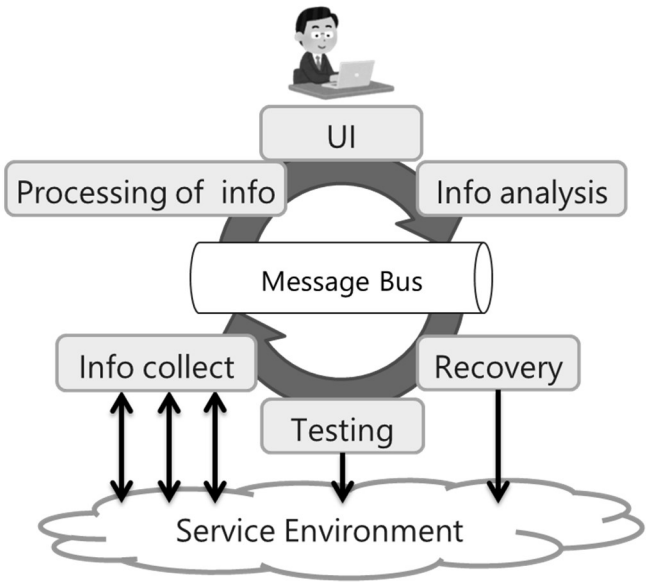

Fig. 3 Concept of workflow-less autonomous system architecture

along with changes in service specifications and equipment can be done without affecting other workers only if new creation of the worker or improvement of the existing worker.

III. CHALLENGES FOR ONE-STOP OPERATION

In this section, we present the challenges in enabling one-stop operation from deployment to maintenance start and the solution to these challenges.

A. *Challenges*

By applying our technology and architecture introduced in Section 2, manual operations of each process of deployment and maintenance can be reduced, but the maintenance system needs to refer to the service information of the maintenance object without operator intervention before maintenance start to enable one-stop operation.

In B2B2X business, which offers collaborative services, Middle B deploys and configures the wholesale service in accordance with the specifications of each wholesale service. Since the deployment system to be used also depends on each wholesale service, the propriety of automatic distribution of service information to the maintenance system when establishment is completed depends on the service specifications. To distribute the service information of a collaborative service to the maintenance system automatically, it must be assumed that operator intervention is required, which is problematic for enabling one-stop operation (Fig. 4).

B. *Requirements for obtaining solution*

To enable one-stop operation from deployment to maintenance start, it is necessary to automatically distribute service information from the deployment system to maintenance system.

