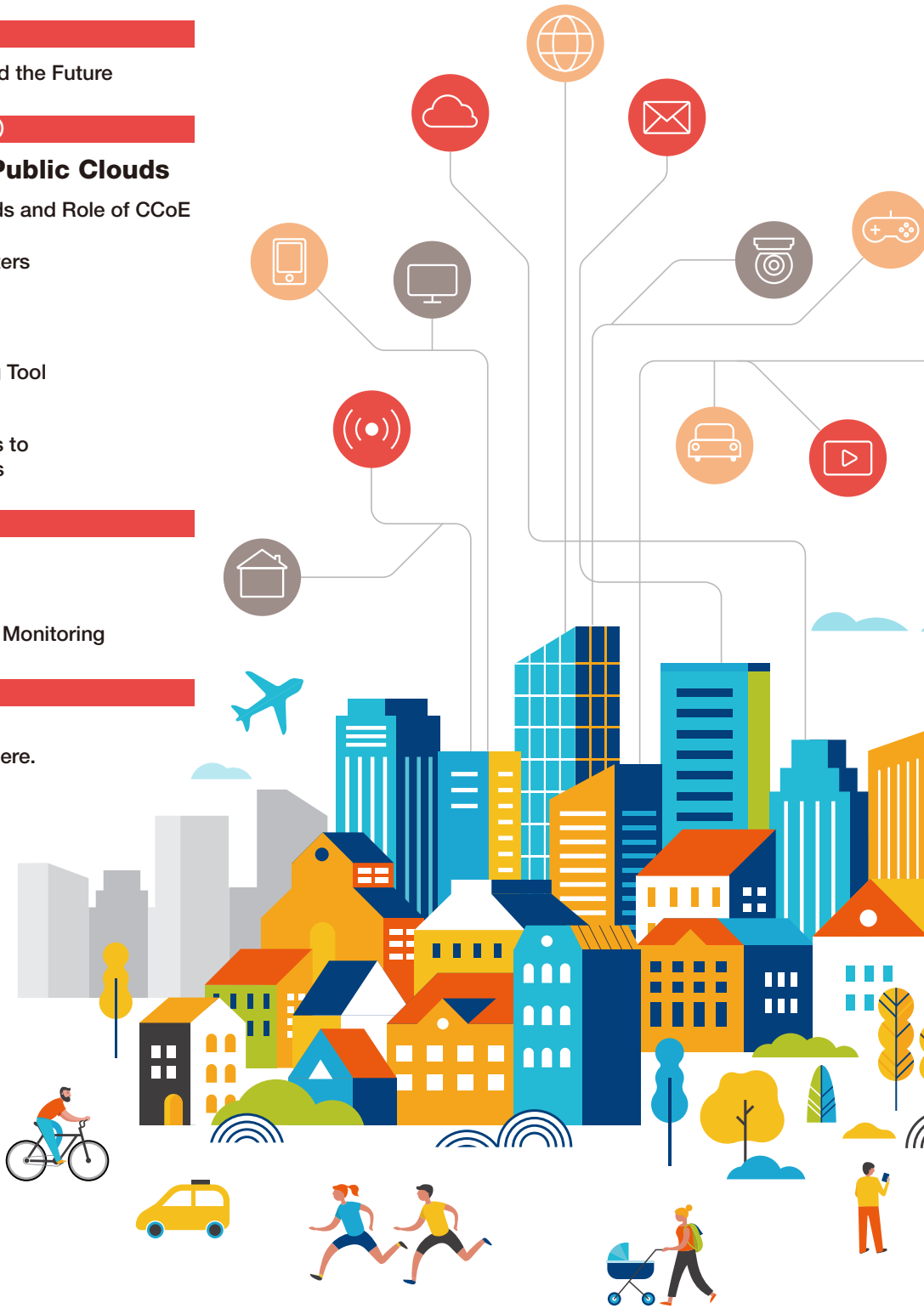
NTT DOCOMO
Technical Journal
Technical Journal
Vol.23 No.1 | Jul. 2021



NTT docomo

NTT DOCOMO R&D Activities Toward the Future

General Manager of
R&D Strategy Department
Takatoshi Okagawa

Social problems in Japan are becoming increasingly diverse and serious as reflected by a shrinking labor force in an aging society, the occurrence of natural disasters, a drop in industrial competitiveness, and the transition to a remote society to prevent the spread of COVID-19. To solve these problems, the Digital Transformation (DX)*1 of society has been moving forward at a rapid pace, and to support this transformation, NTT has proposed the concept of an Innovative Optical and Wireless Network (IOWN) [1]. With a view to promoting DX and making IOWN a reality, NTT DOCOMO R&D Innovation Division is stepping up its research and development efforts centered on a framework called "cyber-physical fusion [2]." This framework is essentially a continuous loop that converts humans, things, and events in the real world (physical space) into information, accumulates that information as data in cyber space, creates value by analyzing that data and applying analysis results to future predictions, knowledge discovery, etc., and feeds that value back to physical space. However, achieving such a cyber-physical fusion will require the evolution of (1) AI, (2) the network, and (3) devices as core technologies and the close coordination of these technologies, as summarized below.

(1) AI is being used to make predictions about the future and drive knowledge discovery by analyzing collected and stored data in cyber space. NTT DOCOMO, for its part, has been applying AI to mobile spatial statistics prepared from operations data of the mobile phone network for use in initiatives such as "AI congestion prediction," which predicts the occurrence of traffic congestion, the scale and time period of such congestion, etc., and optimization of bike relocation in a bike share service.

(2) The network provides connectivity between cyber space and physical space. At NTT DOCOMO, the evolution of the mobile communications network continues with further development of the fifth-generation mobile communications system (5G) known as 5G Evolution and the development of the sixth-generation mobile communications system (6G) beyond 5G. At the same time, we seek to accelerate the process toward a "mobile-fixed integrated network" that merges the fixed and mobile networks that have traditionally developed independently and to achieve the next-generation network including IOWN.

(3) Devices act as contact points with customers in the real world, and at NTT DOCOMO, we have been focusing our efforts on glasses-type devices using eXtended Reality (XR) technology. We have undertaken the research and development of various elemental technologies and have been proactively involved in the creation of heretofore nonexistent technologies such as 8K Virtual Reality (VR) (omnidirectional 8K video) and volumetric video*2 using Head-Mounted Displays (HMDs). Going forward, we plan to implement these novel technologies in devices that can provide customers with new experiences at events and elsewhere.

If we were to classify these technologies according to approach, we would divide them into network technology as a fundamental platform and service technology above that platform to provide customers with services. It is important, however, that these technologies

be closely linked. Furthermore, in addition to technology development, our future efforts will include the strengthening of our service creation and development abilities that will become increasingly important in responding flexibly to severe competition from other companies and to a society with a high degree of uncertainty. We will also promote the evolution of the DOCOMO Open Innovation Cloud as a distributed and consolidated computer infrastructure for connecting the 5G network and service platform and its implementation in real-world fields.

To promote and merge core technologies (1) – (3) and to keep the cyber-physical fusion loop moving, it will be important to interface with other departments in NTT DOCOMO and with the NTT Group while also enhancing our tie-ups with partner companies. Our R&D departments will collaborate with the Corporate Sales and Marketing and Smart-life Business departments to enhance our service creation and development abilities. We will also promote the embodiment of the future mobile-fixed integrated network, further 5G enhancements, and R&D toward 6G in collaboration with the NTT Group. Finally, we will continue to promote research and development on a global scale by expanding tie-ups with domestic and international vendors and ramping up our international standardization activities as reflected by our support for a "5G Open RAN Ecosystem [3]," which was launched with the aim of globally expanding open radio access networks (Open Radio Access Network (O-RAN)*3, virtualized Radio Access Network (vRAN)*4).

REFERENCES

[1] J. Sawada, M. Ii and K. Kawazoe: "(IOWN) Innovative Optical and Wireless Network—Beyond the Internet," NTT Publishing Co., Ltd., 2019.

[2] N. Tani: "R&D for Continuous Creation of New Business Value," NTT DOCOMO Technical Journal, Vol.22, No.4, p.1, Apr. 2021.

[3] NTT DOCOMO Press Release: "Creation of '5G Open RAN Ecosystem' to Accelerate Open RAN to Operators Globally," Feb. 2021.

*1 DX: The use of IT technology to revolutionize services and business models, promote business, and change the lives of people for the better in diverse ways.

*2 Volumetric video: 3D video captured with specialized equipment enabling free viewpoint viewing and interactive video expression.

*3 O-RAN: A radio access system configured with an open interface for improving function extendibility as defined by the O-RAN Alliance.

*4 vRAN: A radio access system for implementing a radio access network in a more open and highly flexible form by applying virtualization technology using general-purpose processors, accelerator, etc.

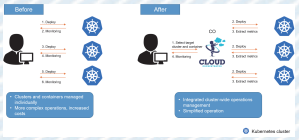
NTT DOCOMO
Technical Journal
Vol.23 No.1 Jul. 2021

[Contents]

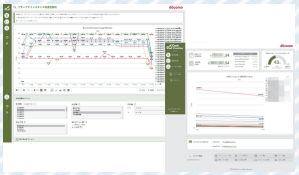
DOCOMO Today

Special Articles on Use of Public Clouds

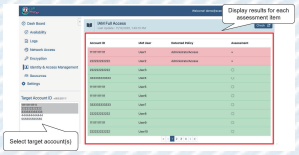
Technology Reports (Special Articles)



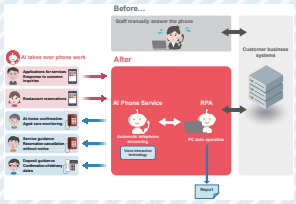
(P.13)



(P.24)

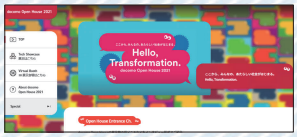


(P.34)



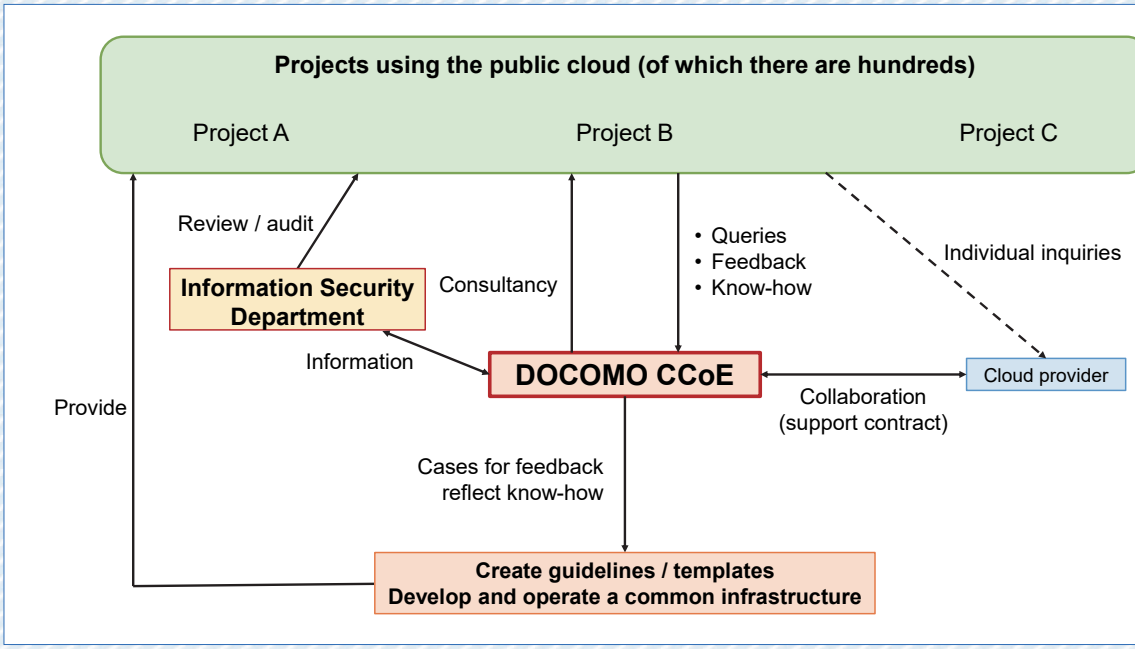
(P.64)

Technology Reports



(P.72)

Event Reports



Technology Reports (Special Articles) NTT DOCOMO’s Use of Public Clouds and Role of CCoE (P.5)
NTT DOCOMO’s internal cloud control system

Technology Reports (Special Articles) Public Clouds CCoE Serverless
Special Articles on Use of Public Clouds

NTT DOCOMO's Use of Public Clouds and Role of CCoE

Innovation Management Department Hiroki Moriya Takeshi Mori
DOCOMO Innovations, Inc. Yasuhiro Naoi

NTT DOCOMO has been providing a multitude of services using public clouds for over ten years. Today, activities continue toward more efficient and optimal use of public clouds centered about NTT DOCOMO's Cloud Center of Excellence (CCoE). This article describes those activities and the public cloud usage system at NTT DOCOMO and introduces recent development trends on the cloud.

1. Introduction

The use of public clouds[*1] in companies and organizations has been increasing rapidly in recent years, and business expansion using public clouds has practically become the norm as reflected by the catchphrase "cloud first." At the same time, the need has arisen for many companies and organizations to expand or reform their business operations by leveraging the swiftness and flexibility of public clouds that are not on-premise[*2]. On the other hand, using a public cloud often requires an approach different from that of constructing and operating a conventional on-premise IT system, and for this reason, there are not a few companies and organizations that cannot introduce and efficiently use public clouds without problems.

In general, a number of issues must be given attention when using a public cloud. These include service testing on introduction, implementation of security measures, drafting of usage policies, establishment of a usage system within the company, acquisition of skills, personnel training, collection of fast-changing cloud information, and know-how development. To make good use of a public cloud, it is important to recognize how its use differs from that of conventional on-premise IT systems and to proactively keep up with changes that can frequently

*1 Public clouds: Cloud computing services that anyone can use over the Internet.

occur on a public cloud.

NTT DOCOMO has also been using public clouds for many of its services for over ten years. As a result, it has faced the issues described above and continues to this day with activities that aim to achieve more efficient and appropriate use of public clouds. In this article, we describe those activities and the public cloud usage system at NTT DOCOMO.

2. Use of Public Clouds at NTT DOCOMO

This section describes the use of public clouds at NTT DOCOMO as of December 2020.

2.1 Scale of Public Cloud Use

At NTT DOCOMO, the use of public clouds began in 2009 for research-and-development and testing purposes. Following this, the range of use began to expand leading to the adoption of public clouds for large-scale commercial services in 2012. The volume of use continued to increase on a yearly basis, and as of December 2020, total use came to more than 900 accounts on Amazon Web Services (AWS)[*3], more than 250 accounts on Google Cloud Platform (GCP)[*4], and more than 50 subscriptions on Microsoft Azure[*5] (hereinafter referred to as "Azure").

2.2 Range of Public Cloud Use

NTT DOCOMO uses public clouds in a wide range of fields. These include Web services, back end system[*6] for mobile applications, data analysis platforms, machine learning, and internal company systems.

2.3 Public Cloud Usage System

NTT DOCOMO has set up a usage system to facilitate the efficient use of public clouds (**Figure 1**). The most outstanding feature in the use of public clouds at NTT DOCOMO is the existence of a definitive Cloud Center of Excellence (CCoE)[*7]. This is a team having a wide range of specialized knowledge related to public clouds, and at NTT DOCOMO, the use of public clouds has been centered on this CCoE.

3. NTT DOCOMO CCoE Activities

This section describes in detail the activities of the CCoE that plays a core role in NTT DOCOMO's public cloud usage system.

3.1 Consultation

Providing support when using public clouds is an important role of the CCoE. Given a certain in-house project, the CCoE provides support for system design when constructing a system on a public cloud, conducts reviews, and presents methods for efficiently satisfying security requirements. Initial system design is particularly important when using a public cloud since it can greatly affect subsequent costs and system operation. The CCoE often provides support at the time of system design for this reason.

In addition, a sudden increase in the number of system users when beginning system operation will inevitably lead to an escalation of costs, so to optimize costs in such a situation, the CCoE will provide support for grasping cost factors, reviewing design, etc.