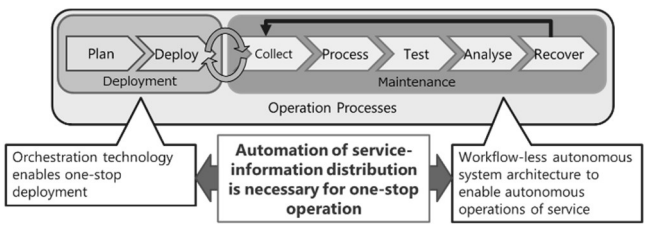

Fig. 4 Challenges to enabling one-stop operation

©Copyright IEICE - APNOMS 2021

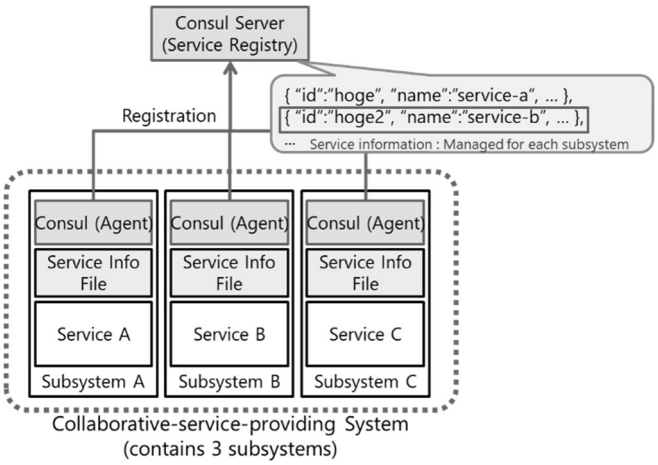

Fig. 5 Service discovery pattern with Consul

To automatically collect service information from the service environment, there is a service discovery pattern [13] in the microservice architecture. It is possible to distribute the service information collected from the collaborative service to the maintenance system if software applied to the service discovery pattern, such as Consul [14], is used. However, it is necessary to solve the following problems to collect service information and manage the registration state in the service registry which manages the service information. (1) Operator intervention is required to collect certain information, and (2) software cannot manage the registration status for each collaborative service.

Fig. 5 illustrates service-information collection from a service environment using the service discovery pattern using Consul. The elements of the service discovery pattern consist of two actors: the service registry, which manages service information, and the self-register agent, which registers service information from installed systems in the service registry. In Consul, the server and client support each actor. When there are multiple subsystems (three in Fig. 5) that constitute the collaborative service, the Consul client refers to a file (service-information file) containing service information of the subsystem to which it belongs and registers the service information in the Consul server, as shown in this figure.

At this time, the file must contain all the service information that needs to be registered, but since the service information includes information that is determined at the time of deployment, it is necessary for the operators to intervene in the service-information registration and determine the information in the file after the subsystem is deployed (Challenge (1)).

As shown in Fig. 5, the service discovery pattern manages service information not by collaborative services but by subsystems equipped with self-register agents. Therefore, if this technology is simply applied and maintenance is started upon notification of service-information-registration completion of a subsystem, problems may occur, such as issuing an alarm due to incomplete registration of the entire collaborative service (Challenge (2)).

Our proposed method enables automatic service-information distribution from deployment to maintenance start to be a one-stop operation on the assumption that the following requirements are satisfied.

1. Enabling service-information collection using the service discovery pattern

Table. 1 Example of Service Information and Possibility of Preset

Type	Main Items	Overview	Possibility of preset
Service Info	Service ID, Service Name	Information about Service	– (ID determined at deployment)
Server Info	Server ID, Server Name, Flavor Info	Information about Server	– (ID determined at deployment)
Server IF Info	IF Name, Subnet Name, IP Address (local or global)	Information about NW-IF of Server	– (Subnet-info determined at deployment)
SSH Info	Username, Port Number, Authentication Type, Password	Information about SSH Connection to Server	✓
Curl Test Info	URL, Method, Body, Header Info (Name/Value)	Information about Curl Test to Server	✓

2. Elimination of the need for operator intervention after the deployment process
3. Implementation of registration-status management for each collaborative service

IV. Proposed Method

To satisfy the requirements mentioned in Section 3B and enable the automatic distribution of service information, our proposed method registers service information, which is necessary for maintenance operations in a maintenance system, by using an Open Source Software (OSS) that uses the service discovery pattern and eliminates the need for operator intervention after the deployment process. It also includes a collaborative-service-information management unit for managing the service-information registration status of a collaborative service and for the maintenance system to start maintenance when the registration of the entire collaborative service is completed.

A. Automatic service-information distribution method

Table 1 shows an example of service information that can be determined before the deployment process and be preset in the service-information file on the subsystem. The table also shows an example of information that is not determined. Information on the SSH connection required for remote connection and information required for the curl command test to confirm the communication status of the Web API can be preset in the service-information file, whereas identifiers for collaborative service management on the deployment system (service information ID), identifiers of servers on the system-construction infrastructure (server ID), and information on the network interface of each server are determined through the deployment process. Therefore, the presetting of information is difficult.

However, when all the information that cannot be preset is determined during deployment, such information is arranged on the deployment system during the deployment process.

Our proposed method registers service information, in which information that can be set in advance before deployment is described in a service information file, and distributes information to be determined during deployment from a deployment system to a subsystem and registers it in a maintenance system using a service discovery pattern.