3.2 Coordination and Optimization of Cloud Costs

In general, a public cloud provides much flexibility in the way that it is used, but at the same time, an expanding number of services and purchasing

*2 On-premise: An environment in which a company owns, maintains, and operates the hardware making up its system.
*3 AWS: A cloud computing service provided by Amazon Web Services, Inc.
*4 GCP: A cloud computing service provided by Google LLC.
*5 Microsoft Azure: A cloud computing service provided by Microsoft Corporation.
*6 Back end system: The system that is centrally operated on servers or other hardware as opposed to operation on user mobile terminals or computers.

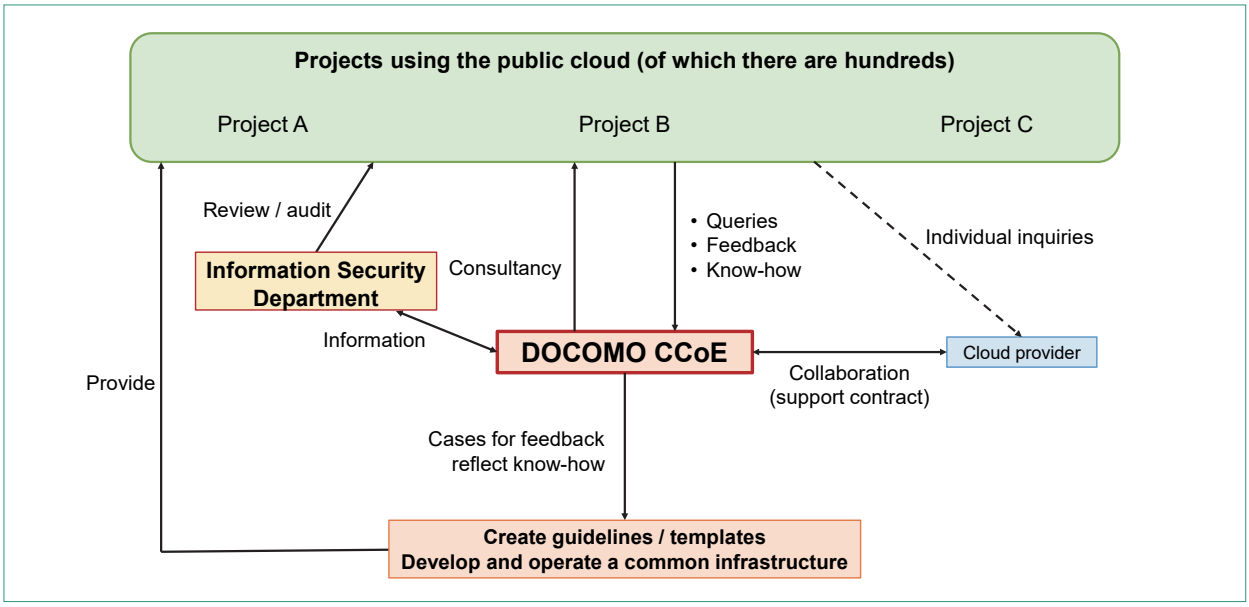


Figure 1 NTT DOCOMO public cloud usage system

options can easily make the payment of expenses and account processing overly complicated. An increase in the number of cloud-based projects can therefore increase the labor involved in these tasks. For these reasons, the CCoE centrally concludes contracts with cloud operators and coordinates and processes payments for each project in a lump-sum manner thereby reducing the workload of each project.

There is another advantage to making lump-sum payments in addition to reducing the load of business processing. Public cloud operators offer a volume discount option as the amount of use increases, so lumping payments together can ensure a certain volume and minimize usage fees over the entire company.

3.3 Collection and Dissemination of Up-to-date Information

Public clouds are evolving rapidly and keeping up with the latest information about clouds can be difficult for project members burdened with project tasks. The CCoE has therefore taken the initiative in keeping up to date on the state of public clouds. For example, the CCoE actively participates in technical events related to public clouds to gather information and give presentations. It also reflects gathered information in guidelines as know-how, performs verification tests on its own, and disseminates information to project members within the company.

3.4 Creation and Dissemination of Cloud Business Support Tools

The CCoE prepares a variety of support tools and disseminates them within the company so that each project can efficiently use public clouds.

Although the cloud offers many and varied functions for accelerating business development, using them incorrectly may lead to a fault on the cloud and service interruption and making erroneous settings may cause a security accident to occur. It is therefore necessary that users have the technical

*7 CCoE: An exclusive team within an enterprise that establishes best practices and creates essential systems and governance to make cloud usage successful.

knowledge for using the cloud correctly, but the knowledge level of users varies, so from a company perspective, how to raise the level of technical knowledge of all employees is an issue of concern.

With the above in mind, the CCoE prepares guidelines that feature cloud usage methods from the NTT DOCOMO point of view. These guidelines make it possible to acquire the minimally required amount of knowledge on the use of public clouds in a short time even for users with little knowledge of clouds. These guidelines cover the use of AWS, GCP, and Azure. Here, we present the list of guidelines prepared for AWS in **Table 1** and an excerpt from those guidelines in **Figure 2**.

Table 1 List of guidelines (AWS)

No	Type	Description
1	Cloud development guidelines	Describes mindset and manners when using the cloud and guidelines that should be considered and implemented in each phase of development flow. Covers important points such as design and security and minimizes mistakes in usage.
2	Security design patterns	Minimizes omissions in security considerations when using the cloud and constructing a system. Prepares requirements in line with ISO/IEC27017 beforehand to enhance compliance with ISO management measures. Lists essential security requirements when constructing a system using AWS.
3	Security templates	Provides AWS CloudFormation templates for generating instance groups that provide network configurations, network filtering functions, and basic functions that take security design patterns into account. Simplifies the implementation of security measures.
4	IAM design patterns	Lists best practices in the design of IAM policies within NTT DOCOMO in line with AWS account usage patterns.
5	Incident response guidelines	Lists responses to incidents such as cyber attacks in in-house systems and service-providing systems on the Internet using AWS combined with actual case studies.
6	Cost optimization guidelines	Describes cost determination/analysis methods and cost reduction/optimization methods for personnel managing AWS costs.
7	System migration guidelines	Lists key points and matters deserving attention during a system migration based on knowledge gained in past migration examples to facilitate a smooth migration from an on-premise system to AWS.
8	Common platform guidelines	When beginning to put multiple accounts and multiple systems into operation, raising efficiency by unifying operations and standardizing operation systems can be an effective approach. These guidelines list methods for smoothly achieving greater efficiencies in operations by leveraging the characteristics of cloud computing.
9	Container guidelines	Describes which tools to use and how to use them to achieve effective and safe use of containers in each project. Targets personnel who wish to incorporate containers in service development/operation.
10	Serverless guidelines	Lists methods for efficiently and effectively using serverless computing in each project when developing and operating serverless systems on AWS. Answers questions like "What is the best way to implement a serverless system?" and "In what way and in what kind of services can serverless computing be used?"
11	DevOps guidelines	Describes which tools to use and how to use them to achieve efficient and effective practice of DevOps in each project. Targets personnel who wish to incorporate a DevOps mindset in service development/operation.

IAM: Identity and Access Management
IEC: International Electrotechnical Commission
ISO: International Organization for Standardization

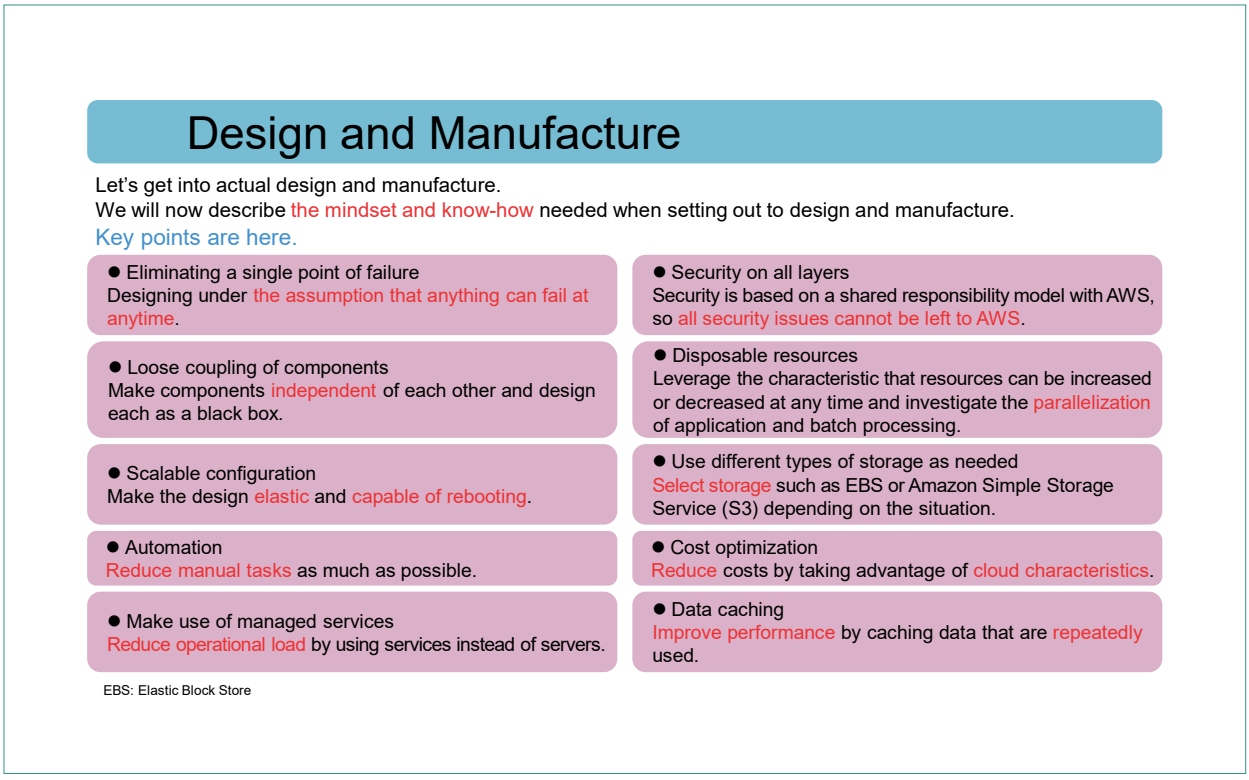


Figure 2 Guideline example (excerpt from cloud development guidelines)

As one of these guidelines, security design patterns[*8] aim to minimize omissions in security considerations when constructing a system using a cloud. They do this by listing requirements in accordance with the security check items (ISO/IEC27017, JISQ27002, etc.) of the NTT DOCOMO Information Security Department plus requirements in a cloud environment and requirements and specifications when constructing a system using AWS alongside the functions and services provided by AWS. These patterns facilitate compliance with security check items. The person in charge of system construction need not test each and every security check item to satisfy requirements. Constructing the system while referring to these security design patterns according to use will satisfy the security check items in due course. An excerpt from these security design patterns is shown in **Figure 3**.

The speed at which public clouds add functions and release updates requires that these support tools be updated just as frequently. The CCoE incorporates agile development[*9] in these support tools and is actively involved in keeping up with frequent function additions and system updates.

4. Latest Development Trends at NTT DOCOMO

This section describes a recent case study on applying serverless technology at a new NTT DOCOMO development site.

*8 Security design patterns: Guidelines that describe methods for satisfying security requirements when using public clouds.

*9 Agile development: Generic name for a group of lightweight application development methods for quick and adaptive software development.

ISO/IEC27017 or JISQ27002		AWS Security Design Patterns		
Section No.	Management Measure	Requirements in a cloud environment	Requirements and specifications when constructing a system using AWS and the functions and services provided by AWS	
12.1.4	The development environment, test environment, and operation environment should be separated to reduce unauthorized access or risk of change to the operation environment.	・A system constructed in a cloud environment shall be divided into commercial, development, test, and backup environments. —Separate contracts if possible —Separate networks, etc. ・Although the cloud has a configuration exposed to the Internet, it shall be possible to logically separate a network by firewalls, routings, etc. and to configure network segments according to roles. ・Given a service in a multitenant environment, the ability to separate use should be verified so that other users and environments (commercial, development, etc.) are not affected. —Physically separable —Logically separable	**【Requirements and specifications】** ・Make AWS accounts separate when separating commercial, development, test, and backup environments. If difficult, separate using the VPC function. ・Separate network segments according to role (DMZ, remote-connection network segment, internal-server network segment, etc.) using the VPC function or subnet function.	**【Functions and services provided by AWS】** ・Environments can be logically divided among tenants by allocating AWS accounts by application, and network environments can be logically divided using the VPC function. If physical division of environments is required, either of the following can be used: —Hardware-exclusive instance —Dedicated hosts ・Multiple virtual subnets can be created within a VPC by the subnet function, which means that subnets can be configured according to role in this way. ・"VPC peering" enables two VPCs to be treated as if they exist within the same network. **【Note】** ・Availability can be improved by a Multi-AZ configuration.
13.1.3	Information services, users, and information systems should be separated into different groups on the network.			
14.1.2	Information included in application services provided over the public network should be protected from malicious behavior, contract disputes, and unauthorized release or alteration.	Function requirements		
14.1.3	Information included in transactions of application services should be protected to prevent the following items from occurring: —Incomplete communications —Erroneous communication-path settings —Unauthorized message alteration —Unauthorized release —Unauthorized message duplication or regeneration	Function requirements		

AZ: Availability Zone
DMZ: DeMilitarized Zone
JIS: Japanese Industrial Standards
VPC: Virtual Private Cloud

Figure 3 Example of security design patterns (listed with reference to the ISO27017 standard in the version for external use)

4.1 Appearance of Serverless Technology

In the world of cloud computing, so-called DevOps, which increases productivity by integrating development and operations that had previously been done separately, and architecture design, which assumes a transition from the use of conventional virtual machines[*10] to containers[*11], have found widespread use. The use of containers has been successful in accelerating the development and release cycle, but operation monitoring and maintenance operations for security purposes are still needed the same as with virtual machines. Despite this trend in containerization, public cloud operators including AWS have already begun to provide a series of managed services[*12] as "serverless" technology that includes server operation management via middleware[*13]. This is technology for using a cloud infrastructure that revolves around a different axis than that of containerization. It enables developers and operation managers to rely on the cloud operator for all infrastructure operations and management without having to worry about the existence of servers. It also enables more resources to be concentrated in the design of business logic.

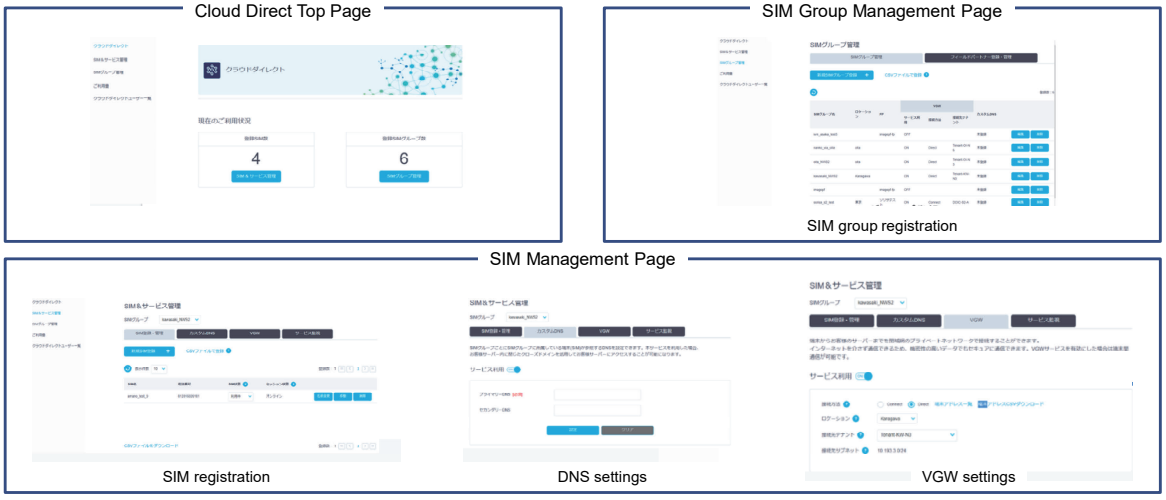
4.2 Case Study of Serverless Application Development at NTT DOCOMO

The Web portal for the DOCOMO Open Innovation Cloud released in June 2020 by NTT DOCOMO is a general Web application system based on React[*14]

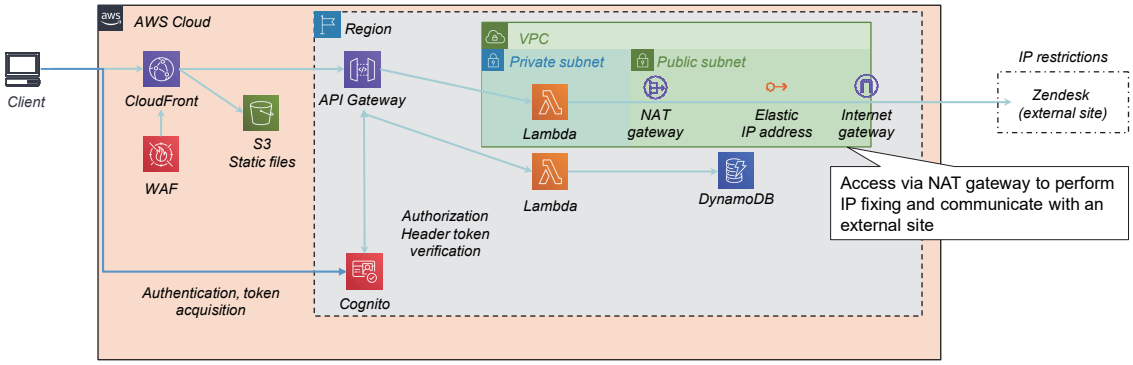
*10 Virtual machines: Computers such as servers constructed in a virtual manner by software.
*11 Containers: As one type of computer virtualization technology, a method for creating a dedicated area called a container on one host OS and running necessary application software within that container.
*12 Managed services: Cloud services whose resource provisioning, operation, etc. are mostly the responsibility of the cloud operator. Among cloud computing services, these are referred to as PaaS and SaaS, for example.
*13 Middleware: Software providing functions for common use by multiple applications.

as described below (**Figure 4** (a)).

In the development of the DOCOMO Open Innovation Cloud, the time taken until its first release was four months and operation and maintenance personnel was limited to the minimum number needed. Here, to offload*15 server management to



(a) Screenshot of each Web page



(1) In the Client's initial access, Cognito issues an authentication token at the time of user authentication. Then, from this point on, the Client can use this authentication token issued by Cognito to gain access again.
(2) A request from Client is received at CloudFront linked to WAF. CloudFront passes a static content request to S3 and an API request to API Gateway.
(3) A static content request treats S3 as a Web server that directly returns that content to the client.
(4) An API request is processed by back-end Lambda via API Gateway.
(5) Lambda uses DynamoDB as persistent storage for reading/writing user data and returns API processing results to Client.

API Gateway: API proxy service managed by AWS
CloudFront: CDN service managed by AWS
Cognito: User management service managed by AWS
DynamoDB: A managed service providing key-value data stores
Lambda: A managed service in which a function is the unit of processing
S3: An object storage service managed by AWS
WAF: A Web application firewall managed by AWS

(b) System configuration around the Web

Figure 4 DOCOMO Open Innovation Cloud

*14 **React:** A JavaScript library for creating user interfaces.

*15 **Offload:** The transfer of system, service, or network processing to separate but similar services to reduce the processing load of the original service.

AWS as much as possible, we adopted serverless architecture overall using, for example, AWS Lambda*16 (Fig. 4 (b)). As a result, we shortened development time considerably and succeeded in releasing the Web site in a relatively short period.

At the time of this writing, the DOCOMO Open Innovation Cloud has been in operation for one year. There have been zero problems due to infrastructure such as the network or servers. In terms of availability, we have benefited from the fact that redundancy and fault tolerance*17 are built into this serverless architecture beforehand. We also achieved our initial goal of keeping the number of operation and maintenance personnel to two or three people. In addition, this serverless service can automatically deal with situations in which an increase in user access requires scalability. As for charges, you only pay for what you use, so we were able to significantly reduce costs in this project.

5. Conclusion

This article described the use of public clouds at NTT DOCOMO and the role played by the CCoE. Amid expectations that the use of public clouds will expand from here on, using them wisely in a way that leads to business success will become increasingly important. For this reason, we would like to see the support provided by the CCoE expand even further so that each and every project can make full use of public clouds.

Innovation in prompt deployment and practical use of new serverless technology in commercial services is a next-generation challenge. With this in mind, we will promote efficient development practices within NTT DOCOMO with a view to increasing the number of successful case studies.

*16 AWS Lambda: A type of FaaS provided by AWS that provides an execution environment for application code so that the user need only register created source code to run the application.

*17 Fault tolerance: The ability of a function to continue operation as usual even in the event of a system fault.

Technology Reports (Special Articles) Kubernetes AWS Containers
Special Articles on Use of Public Clouds

Managing Multiple Kubernetes Clusters with a Cloud Orchestrator

DOCOMO Innovations, Inc. Masato Takada Yas Naoi

Kubernetes was originally created by Google developers, but is now developed as an open-source project by the CNCF. It is a system that enables users to operate containerized applications efficiently in any environment, and is currently used by many companies including industry leaders. Around 2018, a growing number of companies started using multiple Kubernetes clusters in individual projects, but at the time there were no tools for managing multiple Kubernetes clusters. Therefore, DOCOMO Innovations, Inc. developed an open-source Cloud Orchestrator to facilitate the centralized management of multiple Kubernetes clusters. This software is currently being used on DOCOMO's internal commercial systems, an example of which is described in this article.

1. Introduction

With the advent of Docker*1 in 2013, applications and system environments could be logically separated by using container virtualization*2 technology to virtualize containers at the OS level. This made it possible to deploy*3 and update containers in workloads*4 even faster than before. Docker, on the other hand, has had issues related to container management, scalability, and automatic recovery. Users have had to solve these issues before using each environment, but this can involve a lot of work.

Kubernetes [1], which has recently gathered a broad community of users, is an open-source solution

*1 Docker: Container-type virtualization software. A registered trademark of Docker Inc.
*2 Container virtualization: A computer virtualization technique where dedicated areas called containers are created on a single host OS, and the necessary application software is run within these containers.
*3 Deploy: Installing applications by placing them in their execution environments.

that aims to solve this problem. Kubernetes takes its name from the Greek word for "helmsman," reflecting its purpose as a tool for managing and automating the operation of containers. It provides developers with great benefits such as management of multiple containers, autoscale*5, and automatic recovery functions without having to prepare a framework*6 for containers on their own. Kubernetes has already become a de facto standard in the field of cloud computing. For example, Amazon Web Services (AWS)*7 uses Elastic Kubernetes Service (EKS)*8, Microsoft Azure*9 uses Azure Kubernetes Service (AKS)*10, and Google Cloud Platform (GCP)*11 uses Google Kubernetes Engine (GKE)*12. In addition to these managed services*13, there are also many other services provided by private cloud*14 vendors such as VMware and Red Hat.

At DOCOMO, some projects started deploying multiple Kubernetes clusters on their own systems from around 2017. However, when using multiple clusters*15, system operators were faced with issues of increased operating costs and variations in resource utilization between clusters. For this reason, we developed a Cloud Orchestrator (CO) as an open-source orchestration tool for managing multiple Kubernetes clusters on Drupal*16. This article describes the features of CO and presents some examples of its use within the company.

2. Kubernetes

2.1 Overview

While Docker is a useful tool for running on a single server, it suffers from a number of issues in large-scale environments consisting of multiple servers. Kubernetes is a container orchestration tool designed to manage containers for large-scale environments of this sort, and has become the de facto standard worldwide. Kubernetes evolved from Borg, which was originally developed by Google engineers. Having accumulated the container orchestration know-how that was used by Google, Borg became the progenitor of Kubernetes. In 2014, Kubernetes was open sourced in the Kubernetes project, which was transferred to the Cloud Native Computing Foundation (CNCF)*17 in 2016 for community-based development.

2.2 Architecture

As shown in **Figure 1**, Kubernetes consists of two types of nodes: a master node and worker nodes. In a cluster, applications and their setting values are managed in units called resource objects. A resource object is defined by a manifest file,*18 which the user works with when creating or updating the resource object. The constituent elements of Kubernetes are as follows:

1) Master Node

A node that manages an entire cluster and performs the roles of worker node management and pod management. A control plane is set as the default when deploying a cluster. The control plane is a component that controls the components that control a cluster and manages the cluster's internal state and configuration. The user operates the entire cluster by using a command line tool called kubectl (described later) to access the Application

*4 Workload: An indicator of the size of a system's load, such as the CPU utilization rate. In particular, in a public cloud environment, the workload may represent the system itself, including the OS and application code running on the cloud. In this paper, we use the term in this latter sense.

*5 Autoscale: A system that automatically adjusts the number of virtual servers on demand according to the quantity of resources required for processing at any given time, such as network traffic and CPU usage.

*6 Framework: Software that encompasses functionality and control structures generally required for software in a given domain. In contrast to a library in which the developer calls individual functions, code in the framework handles overall control and calls individual functions added by the developer.

*7 AWS: A cloud computing service provided by Amazon Web Services, Inc.

*8 EKS: A managed Kubernetes service in AWS.

*9 Microsoft Azure: A cloud computing service provided by Microsoft Corporation.

*10 AKS: A managed Kubernetes service in Microsoft Azure.

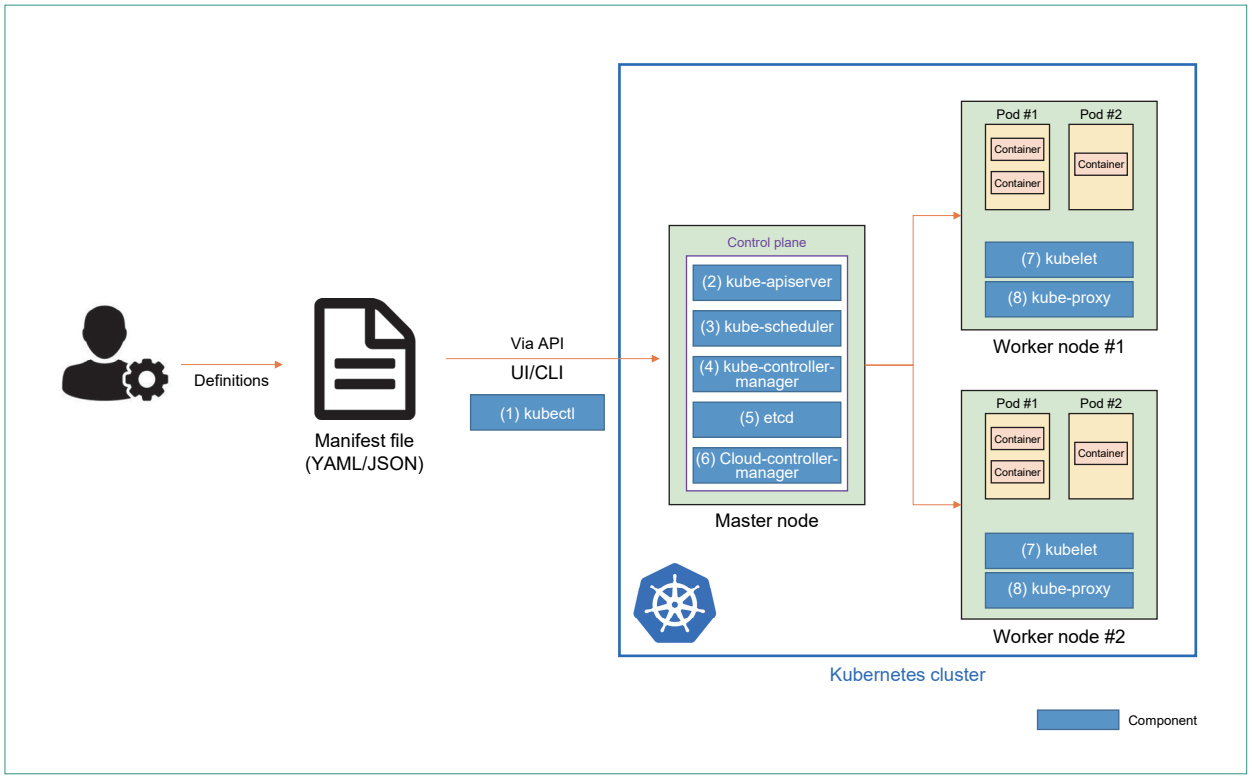


Figure 1 Kubernetes architecture

Programming Interface (API)*19 server provided by the control plane.

2) Worker Node

A node hosting a pod that stores the application containers. This includes components such as kubelet and kubeproxy, which are described later.

3) Pod

An execution unit in a Kubernetes application that encapsulates the application's container, storage, network information (network ID, IP address, etc.) and options for managing execution methods. It can also store multiple containers.

4) Manifest File

A file that describes the resource configuration, using JavaScript Object Notation (JSON)*20 or YAML (a recursive acronym for "YAML Ain't Markup Language")*21 as the file format. By declaring this file via the API, it is possible to manipulate resources in the cluster.

2.3 Components

The components shown in Fig. 1 are described below.

(1) kubectl

A command used by the user to send requests to kube-apiserver in order to create, update and delete resources.

*11 GCP: A cloud computing service provided by Google LLC.
*12 GKE: A managed Kubernetes service in GCP.
*13 Managed service: Cloud services whose resource provisioning, operation, etc. are mostly the responsibility of the cloud operator. Among cloud computing services, these are referred to as PaaS and SaaS, for example.
*14 Private cloud: Refers to an in-house cloud system configured in a corporation or organization, and provided to various in-house divisions or group companies. In contrast, open cloud services that do not restrict their services to certain users are called "public cloud" services.
*15 Cluster: A grouping of multiple servers as a single server group.
*16 Drupal: An open-source content management system, similar to WordPress and Joomla.
*17 CNCF: A project of the Linux Foundation that was created in 2015 to support the development of container technology and collaboration with the technology industry on the advancement of this technology.
*18 Manifest file: A configuration file that declares the functions and other items used by an application. A manifest file must be prepared for each application of every Kubernetes resource.

(2) kube-apiserver

This is responsible for the front end that provides the external API of a Kubernetes cluster.

(3) kube-scheduler

Monitors to check if a new pod has been assigned to a worker node. If not, it assumes responsibility for running the pod. Scheduling decisions are made in consideration of several factors, such as resource utilization and hardware/software/policy constraints.

(4) kube-controller-manager

A component that manages the status of worker nodes and pods via kube-apiserver. It takes responsive action when a node goes down, and manages pod replication.*22

(5) etcd

A key-value store*23 that stores all the Kubernetes cluster information. It can also be used as a Kubernetes data store*24.

(6) cloud-controller-manager

Manages objects specific to a cloud operator, such as nodes, routing, and storage.

(7) kubelet

An agent that runs inside worker nodes and guarantees the operation of each pod. It also monitors the content defined in the manifest file to make sure they match the container settings, and manages the execution environment*25 of nodes and containers.

(8) kube-proxy

Performs communication control (routing) between containers.

2.4 Supported Functions

Kubernetes not only manages containers on multiple servers, but also supports a number of painstaking aspects of operation, such as automatic scaling of containers and automatic recovery*26 in the event of a failure. Some key features are discussed below.

1) Network Load Balancing

When accesses are concentrated on some of the containers, the state of these containers is stabilized by distributing traffic to other containers.

2) Rolling Updates/rollbacks

To apply updates, the user only needs to change the state of the container in the manifest file. Kubernetes will then take care of applying this updated state (rolling update). If an application update fails, it can be easily reverted to its previous state by simply restoring the manifest file to its previous state (rollback).

3) Automatic Picking

For each task, it is possible to define which nodes should be executed, which resources should be used, and the priority for each task.

4) Automatic Repairs

If a container is stopped due to a failure or some other issue, the system automatically detects this state and restarts it.

2.5 DOCOMO's Challenges with Kubernetes

One project at DOCOMO has already been using several large Kubernetes clusters on commercial systems since around 2017. However, Kubernetes is only able to manage its own cluster, so the

*19 API: A specification describing the interface whereby software components can exchange information with each other.

*20 JSON: A data description language based on JavaScript object notation.

*21 YAML: A notation and processing format for describing data structures, similar to XML and JSON.

*22 Replication: In a database, a mechanism for providing redundancy and creating backups by replicating data to other servers.

*23 Key-value store: A storage system that manages records (data) as combinations of keys and values, unlike a conventional relational database.