The flow of the proposed method is shown in Fig. 6. In addition to the deployment and maintenance systems, which use current technology, an automatic service-information distribution system, which relays the service-information distribution, is constructed.

This system has an inter-device cooperation-processing unit, which mediates the cooperation between the deployment system, maintenance system, and collaborative-service-providing system. In the service-information distribution

©Copyright IEICE - APNOMS 2021

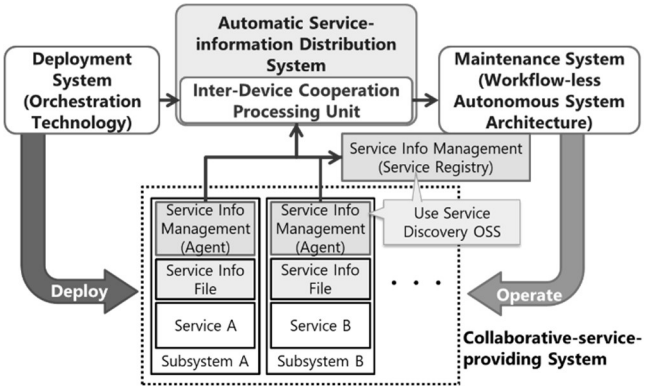

Fig. 6 Flow of proposed method

between a deployment system and collaborative-service-providing system, an inter-device cooperation-processing unit requests configuration information of the deployed system to the deployment system upon notification of collaborative-service-deployment completion from the deployment system and extracts service information of the collaborative-service-providing system from the acquired configuration information. The extracted service information is converted into a format that can be input to the service-information management client (self-register agent) on the subsystem, and the information is input to the subsystem unit through the API of the service-information management client. Thereafter, on the basis of the service discovery pattern, the service information is registered in the service-information management unit (service registry) in units of subsystems together with the preset service information.

The proposed method enables automatic distribution of service information using the service discovery pattern, including information that had previously required manual input after deployment.

B. *Collaborative-service-information management unit*

The service-information management unit shown in Fig. 6 processes a service-information registration request from a subsystem that comprises the collaborative-service-providing system and notifies an inter-device cooperation processing unit that had registered in advance an information-registration completion notice every time service-information registration completion in each subsystem was completed. The inter-device cooperation-processing unit refers to the information of the server included in the notification every time the notification is received and the service information acquired from the deployment system during service-information distribution processing shown in Section 4A, collates it with the list of the servers constituting the collaborative service, and determines the service-information registration status of the entire cooperation service. During judgment, if notifications from all subsystems comprising the collaborative-service-providing system can be confirmed, the service-information registration completion of the collaborative service is recognized by the maintenance system, and if there is a subsystem that cannot be confirmed, the maintenance system waits for the next notification (Fig. 7).

With the proposed method, it is possible to avoid the maintenance start by the maintenance system under the condition that the registration of the entire is not completed by managing the registration situation of the service information of the entire collaborative service not in the subsystem unit and by notifying the maintenance system of this.

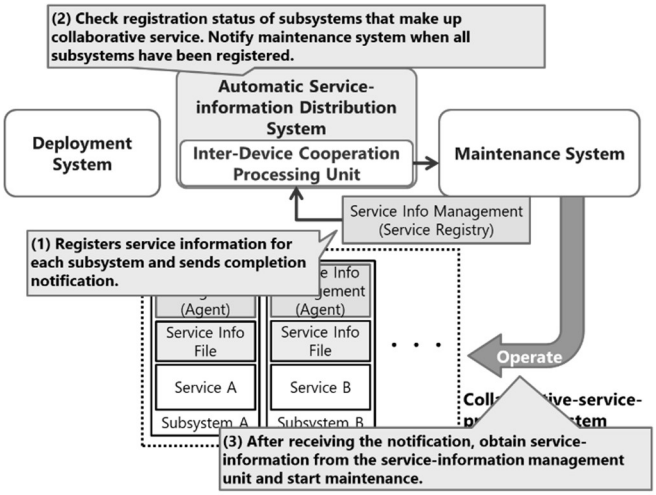

Fig. 7 Outline of collaborative-service-information management unit

V. EXPERIMENT

We implemented the proposed method and evaluated the verification results of the automatic distribution of service information satisfying the requirements discussed in Section 3B and the autonomous execution of maintenance-system operation on the basis of the service information by the maintenance system in a real environment. Since the automatic distribution of the service information with the proposed method had been verified [15], we mainly confirmed the execution action of maintenance-system operation after the distribution of service information in this study.

A. *Configuration*

The configuration of the experimental system is shown in Fig. 8. With the proposed method, an automatic service-information distribution system and inter-device cooperation processing unit are implemented as workers in the maintenance system. Consul is used as the service-information management unit and service-information management client and installed on the maintenance system. Mattermost [16] and Kibana [17] were used as user interface (UI) workers for operators who output information collection, test, and analysis results, and Beats (Metricbeat, Packetbeat) [18] were used as information-collection workers. RabbitMQ [19] was used as the message bus among workers. The service-delivery system was deployed on a virtualization platform (VMware). The experiment was carried out in a scenario in which the server configuration of the service-

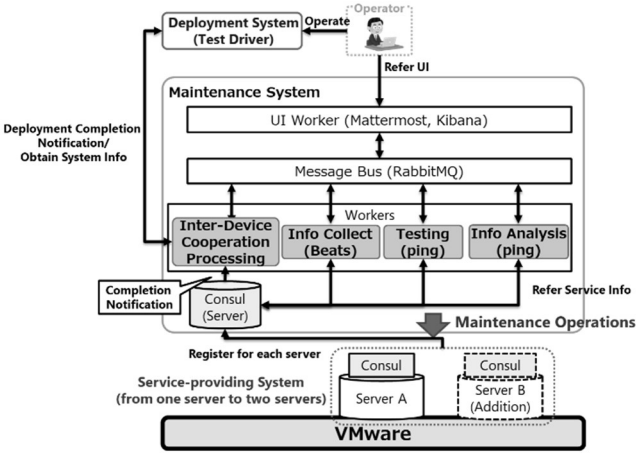

Fig. 8 Experimental System

©Copyright IEICE - APNOMS 2021

providing system increased from one to two. On the deployment system, the deployment-completion notification was transmitted to the inter-device cooperation processing unit, and the test driver, which returns the configuration information of the deployed collaborative service for the configuration-information acquisition request from the inter-device cooperation-processing unit, was used.

B. Results

The evaluation experiment was carried out on the basis of the following three viewpoints.

1. Service information is registered in Consul (server) upon notification of deployment completion to the inter-device cooperation-processing unit.
2. When the service-information registration of the entire collaborative service is completed, the maintenance operation by the maintenance system for the service-providing system starts.
3. When a server comprising a service-providing system is added and a deployment-completion notice is received, the service information on Consul is updated, and the operation of the maintenance system changes in accordance with the changed service information.

Regarding viewpoint 1, we confirmed on the graphical user interface how the service information of the deployed service-providing system (one server) was registered on the Consul server by the operation of the test driver.

Regarding viewpoint 2, the information collection from server A started, confirming that the information collected on Kibana was visualized.

Regarding viewpoint 3, the operation of the test (ping) worker was confirmed when the collaborative-service-providing system included two servers together with the newly deployed server B. Communication between servers A and B was confirmed, and the result was shared with an information-analysis (ping) worker through a message bus, and the analysis result of the communication state between servers was confirmed on Mattermost shared by an information-analysis (ping) worker.

From the above results, the automatic distribution of service information with the proposed method is possible even in a real environment. The method can be linked with the maintenance system that uses our workflow-less autonomous system architecture, enabling the automatic tracking of the operation content due to changes in the service environment triggered by the configuration change in the collaborative service.