*24 Data store: A system that stores data.

*25 Execution environment: A system environment that only supports execution processing and lacks systems required for software development (e.g., libraries and test environments).

*26 Automatic recovery: A system that automatically switches to a redundant standby system in the event of a system failure.

system operators had to manage multiple Kubernetes clusters individually. For example, to ascertain the status of containers across multiple clusters, it was necessary to access each individual cluster to check its status. Furthermore, since the clusters were divided according to their function, there was a large disparity in resource utilization between clusters. For example, while one cluster had to temporarily add instances*27 due to a shortage of resources, there were other clusters with resources that were sitting idle. These issues would not have arisen if load balancing*28 and job scheduling functions had existed so that tasks could be distributed among Kubernetes clusters.

When this issue came to light in around 2018, there was no open-source software or external tools capable of solving it. For this reason, we developed CO. From around 2019, other similar products became available from leading companies (such as Uber and Rancher) that have been using Kubernetes intensively.

3. What is CO?

3.1 Development Background

DOCOMO's CO project was launched in early 2018 on an open-source platform called Drupal. Our initial goal was to manage multiple accounts, primarily AWS. DOCOMO has a large number of accounts, and it is not uncommon for a single project to have multiple accounts for different purposes such as development, Quality Assurance (QA)*29 and commerce. The AWS website is basically divided into services by region*30, so even parts of the service cannot be checked on the same page if the service spans multiple regions. As a result, there were many cases where developers launched instances in various regions without the administrator's knowledge, and then failed to stop or delete these instances, which only came to the administrator's attention when viewing the billing information*31. In order to prevent such problems, CO provides a function for visualizing the list of resources under all registered account information on the same page, a function for automatically stopping a cluster based on resource utilization and time constraints, and a single sign-on*32 function. When we ran into Kubernetes issues within DOCOMO, we extended CO based on the features we were using for AWS and started applying them to Kubernetes clusters around the end of 2018.

3.2 Architecture & Components

CO consists of three main parts: a user interface (including a portal function), various functional parts for purposes such as user management and cluster management, and a connector part for cloud and Kubernetes clusters.

As shown in **Figure 2**, CO provides portal functions. From this screen, it is possible to see at a glance the cloud information and Kubernetes clusters that are currently in use. In addition, by bringing up the cluster information, it is possible to visualize the current status of information in the form of tables and pie charts. The display can be customized from the management screen and can be freely changed by the user. In addition to this visualization function, the portal functions include many

*27 Instance: A virtual server that is made available on demand in cloud computing. A virtual server has a sporadic life cycle from start to finish. For example, it might start and finish only when a certain process takes place.
*28 Load balancing: A mechanism for distributing high-load processes across multiple servers.
*29 QA: In software development, the act of ensuring the quality of deliverables or providing quality assurance.
*30 Region: The region in which a data center providing cloud services is located.
*31 Billing information: Information about charges accrued through the use of cloud services.
*32 Single sign-on: The ability to log in to multiple services with a single account. Allows users to access all the functions to which they are entitled by only performing a single system authentication step.

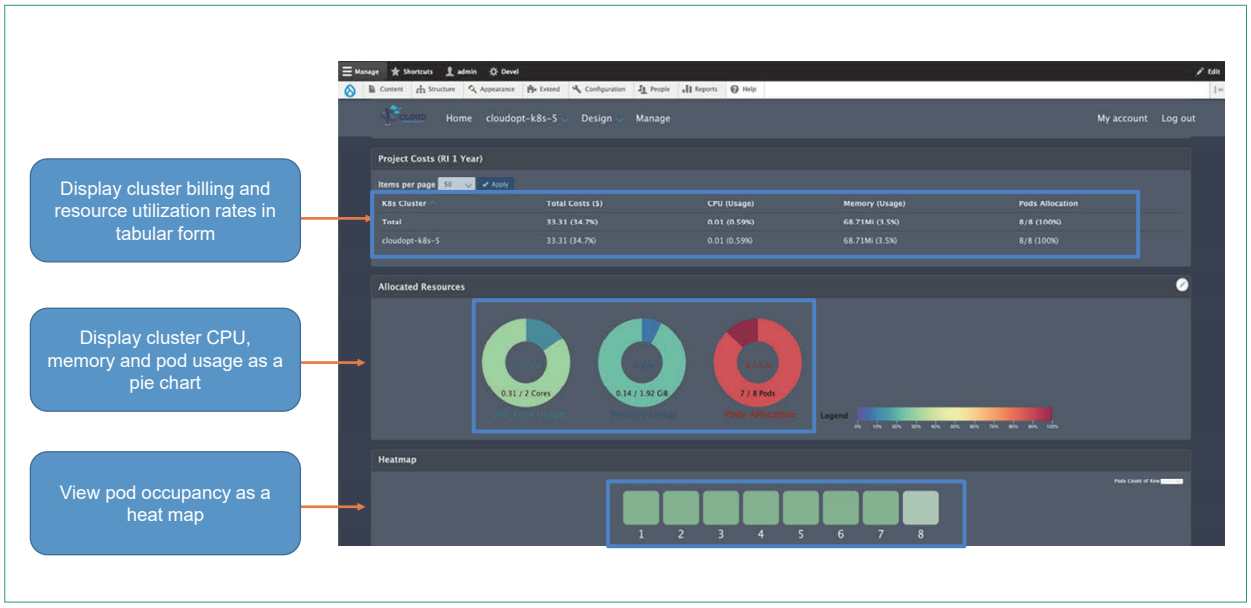


Figure 2 CO portal screen

other functions such as deploying tasks to be run on a cluster, and selecting clusters for deployment. Furthermore, since Drupal provides REST (REpresentational State Transfer) API*33 functions, it is possible to execute functions on the portal via the API.

As shown in **Figure 3**, CO provides many functions such as user management, cluster management, and job scheduling. Drupal's role*34 management features enable the definition of basic CRUD (Create, Read, Update and Delete)*35 functions for any function or service. By extending this functionality, CO can specify CRUD operations for all resource objects in a Kubernetes cluster. With this role management, even if different users deploy their own tasks on the same cluster, interference between these users can be prevented as long as each user is granted a role that can only affect his or her own tasks (i.e., the minimum necessary role). This idea is a fundamental feature underpinning the cost optimization discussed below.

The connector part uses APIs provided by various clouds and Kubernetes clusters, which we modularize and make available to users. The users can turn on these module functions and start receiving information periodically.

3.3 Key Features

This section describes the main functions of CO.

1) Single Sign-on

By making use of Drupal's single sign-on feature, CO can be linked to various clouds. Also, since Kubernetes clusters do not have user authentication functions, CO provides its own user authentication functions.

*33 REST API: An API conforming to REST. REST is a style of software architecture developed based on design principles proposed by Roy Fielding in 2000.

*34 Role: A group of users that grants certain privileges to its members.

*35 CRUD: An acronym representing the four basic operations of software used to provide systems and services (create, read, update & delete).

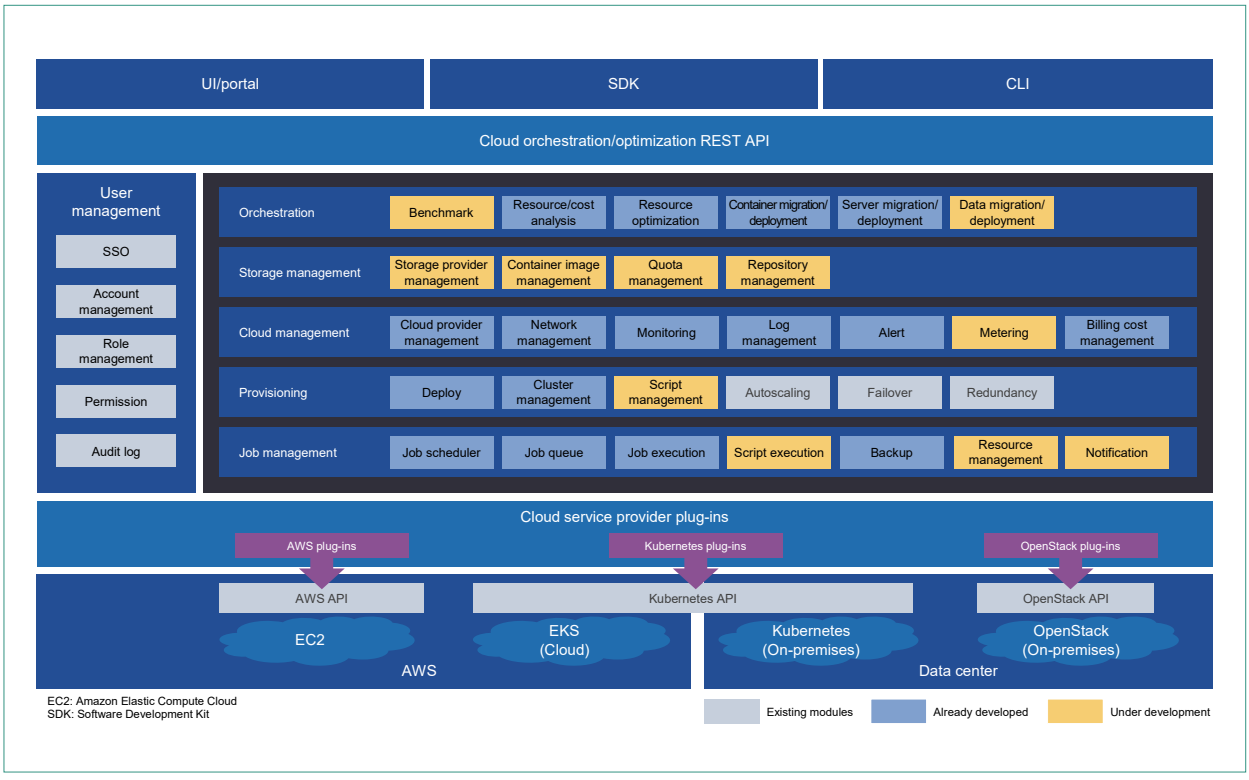


Figure 3 Cloud orchestration/optimization components

2) Resource Optimization

When CO manages multiple Kubernetes clusters, this function automatically performs load balancing between clusters. For example, when deploying any container, it automatically deploys it to the least occupied cluster based on the current resource usage situation. It is also possible for the user to select the deployment destination cluster beforehand.

3) Scheduling

CO includes time and resource based scheduling functions. For example, when a user wants to perform batch processing[*36] at a particular time of day, such as late at night, the task's deployment time and end time can be set in advance. It is also possible to deploy low-priority tasks that run only when resources are available.

4) Cost Calculation

We specified our own logic to calculate the total instance cost of the master node and worker nodes based on the resource usage (memory, CPU, pod counts). This makes it possible to calculate the cost in Namespace[*37] units.

5) Multi-cloud Management

Like public/private cloud solutions such as AWS, OpenStack[*38] and VMware, Kubernetes clusters can also be managed in a unified manner. Users can check their resource usage status in a unified

*36 Batch processing: The automatic processing of data in batches without user interaction.

*37 Namespace: In Kubernetes, a virtual grouping that combines several different resources into a single unit.

*38 OpenStack: Cloud-infrastructure software that uses server virtualization technology to run multiple virtual servers on a single physical server. It can allocate virtual servers to different cloud services in use. OpenStack is open-source software.

manner even if they are working with different accounts or clusters.

3.4 How to Use CO

CO is an open-source Drupal project that can be used by anyone [2].

4. DOCOMO Case Study

4.1 Efficient Management of Multiple Clusters and Free Resources

1) Managing Multiple Clusters

CO is utilized in DOCOMO's commercial system. This system features multiple EKSs, with each cluster having more than 100 nodes. As shown in **Figure 4**, before the introduction of CO, administrators had to check the status of clusters and pods from the management screen of each cluster, and had to use a Command Line Interface (CLI)*39 to deploy containers, which made their work difficult.

However, after the introduction of CO, administrators no longer needed to check each cluster individually because CO obtains metrics*40 from every cluster. Also, when the management screen is used to register the details of a container and the Kubernetes cluster in which it is to be deployed, CO can perform the deployment automatically. If multiple Kubernetes clusters are specified as the deployment destination, then the system can automatically deploy containers to the selected cluster group and obtain metrics from each cluster. As mentioned above, in cluster selection, the user does not have to select a specific cluster, and can instead leave CO to use its resource optimization

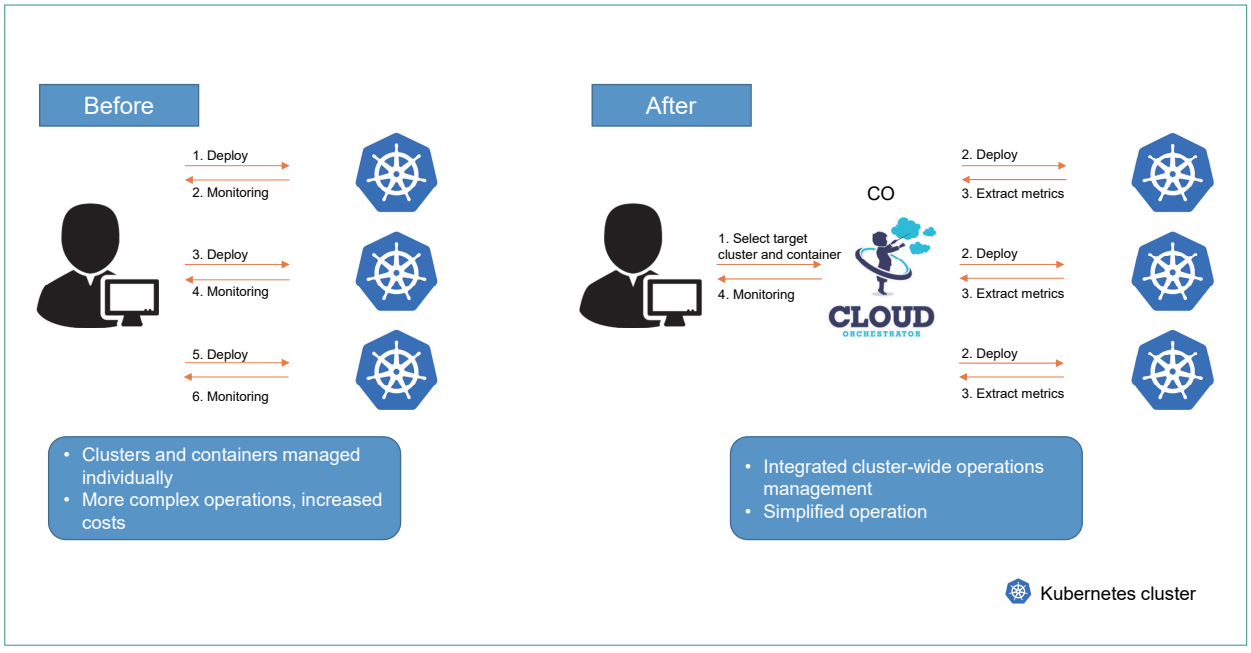


Figure 4 Changes resulting from CO adoption

*39 CLI: An operating method in which all instructions to a computer or software are given in the form of text.

*40 Metric: A quantitative item of information corresponding to the value of some parameter at a particular point in time, such as the CPU utilization rate, memory usage, or number of pods.

function to automatically select a free cluster.

2) Efficiency of Available Resources

As shown in **Figure 5**, CO has a time-based and resource-based scheduling functions. In DOCOMO's commercial system, the resource utilization rate of the entire system tended to be proportional to the user's usage. Specifically, the utilization rate can easily reach 80% or more during the daytime, when users are most likely to be using the system, but drops to around 20% late at night. In order to use the available resources, the CO administrator registers the container to be deployed, the scheduling method, and the necessary parameters, and CO then deploys the container according to the registered scheduling method. The containers in this case do not require real-time performance and often have a lower priority than other containers, so a mechanism is employed whereby the removal of these containers is prioritized during bursts*41 of cluster resource usage. In this commercial system, a time-based scheduling function is used to aggregate the system logs and update the machine learning models late at night. In addition, when the resources of one cluster are overwhelmed, the resource-based scheduling function automatically distributes some tasks to other clusters to equalize the resources of the entire system instead of temporarily adding resources.