VI. Conclusion

We proposed an automatic service-information distribution method that enables cooperation between deployment and maintenance systems, which cooperates a deployment system that uses our orchestration technology, a maintenance system that uses our workflow-less autonomous system architecture, and a collaborative-service-providing system that uses a service discovery pattern through the proposed method, enabling one-stop operation from the deployment of a collaborative service to maintenance start.

We confirmed the feasibility of the proposed method through its implementation and evaluation using an OSS such as Consul and confirmed its superiority from the viewpoint of enabling automatic information distribution and tracking due to the configuration change of a collaborative-service-providing system against the conventional method of inputting service information to the maintenance system manually.

For future work, we will investigate a method of automatically deploying the service-information-distribution client on a collaborative-service-providing system, which is necessary for service-information distribution with the proposed method, on the subsystem in the opportunity of the deployment of the collaborative service.

References

[1] ITIL, Foundations of IT Service Management with ITIL 2011

[2] Service Function Chaining (SFC) Architecture, http://www.rfc-editor.org/info/rfc7665

[3] ETSI ISG NFV, http://www.etsi.org/technologies-clusters/technologies/nfv

[4] K. Takahashi, H. Tanaka, N. Onai, H. Yazaki and H. Kato, "An Orchestrator that realizes Coordination Fulfillment among Multiple Service Providers," IEICE Tech. Rep., vol. 117, no. 491, ICM2017-71, pp. 91–96, March 2018.

[5] K. Takahashi, R. Katayanagi, N. Onai and M. Ohtani, "Support Method to Generate Catalogs for Orchestrator using Catalog Definition Templates," 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), 2019, pp. 1-4, doi: 10.23919/APNOMS.2019.8892886.

[6] N. Tanji, A. Takada and K. Yamagoe, "Autonomous Management Loop by Componentization and Autonomization of Operation Function," IEICE Tech. Rep., vol. 118, no. 118, ICM2018-13, pp. 13–18, July 2018.

[7] T. Ikegaya, K. Takahashi and S. Kondoh, "A Study of Synchronization Method for Autonomous Management Loop," Proceedings of the 2019 IEICE General Conference, B-14-5, 2019.

[8] T. Ikegaya, K. Takahashi and S. Kondoh, "Operation Worker Development Method for Autonomous Management Loop," IEICE Tech. Rep., vol. 119, no. 299, ICM2019-26, pp. 27–32, November 2019.

[9] T. Ikegaya, K. Takahashi and S. Kondoh, "Message Control Method for Autonomous Management Loop," IEICE Tech. Rep., vol. 119, no. 438, ICM2019-43, pp. 11–16, March 2020.

[10] K. Takahashi, T. Ikegaya, R. Katayanagi, S. Kondoh and T. Toyoshima, "A Workflow-less Autonomous System Architecture for Assurance Operations using Choreography Architecture," 2020 16th International Conference on Network and Service Management (CNSM), 2020, pp. 1-5, doi: 10.23919/CNSM50824.2020.9269058.

[11] StackStorm, https://stackstorm.com/

[12] Jenkins, https://jenkins.io/

[13] Pattern: Client-side service discovery, https://microservices.io/patterns/client-side-discovery.html

[14] Consul by HashiCorp, https://www.consul.io/

[15] R. Katayanagi, K. Takahashi and S. Kondoh, "Auto-registration Method of Service Information by Cooperation between Deployment System and Maintenance System," Proceedings of the 2020 IEICE Society Conference, B-14-3, 2020.

[16] Mattermost: Open Source, Self-hosted Slack Alternative, https://mattermost.com/

[17] Kibana: Explore, Visualize, Discover Data | Elastic, https://www.elastic.co/kibana

[18] Beats: Data Shippers for Elasticsearch | Elastic, https://www.elastic.co/beats/

[19] RabbitMQ, https://www.rabbitmq.com/

©Copyright IEICE - APNOMS 2021