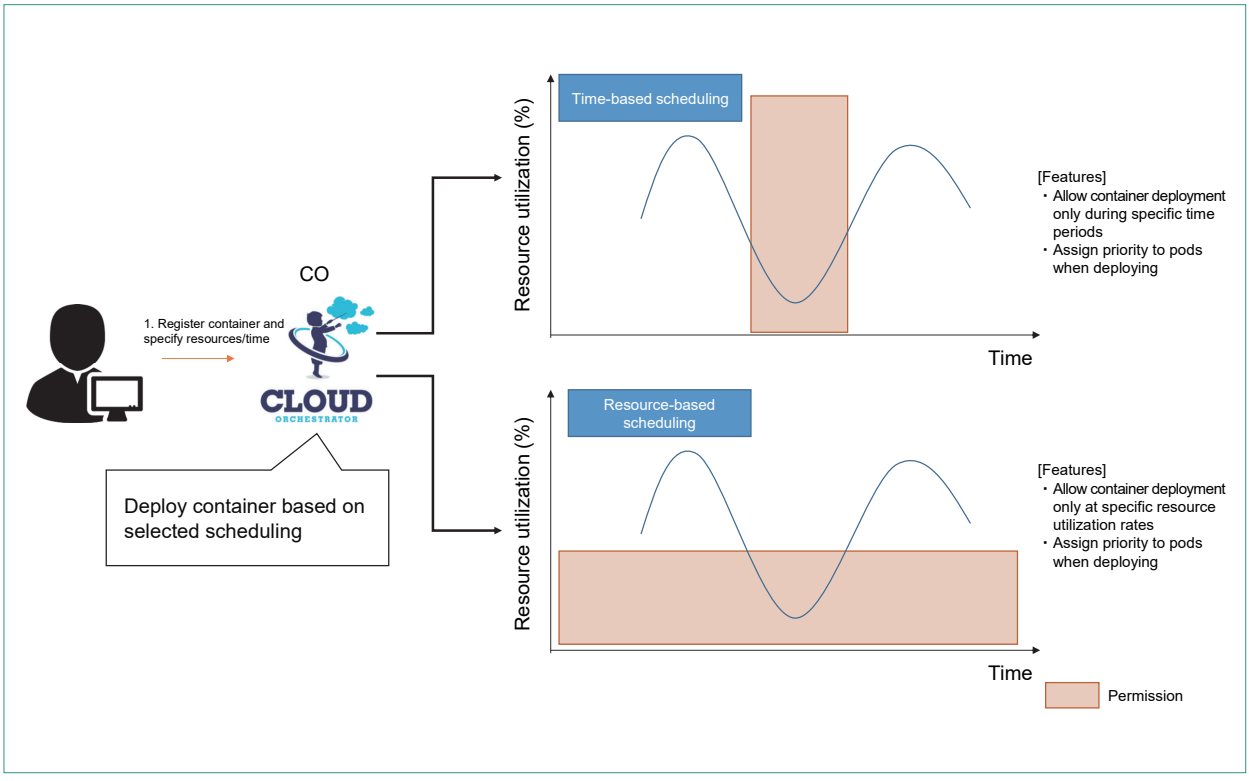


Figure 5 Scheduling function

*41 Burst: A momentary concentration of multiple signals at a fixed point in a device, caused by a temporary increase in network traffic, for example.

4.2 Separation of the Application and System Layers

As shown in **Figure 6** (a), each of DOCOMO's projects creates its own AWS account in which to develop applications and build systems. With this model, every project needs to be aware of AWS, manage its own accounts and system environments, and understand the security policies. As a result, deploying services can take a large amount of time. It also gives rise to various other issues, such as increased security risks due to the absence of security experts and increased cloud costs due to non-global optimization.

To address these issues, we considered separating the application and system layers at the boundary of CO (Fig. 6 (b)). Although this idea is currently only at the conceptual stage, it would eliminate the need for application developers to manage layers below CO and deal with cumbersome security management as long as their services are containerized, allowing them to concentrate their efforts on service development. On the other hand, containerization means that system administrators no longer need to be aware of the content of an application. Centralized management of clusters by system administrators is expected to improve the

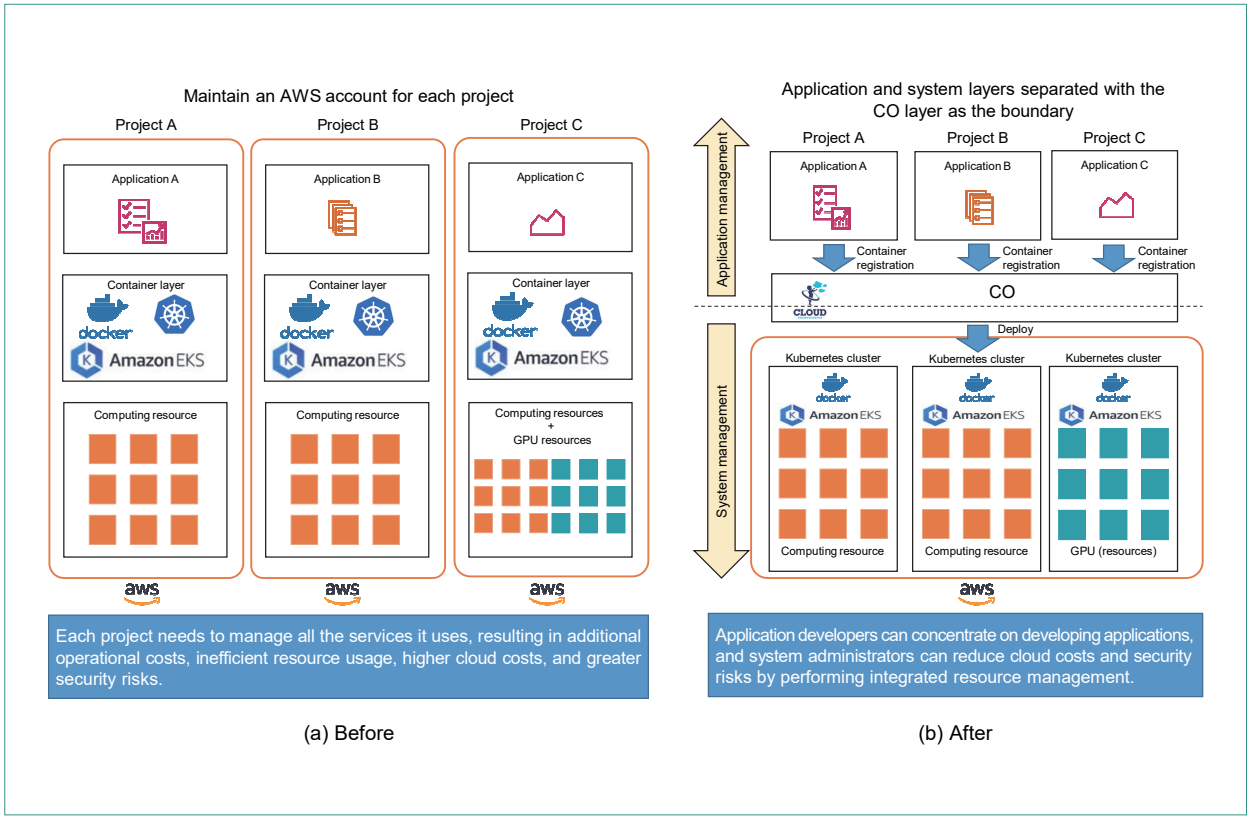


Figure 6 Separation of application and system layers by CO

overall efficiency of resource usage and reduce security risks, ultimately bringing down cloud costs company-wide.

However, with this approach, multiple services can run on the same cluster, and containers that perform heavy processes may affect other services. In addition, application developers will no longer be aware of cloud costs, so they may run processes that are less efficient. In this regard, in addition to using the Resource Quota*42 and Limit Range*43 functions of Kubernetes, CO uses a cost calculation function based on information obtained from the Kubernetes cluster to notify the application developer of fees accrued through the use of resources such as CPUs, memory, and execution pods. This can make application developers more cost-conscious.

We are currently in the process of validating the above method in cooperation with the Service Innovation department. There are not enough functions on the CO side to satisfy this approach, so we will continue to update them.

5. Conclusion

We have described a CO system that manages multiple Kubernetes clusters. We hope to continue to improve CO to make use of the knowledge accumulated by DOCOMO to further streamline the use of cloud services.

REFERENCES

[1] Kubernetes Web site.
https://kubernetes.io/
[2] Drupal: "Cloud."
https://www.drupal.org/project/cloud

*42 Resource Quota: A setting that indicates the amount of resources allocated to a container.
*43 Limit Range: A setting that determines minimum and maximum values with which to limit resource requests from a container.

Technology Reports (Special Articles) Billing Management Cost AWS
Special Articles on Use of Public Clouds

Cloud Cost Optimization Measures

Innovation Management Department Tetsuo Sumiya

Today, public cloud services are extensively used by businesses in Japan and other countries. However, major public cloud systems are billed on a pay-as-you-go basis and if they are deficient in their ability to manage resources such as virtual machines, they can cause unexpected costs by activating resources unnecessarily. It is therefore necessary to consider resource management and cost optimization measures from the initial design stage. This article describes the know-how and cloud cost optimization measures that NTT DOCOMO has cultivated to date.

1. Introduction

In recent years, more and more companies have been using public clouds to provide their services. Unlike the structure of conventional systems in data centers, public cloud services make it easy for users to set up virtual machines*1 and other resources from a management console with just a few clicks. This greatly speeds up system construction and contributes to the enhancement of corporate competitiveness. However, because of their ease of use, many enterprises still struggle to manage and control the costs of public clouds. Major public cloud services are fundamentally billed on a pay-as-you-go basis, and if they are deficient with regard to the management of resources such as virtual machines, they can cause unexpected costs by activating resources unnecessarily. It is therefore necessary to consider resource management and cost optimization measures from the initial design stage.

To optimize costs, it is important to first visualize them to make the effects of cost management measures visible, and to perform repeated cost

*1 Virtual Machine: A computer (e.g., a server) that is implemented virtually in software.

reduction measures and verify their effectiveness. Continuous checks are performed on the usage of resources with visualized costs. Depending on their usage, it is necessary to study the use of payment plans and review the capabilities of the virtual machines being used, such as Elastic Compute Cloud (EC2)*2 instance*3 types in Amazon Web Services (AWS)*4 (**Figure 1**).

This article describes the key concepts of cloud cost optimization initiatives and discusses our specific know-how.

2. Cost Optimization

Cost optimization is considered with the following priorities (**Figure 2**).

STEP 1: Review the constructed architecture

The most effective way to reduce costs is to review the entire architecture and, where possible, to consider using managed services that are available off the shelf from cloud providers rather than building them yourself. Since it is difficult to alter systems during live operation, it is important to design

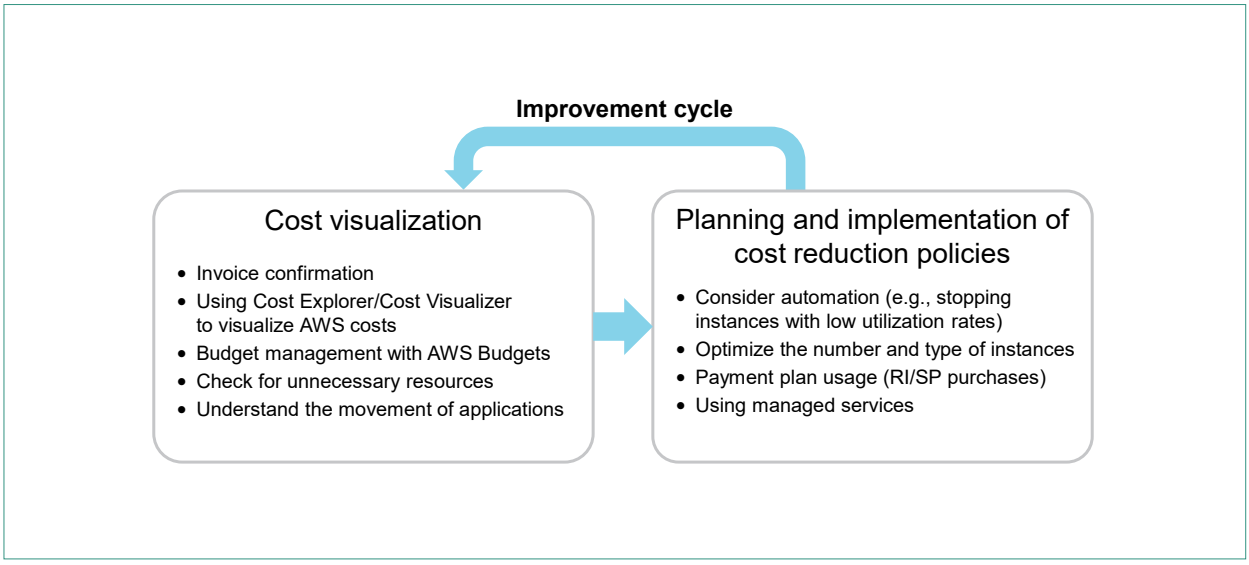


Figure 1 Cost optimization improvement cycle

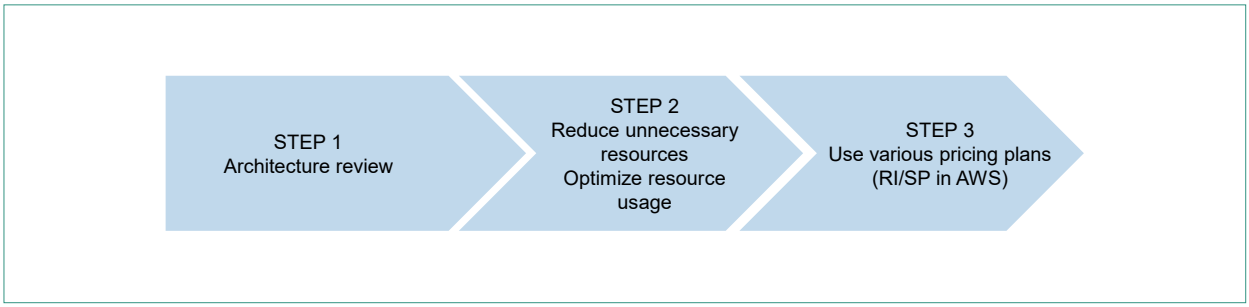


Figure 2 Cost optimization study process

*2 EC2: An IaaS offering provided by AWS. Provides services on virtual machines.
*3 EC2 instance: A virtual machine provided by AWS.
*4 AWS: A cloud computing service provided by Amazon Web Services, Inc.

systems with a firm awareness of costs during the initial design and system renewal stages.

STEP 2: Delete resources that are unnecessary and optimize the use of those that are necessary

When accounts are active for a long time, they sometimes accumulate unused resources, and can end up using excessive resources that were not originally needed. It is possible to reduce costs by incorporating regular resource reviews into operations to ensure that nothing is wasted.

STEP 3: Use various pricing plans

Major public cloud providers offer pricing plans whereby users can reduce their usage fees by committing in advance to a minimum level of resource usage for a specified period. For example, AWS offers pricing plans called Reserved Instance (RI) and Saving Plans (SP) that allow users to lower their fees by committing to one or three years of usage. Google Cloud Platform*5 offers a fixed usage discount, and Microsoft Azure*6 offers a similar pricing plan in the form of Reserved Virtual Machine Instances. If the service is inevitably going to be needed for a certain period of time on an ongoing basis after optimizing resources, this type of pricing plan can be used to reduce costs. However, when committing to use resources, it is difficult to implement major changes to the system configuration and review the resource usage within the period of this commitment, so the usage should be considered after conducting the studies of steps 1 and 2.

3. Cost Visualization

In consideration of the above points, it is essential to have a continuous grasp of the cost structure based on a visualization of the current costs so as to ascertain which parts of the system are incurring costs.

Major public cloud providers offer cost visualization tools. In this section, we will describe the Cost Explorer tool provided by AWS. We will also describe the Cost Visualizer tool, which we developed before the launch of Cost Explorer, and which is used throughout NTT DOCOMO.

3.1 Understanding Usage with Cost Explorer

Cost Explorer is a standard AWS tool that allows users to view a breakdown of their billing status in graphical form (**Figure 3**). This makes it possible to subdivide costs in various ways, such as by service and by member account*7. By default, it can output multiple reports on aspects such as RI/SP utilization and coverage.

As a precaution, Cost Explorer should be used by creating and accessing an Identity and Access Management (IAM) user*8 with only the minimum necessary privileges (e.g., only the ability to view costs).

3.2 Using Cost Visualizer to Ascertain the Usage Status

Cost Visualizer is a cost analysis tool developed and provided by NTT DOCOMO. Since the aforementioned Cost Explorer was not available when NTT DOCOMO began using AWS on a large scale in 2012, we developed Cost Visualizer in-house

*5 Google Cloud Platform: A cloud computing service provided by Google LLC.

*6 Microsoft Azure: A cloud computing service provided by Microsoft Corporation.

*7 Member account: An account that is not a manager account and belongs to an organization that consolidates multiple AWS accounts.

*8 IAM user: A user created with the IAM service who is permitted to access the AWS environment.

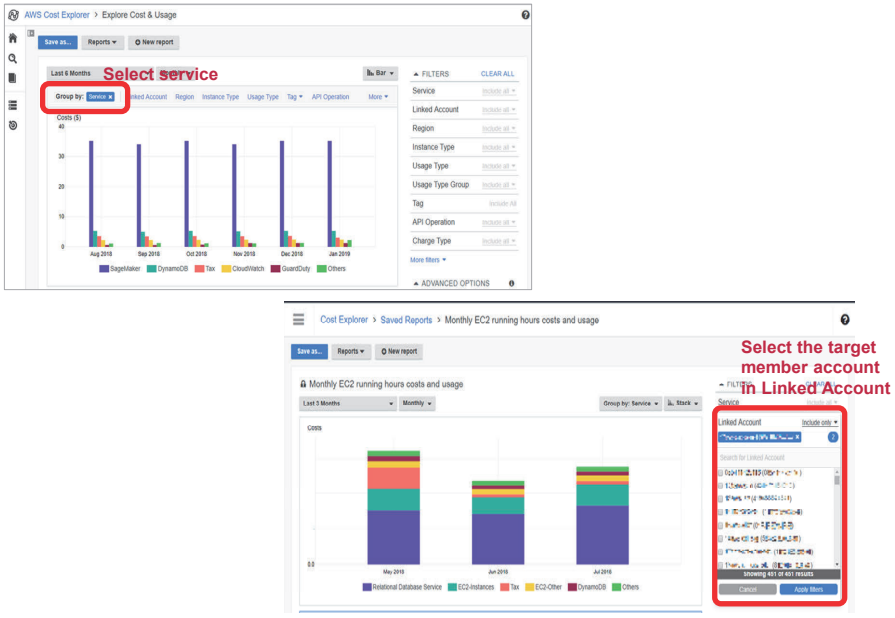


Figure 3 Cost Explorer screenshot

due to the need for cost management.

Cost Visualizer can be used regardless of the AWS support level or billing information access privileges because its privileges are set separately from AWS and the account management is performed separately. It also supports features that are not supported by Cost Explorer, such as the ability to manage privileges in greater detail, and functions for displaying pie charts and data groupings (**Figure 4**).

The system architecture of Cost Visualizer is shown in **Figure 5**. The Cost and Usage Report (CUR) provided by AWS is automatically stored in Amazon Simple Storage Service (Amazon S3)*9. This data is extracted and transformed by AWS Glue*10, which is an ETL (Extract, Transform, Load) service that loads data into a database and makes it available for use, and is then loaded into a database on the virtual machine running Cost Visualizer. We opted to set up a database on a virtual machine instead of using the Relational Database Service managed by AWS because it is internally designed to continuously process queries*11 on large quantities of data so as to minimize the delays until graphs are drawn. Outside the virtual machine, we use a configuration that combines AWS Glue with managed services such as AWS Lambda*12 (a serverless computing platform) and Amazon Dynamo*13 (a key-value store*14 service) in order to reduce costs as much as possible.

*9 Amazon S3: A storage service provided by AWS.
*10 AWS Glue: A PaaS offering provided by AWS. Capable of performing processing for data classification and manipulation.
*11 Query: A database query (processing request).
*12 AWS Lambda: A type of FaaS provided by AWS that provides an execution environment for application code so that the user need only register created source code to run the application.
*13 Amazon Dynamo: A PaaS offering provided by AWS. A highly reliable, high-performance non-relational database service.
*14 Key-value store: A data store with a simple structure that combines keys and values.

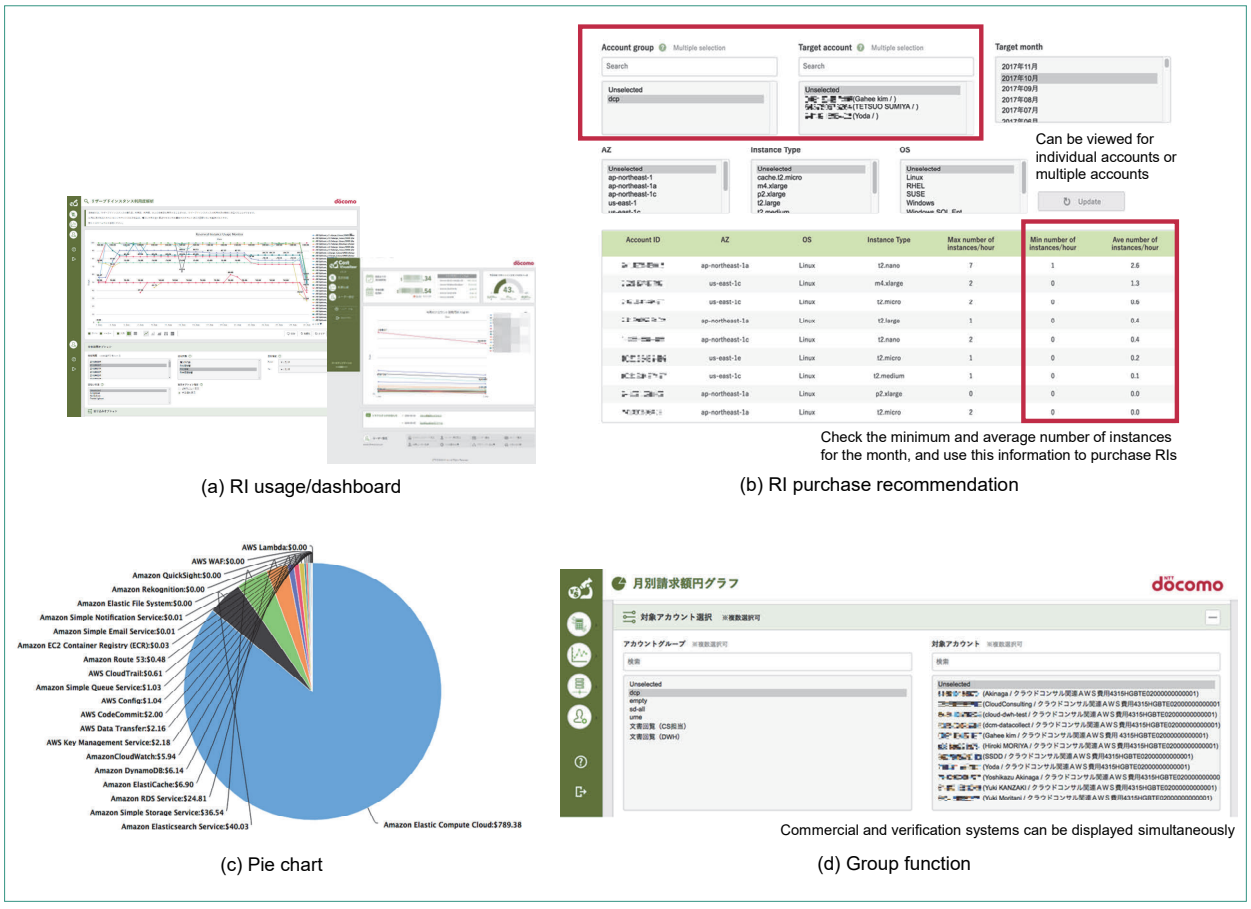


Figure 4 Cost Visualizer screenshot

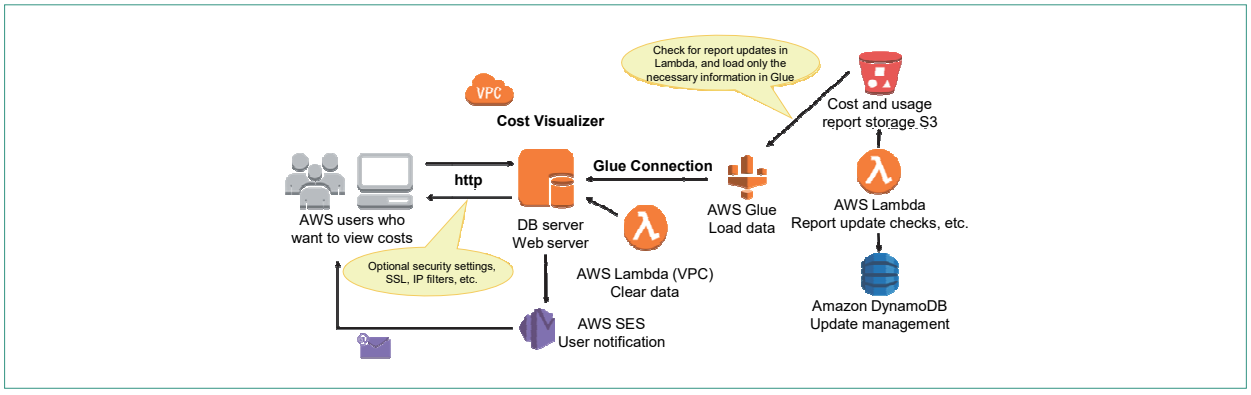


Figure 5 Cost Visualizer system architecture

An example of cost visualization using Cost Visualizer is shown in **Figure 6**. For an architecture centered on virtual machines that does not use managed services, the EC2 costs will dominate in this

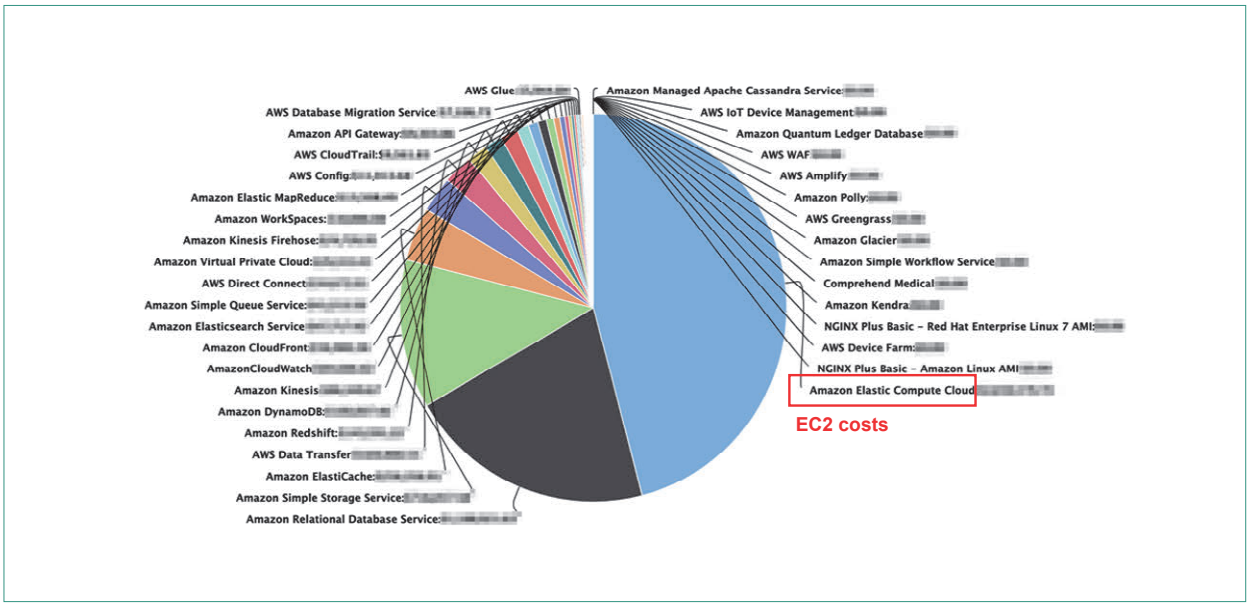


Figure 6 Example of cost visualization by Cost Visualizer

way, indicating that measures to reduce the EC2 costs will be necessary.

3.3 Budget Management Service

Major public cloud providers have budget management services that can send alerts by email or other means when costs or usage have exceeded set limits, or are likely to do so. AWS Budgets allow users to set limits for not only the cost but also for parameters including the quantity of AWS resources used and the RI usage rate. When used together with cost visualization tools, these can enhance the user's everyday cost awareness.

4. Planning and Implementation of Cost Reduction Policies

4.1 Use of Managed Services

Major public cloud providers offer managed services tailored to the characteristics of the processing to be performed, as well as computing resources such as virtual machines. In many cases, these managed services are billed according to the time spent using resources and performing processing, which can cost less than merely provisioning*15 the virtual machines necessary for building such a system.

With AWS, for example, it should be possible to achieve significant cost savings by taking the following steps.

- For services that run for a long time and have few requests, use Lambda instead of EC2.
- When it is necessary to run batch processing*16, consider using AWS Batch*17 instead of EC2. In AWS Batch, computing resources are dynamically scaled*18 according to the volume of batch jobs and the resources they require, thereby reducing costs.
- Switch to a serverless architecture using tools

*15 Provisioning: The process of securing and configuring resources such as servers and networks to run applications.
*16 Batch processing: A processing method where fixed quantities of data are collected and processed all together at fixed intervals.
*17 AWS Batch: A PaaS offering provided by AWS. This service facilitates simple and efficient large-scale batch processing.
*18 Scaling: The optimization of processing power by increasing or decreasing virtual machines that configure communications software whenever processing power is insufficient or excessive according to hardware and virtual machine load conditions.

such as Cognito, API Gateway, Lambda and DynamoDB, as shown in **Figure 7**.

4.2 Identify Unnecessary Resources

Once provisioned, cloud resources incur costs even when they are not actually being used. It is therefore necessary to periodically check whether unnecessary resources are being retained. For example, these resources might include Amazon Elastic Block Store (EBS) volumes[*19] that are not associated with a running EC2 instance, or old EBS snapshots[*20] that are not even tagged.

Although these unnecessary resources can be checked from the console of the cloud service, it can be difficult to do so when dealing with a large number of resources. An efficient way of checking these resources is to use AWS Trusted Advisor. In Trusted Advisor, the items listed in **Table 1** can be checked.

As an example of actual cost reduction, in one project we used Trusted Advisor to check the resource status. As a result, we found that 22 out of 42 block storage devices were not attached. By removing these devices, we were able to reduce the computing cost by about 10%. We also achieved cost savings by deleting over 1,000 untagged snapshots that we found in other projects.

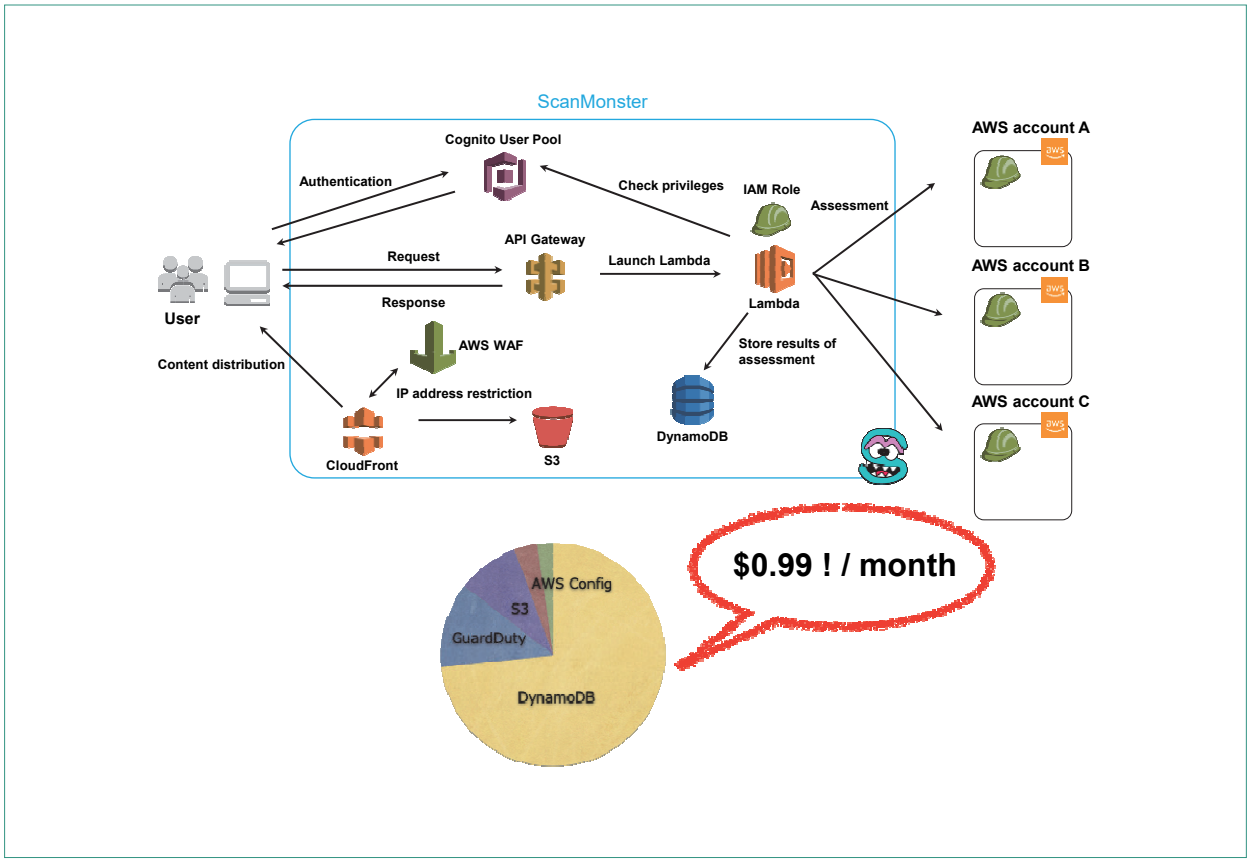


Figure 7 Example of a serverless architecture (ScanMonster)

*19 Amazon EBS Volume: A high-performance, highly available block storage service provided by AWS. Block storage refers to storage in which the recording area is managed by dividing it into units called volumes, and the interior of each volume is further divided into fixed-length units called blocks.

*20 EBS snapshot: Backup data for an Amazon EBS volume.

Table 1 Cost optimization points that can be checked in AWS Trusted Advisor

Item	Overview
Underutilized EC2 instances	Determine usage status based on CPU utilization and network I/O traffic
Idling load balancer	Determine usage status based on the number of requests to the load balancer and the number of associated EC2 instances
Idling RDS DB instance	Determine usage status based on the frequency of connections to RDS DB instances
Infrequently used Amazon EBS volumes	Determine usage status based on whether an EBS volume is not attached to an EC2 instance, or on the frequency of writes
Route 53 latency resource record set	Identify inefficiently configured latency record sets
Underutilized Redshift cluster	Determine usage status based on frequency of cluster connections to Redshift and on CPU usage
Unassociated Elastic IP Address	Check Elastic IP Addresses that are not associated with a running EC2 instance
RI expiration	Check RIs that have expired or will expire in the previous or following 30 days
Amazon EC2 RI optimization	View the recommended number of EC2 RI purchases
Amazon RedShift reserved node optimization	View the recommended number of Red Shift RI purchases
Amazon RDS RI optimization	View the recommended number of RDS RI purchases
SP recommendation	View the recommended number of SP purchases
Amazon ElastiCache reserved node optimization	View the recommended number of ElastiCache RI purchases
Amazon Elasticsearch RI optimization	View the recommended number of Elasticsearch RI purchases

Amazon ElastiCache: A fully managed in-memory data store service provided by AWS.
Amazon Elasticsearch: A managed service provided by AWS based on the Elasticsearch open-source search engine.
Elastic IP Address: A fixed IP address service provided by AWS.
Redshift cluster: A cluster of data warehousing services provided by AWS.
Route 53 latency resource record set: A combination of assets such as domains and EC2 instances that can be registered in Route 53 (a domain name service provided by AWS) to minimize latency from end users.
Load balancer: A device that equalizes the allocation of loads on a server. AWS provides a load balancer as a service.

4.3 Understanding the Movement of Applications

Sometimes, when an application is deployed[*21], it does not generate as much traffic as initially expected and ends up being over-provisioned. It is difficult to shrink resource allocations in on-premises[*22] systems, but in the cloud it is possible to shrink or expand resources as appropriate. Resource usage can be checked using services such as Cloudwatch[*23] for AWS, Cloud Monitoring[*24] for Google Cloud Platform, and Azure Monitor[*25] for Microsoft Azure. In addition to the services provided by cloud providers, there are also monitoring services available from companies such as New Relic and Datadog, and these services can be used to make appropriate changes to resources. AWS has a func-

*21 Deploy: Installing applications by placing them in their execution environments.
*22 On-premises: An environment in which a company owns, maintains, and operates the hardware making up its system.
*23 Cloudwatch: A monitoring service provided by AWS for AWS resources and applications running on AWS.
*24 Cloud Monitoring: A service provided by Google Cloud Platform that monitors Google Cloud Platform resources and the applications running on them.
*25 Azure Monitor: A monitoring service provided by Microsoft Azure for monitoring Azure resources and the applications running on them.

tion called the Compute Optimizer, which can identify idle instances and underutilized instances, and can recommend ways to reduce costs (**Figure 8**).

Also, the latest instance types always tend to be cheaper, so the possibility of switching to the latest instance type should always be kept in mind.

4.4 Considering Automation

In verification environments that do not need to be kept running constantly, costs can be considerably reduced by shutting down overnight and during holidays. For example, by stopping a system for five hours every night on weekdays and altogether at weekends, its running cost can be reduced to 60% or less. If a system can be stopped this much, then it is likely to cost less than the discounted price for a system encumbered with a one-year usage commitment. It is difficult to manually stop a system every day when handling many resources, but this process can be automated to ensure that the system is stopped without fail. Costs can also be reduced by setting up the regular execution of backup scripts and generation management tools to automatically delete old versions.

4.5 Considering Fee Models

After implementing the cost reduction initiatives described above, further cost reductions can be achieved by using fee plans for resources that are absolutely necessary. AWS includes payment plans called RI and SP for computing resources. The RI payment plan makes it possible to reduce fees by committing for a fixed period to a system with specific attributes such as the OS, per-region[*26]/per-Availability Zone (AZ)[*27] deployment, or instance family[*28]. In contrast, SP relaxes these specifications (OS type, per-region/per-AZ, instance family, etc.) and commits the user purely in terms of the

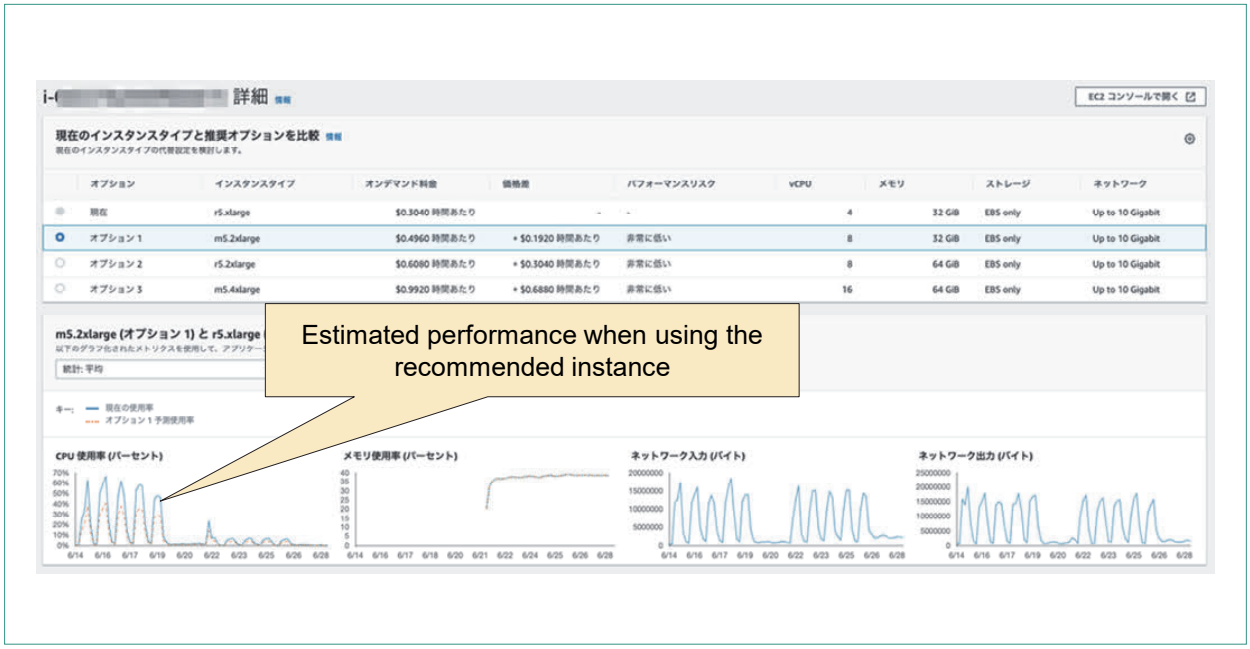


Figure 8 Screenshot of AWS Compute Optimizer

*26 Region: The region in which the data centers providing cloud services are located.

*27 AZ: A collective unit of data centers that are autonomous both physically and in software terms.

*28 Instance family: Instance types are classified by usage, such as "general-purpose," "computing-optimized," and "memory-optimized."

usage fee. As a result, the discount rate is lower for flexible purchases. However, since new instance families are announced from time to time, commitments made based on SP allow for operation with greater flexibility.

5. Conclusion

This article has described the key points of cloud cost optimization and our specific know-how in this field. Since cloud computing systems are billed on a pay-as-you-go basis, proper management of resources and their usage rates is important in terms of cost optimization, while repeated visualization and effectiveness verification is important for management. To optimize costs, it is important to keep cost effectiveness in mind throughout the design process from the initial system configuration, and even during the operational phase. It is also necessary to continue performing periodic checks of the usage status, and to consider the potential for making configuration changes, reviewing instance types, and using different fee plans as dictated by circumstances. In the future, we will consider reducing cloud costs for NTT DOCOMO as a whole through measures such as purchasing SPs with a representative account according to the RI and SP application rates.

Technology Reports (Special Articles) Security Cloud Governance
Special Articles on Use of Public Clouds

Development of a Security Checking Tool For Public Clouds

Innovation Management Department Takuya Nakamura

With the accelerating use of public cloud services around the world, many companies are now using them to run their services and mission-critical system workloads. Public clouds are sometimes used to handle highly sensitive information, making security a very important issue.

This article describes the basic concept of security in public clouds, and discusses the measures and approaches taken by NTT DOCOMO in this regard.

1. Introduction

As of 2021, many companies around the world have started using public cloud*1 platforms such as Amazon Web Services (AWS)*2, Microsoft Azure*3 (hereinafter referred to as "Azure"), and Google Cloud Platform (GCP)*4 to run their services and mission-critical system workloads*5. These platforms are also sometimes used to handle highly sensitive information, making security a very important issue.

However, information incidents including large-scale data breaches and service outages have actually occurred in systems built using public clouds. In one particularly serious case that drew a lot of attention, the personal information of over a hundred million people was leaked from a system built by Capital One in the United States [1]. Many of these incidents would not have occurred had it not been for the use of public clouds. However, these platforms offer many advantages, such as the ability to conduct business at a much faster speed. In 2021, it is no longer tenable for corporate management to choose not to use public cloud platforms for security reasons alone.

*1 Public clouds: Cloud computing services that anyone can use over the Internet.
*2 AWS: A cloud computing service provided by Amazon Web Services, Inc.
*3 Microsoft Azure: A cloud computing service provided by Microsoft Corporation.
*4 GCP: A cloud computing service provided by Google LLC.

NTT DOCOMO has been using public clouds since around 2009. As of 2021, we have gained extensive experience and know-how of using them to run workloads securely. In this article, we describe the basic concepts of security in public clouds. We also introduce NTT DOCOMO's security countermeasures and some examples of efforts we have made to prevent security incidents, and we present the features and configuration of ScanMonster, a security checking tool that we developed based on these efforts.

2. Cloud Security Concepts

Before public clouds became popular, companies often had to prepare their own data centers and servers. This implementation style is now called "on-premises." The biggest difference between on-premises and public cloud platforms is that the former is implemented with servers that the user is entirely responsible for managing, while in the latter, the user does not even need to know about the existence of the data center. In order to successfully use public clouds built on servers that are not directly visible and cannot be directly managed, it is absolutely essential to understand the basic concept of the shared responsibility model, which is described below (in the case of AWS, see [2]).

2.1 Shared Responsibility Model

In a public cloud, the cloud provider operates, manages, and controls various system components, ranging from the host operating system and virtualization layer to the physical security of the facility where the service is operated. This has the benefit of reducing the operational burden on the user, while leaving the cloud provider in charge of providing the service.

However, no matter how many security measures are implemented by the cloud provider, very dangerous situations can still arise depending on the settings made by the user. It is therefore necessary for cloud providers and their users to establish a clear division of responsibilities and to cooperate in the implementation of security measures. This way of thinking is called the shared responsibility model.

In the shared responsibility model of a public cloud, the responsibilities apportioned to users and cloud providers differ depending on the type of service (**Figure 1**). In the case of so-called Infrastructure as a Service (IaaS)[*6] services that provide virtual machines[*7], the cloud provider is mainly responsible for managing physical facilities such as data centers, hardware such as physical machines and networks, and the host OS and virtualization layer. On the other hand, the user is responsible for managing the guest OS and the applications running on it. In managed services[*8] such as Platform as a Service (PaaS)[*9] and Function as a Service (FaaS)[*10], the user has a narrower scope of responsibility, and a greater share of the management responsibilities can be left to the cloud provider. By making appropriate use of such services, a company can concentrate its resources on value-added applications and business areas.

2.2 Cloud Provider Evaluation

In the abovementioned shared responsibility model, users not only have to share the responsibility

*5 Workload: An indicator of the size of a system's load, such as the CPU utilization rate. In particular, in a public cloud environment, the workload may represent the system itself, including the OS and application code running on the cloud. In this paper, we use the term in this latter sense.

*6 IaaS: A service in which hardware such as servers and networks is rented out virtually to users who use this hardware to install and run their own OS and application software.

*7 Virtual machine: Computers such as servers constructed in a virtual manner by software.

*8 Managed service: Cloud services whose resource provisioning, operation, etc. are mostly the responsibility of the cloud operator. Among cloud computing services, these are referred to as PaaS and SaaS, for example.

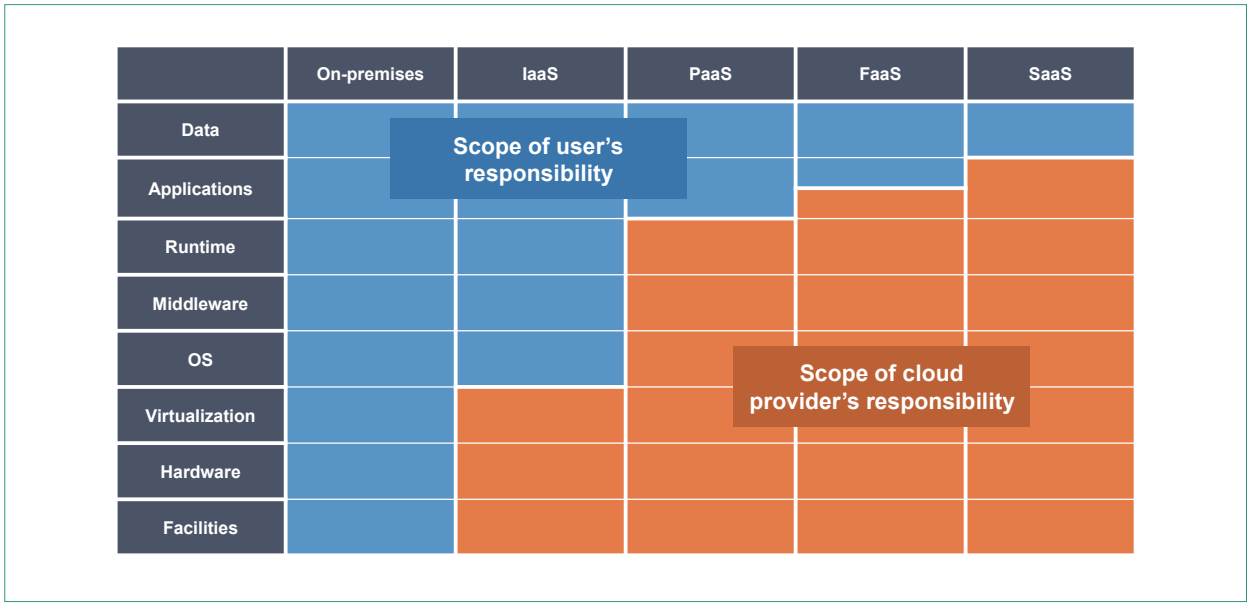


Figure 1 Differences in the scope of responsibility for different types of public cloud service

with cloud providers, but conversely they also have to entrust certain aspects of the service to them, and must therefore be able to trust that the cloud providers can fulfill their responsibilities. But is this sharing of responsibilities really appropriate? For example, concerns might be raised about a cloud provider's ability to provide a stable service, or to guarantee security, or whether data uploaded by users to public clouds is immune to illicit internal access. So how is it possible to evaluate the trustworthiness of a cloud provider?

Evaluation methods for public clouds and cloud providers need to draw a broad distinction between functional and non-functional requirements.

With regard to functional requirements, it is essential to perform preliminary checks on the documents, white papers, and Service Level Agreements (SLAs)[*11] provided by cloud providers. In particular, items such as service availability and performance are often explicitly provided as numerical values in the documentation, allowing them to be evaluated against specific system requirements.

It is also effective to use certificates issued by public institutions and compliance reports issued by audit firms as important decision-making materials for the evaluation of non-functional requirements. Some of these documents are made available to the public by cloud providers, while others have to be obtained through individual non-disclosure agreements. Many cloud providers have been audited by international organizations, including those listed below, and their certifications and reports are available.

- International Organization for Standardization (ISO)[*12] 27017: 2015 Certification
- Payment Card Industry Data Security Standard (PCI DSS)[*13] Attestation of Compliance (AOC)[*14] and Responsibility Summary

*9 PaaS: A service that lends out a platform including an OS and middleware for running applications on the cloud. The user creates and uses application software on the borrowed platform.
*10 FaaS: An event-driven application execution service. Since there is no need to manage resources, users can concentrate on writing code. Services of this sort are generally billed according to the time it takes to execute functions.
*11 SLA: A guarantee of the quality of a provided service.
*12 ISO: A organization that sets international standards in the field of information technology plus all other industrial sectors except electricity and telecommunications.
*13 PCI DSS: A credit card security standard established to protect cardholder details and transaction information.

- Service Organization Controls (SOC)*15 2 Report

These reports also help to determine whether or not the cloud provider in question meets the security standards of various industries. For example, in the financial sector, the Center for Financial Industry Information Systems (FISC) has set out security standards, and many cloud providers have published details on how they are complying with these standards.

In recent years, the EU has also been actively passing laws and regulations to protect the privacy of users, such as the General Data Protection Regulation (GDPR). To implement public cloud services that comply with the laws and regulations of each region or country, public cloud providers use separate physical data centers for each region (or country). This makes it possible for them to provide clear indications of data residency*16 and offer services that are customized to meet the legal compliance requirements of each country/region.

2.3 Measures to be Taken by Users

The measures to be taken by users in public clouds are basically the same as those taken by the users of on-premises systems. For example, they should manage latent vulnerabilities in software and encrypt communications, and when using IaaS services, they should also ensure that security patches are applied to guest operating systems, and set up firewalls to prevent attacks at the network level.

Incidentally, do public clouds have any other advantages apart from the fact that the maintenance and operation of physical data centers and servers is taken care of by the service provider? One of the main factors behind the widespread use of public clouds is their PaaS mode of operation. With PaaS, there is no need for the user to manage the OS, which is necessary with IaaS. Furthermore, in PaaS, and especially FaaS where users only need to their own code, there is no need for users to take responsibility for the management of middleware*17. Therefore, the reduced management obligations of the user compared with on-premises (or IaaS) solutions means that the user bears less of the responsibility for implementing security measures. In other words, the use of PaaS and FaaS has the effect of reducing the user's exposure to security risks.

On the other hand, there have been some incidents that could only have happened in cloud services. In the cloud, all infrastructure*18 is implemented in software. In the past, setting up infrastructure involved entering a server room and disconnecting and reconnecting LAN cables to provide an external network connection, but nowadays this can be done at the push of a button. Although businesses can benefit from this ability to make quick and decisive infrastructure changes from a maintenance terminal with such ease, this also raises major concerns with regard to security.

Cloud providers offer a number of services that are useful for implementing security measures within the user's realm of responsibility. For example, AWS publishes a collection of best practices called the AWS Well Architected Framework*19. It is also very important to use tools such as AWS Trusted Advisor and Security Hub to check whether a system is being operated in accordance with best

*14 AOC: A certificate that indicates a service provider's compliance with the PCI DSS standard.
*15 SOC: Security standards developed by the American Institute of Certified Public Accountants. Also called System & Organization Controls.
*16 Data residency: The location where data is stored.
*17 Middleware: Software providing functions for common use by multiple applications.
*18 Infrastructure: A generic term for the physical or virtual data centers, servers, networks and other equipment needed to run an application.
*19 AWS Well Architected Framework: A set of best practices for design and operation published by AWS.

practices in the realm of user responsibility.

3. Security Controls at NTT DOCOMO

This section describes NTT DOCOMO's security control concepts and systems.

3.1 Security Control Systems

Our security control systems for cloud operations are shown in **Figure 2**. In this figure, the Information Security Department is a company-wide organization that creates and manages security policies, and can also examine individual systems and offer advice on whether they meet these security policies.

NTT DOCOMO owns many of the facilities that make up its communications network and provide communications services. It also still operates many other workloads on-premises. A strict security policy is applied to the execution of these workloads to avoid security incidents. However, the use of a public cloud does not make the security policy any less strict. Instead, a general security policy is established for all systems built within the company, regardless of whether or not they use a public cloud. Therefore, when using a public cloud, it is important to understand particular mechanisms such as the shared responsibility model, and to apply security measures according to their characteristics.

3.2 CCoE Roles and Challenges

As mentioned above, specific concepts and measures must be implemented when using a public cloud. For this reason, NTT DOCOMO established

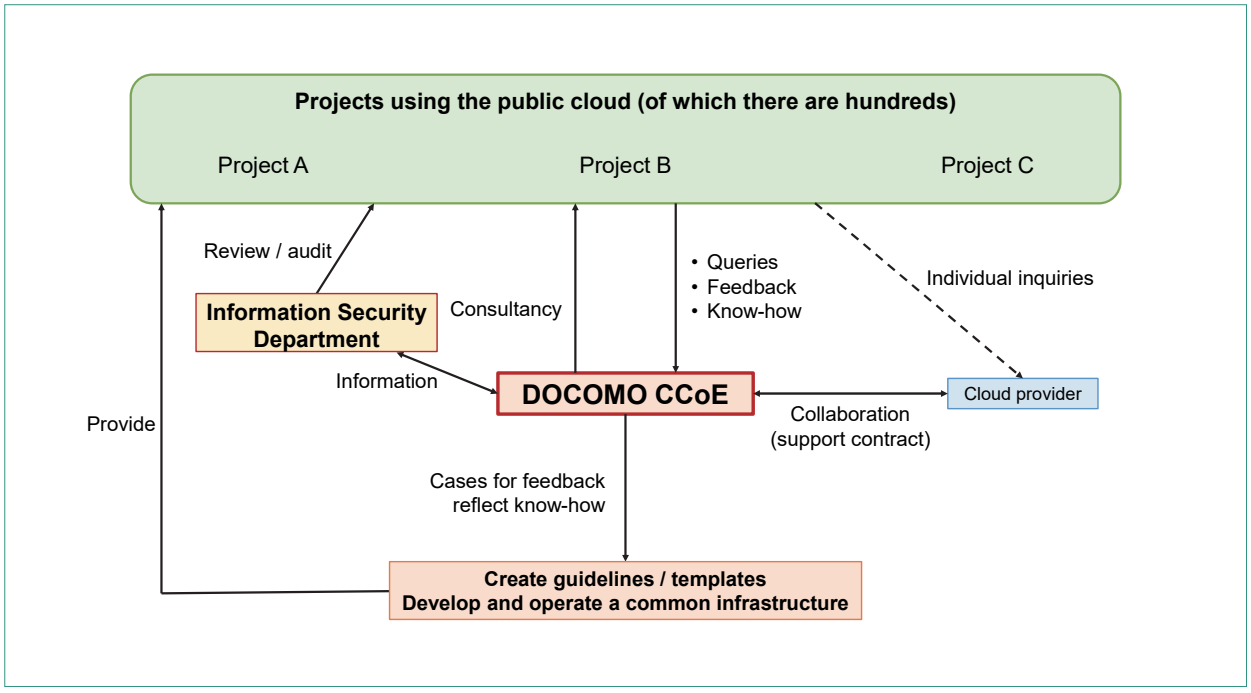


Figure 2 NTT DOCOMO's internal cloud control system

a unit called Cloud Center of Excellence (CCoE), which specializes in supporting the users of cloud computing. This unit shares knowledge with the Information Security Department, provides know-how for each project, and offers consulting on architecture and security.

However, in recent years, the number of systems and projects using the public cloud has grown rapidly, making it difficult for CCoE to keep track of the configuration of every system. Furthermore, although one of the reasons for using the public cloud is to accelerate business, the tightening of control by the Information Security Department and CCoE prevents each project from bringing services to market quickly and adding/modifying functionality with short cycle times. It has therefore become necessary for the members of each project to grasp the business requirements and system configuration by themselves and to make their own judgement of security risks. For this reason, CCoE provides the DOCOMO Cloud Package (a cloud-related knowledge base), and ScanMonster (a security checking tool).

The activities of CCoE are described in more detail elsewhere [3].

4. Visualization and Checking Tools

At the start of each project, the developers first use the DOCOMO Cloud Package to acquire know-how relating to the public cloud in particular, which they can then use to build the system. However, when we conducted interviews relating to each project, many developers said they did not know which know-how to apply, or whether the constructed system would be able to meet policy requirements. To address these issues, CCoE developed ScanMonster, which is a tool that automatically assesses AWS environments.

4.1 ScanMonster Functions

ScanMonster makes it possible to assess an AWS environment in over 70 different ways. Items for assessment are created by referring to the contents of the DOCOMO Cloud Package, the Center for Internet Security (CIS) Benchmarks*20, the AWS Well Architected Framework, and several other indicators. For example, with the Root Account MFA assessment item, it is possible to check for AWS accounts where Multi-Factor Authentication (MFA)*21 has not been set up for the root user*22. The results of this check can be used to encourage users to set up MFA for the root user if they have not already done so. The ACM Validation Method assessment item checks whether the Domain Name System (DNS)*23 is used for domain validation when issuing certificates in AWS Certificate Manager*24. This is an original item that is based on the contents of the DOCOMO Cloud Package and makes use of NTT DOCOMO's public cloud know-how.

Figure 3 shows a screenshot of ScanMonster in operation. Users can select any of the assessment items and run them at the press of a button. The results are shown as either ○ for success or × for failure, so the user can grasp the assessment results at a glance. It is possible to run multiple assessment items, even all of them, at the same time. It is also possible to run simultaneous assessments by selecting multiple accounts from among

*20 CIS Benchmark: A security standard developed by CIS in the United States.
*21 MFA: Multi-factor authentication. An authentication method where a user's identity must be verified with multiple types of evidence (factors).
*22 Root user: A sign-in identity that has complete access to an AWS account. It is considered a best practice to protect this user with strict security by setting up MFA.
*23 DNS: A mechanism that manages the mapping of domain names and IP addresses on the Internet and provides services for converting between them.
*24 AWS Certificate Manager: A service that simplifies the issuance and management of SSL/TLS certificates for use with AWS services.

the assessable AWS accounts configured in advance for each user.

For many of these assessment items, tutorials have been prepared to describe the purpose of each item, its pass/fail conditions, and procedure for fixing failed assessments. A screenshot of a tutorial display in ScanMonster is shown in **Figure 4**. These tutorials make it easy for AWS novices to understand assessment results, estimate business risks, and decide whether to take remedial action.

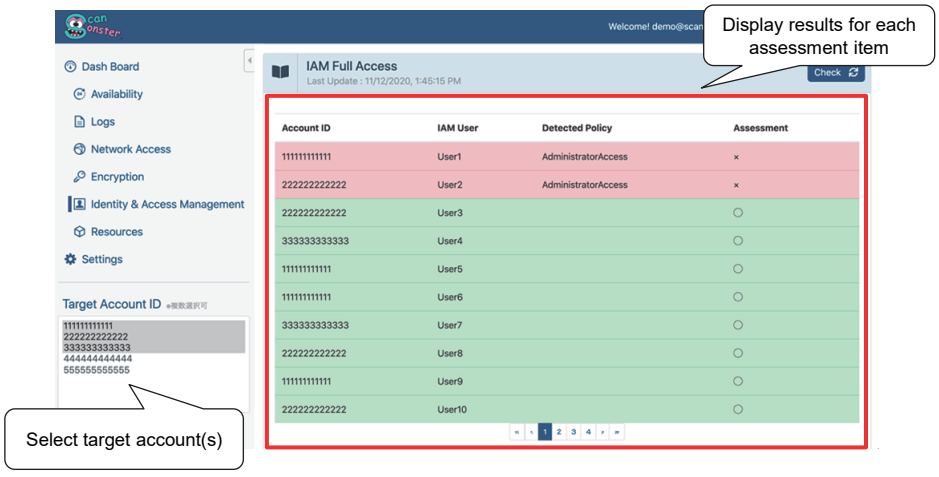


Figure 3　Screenshot of ScanMonster while running an assessment

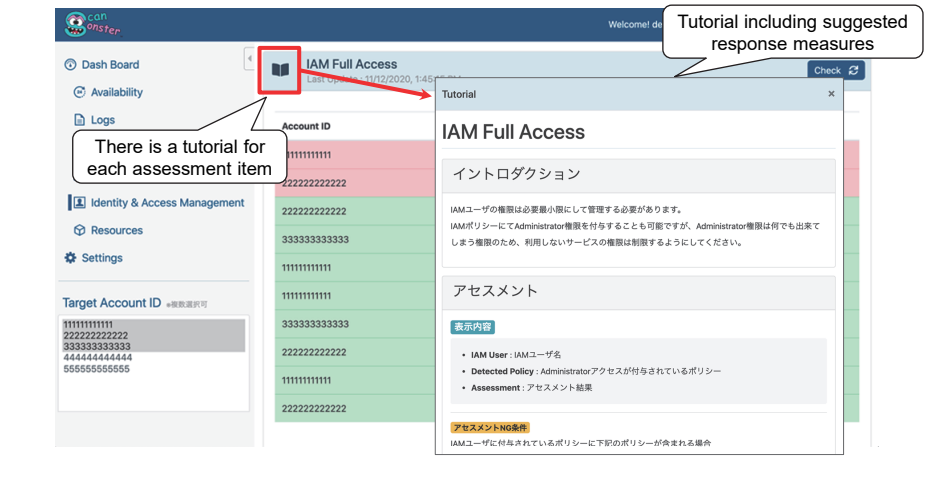


Figure 4　Screenshot of a tutorial display in ScanMonster

4.2 ScanMonster Configuration

1) Serverless Architecture

As shown in **Figure 5**, ScanMonster is itself built on AWS. Instead of configuring components from IaaS products such as Amazon EC2, it uses services such as AWS Lambda*25, Amazon S3*26, Amazon DynamoDB*27, Amazon Cognito*28, Amazon CloudFront*29 and AWS Web Application Firewall (AWS WAF)*30 to create a serverless architecture.

A serverless architecture has two main advantages.

The first is that, just as the name implies, there is no server and therefore no need for infrastructure management. Since all services are managed services, there is no need to monitor operational status of servers or provision*31 them according to load.

The second is that the operating cost is very low. The only parts that are billed based on the duration of use are AWS WAF, which is used for controlling access to the end-points and to Amazon S3 (which stores the front-end data), and Amazon Cognito, which is used for setting the user management and access permissions. For AWS Lambda and Amazon DynamoDB, since usage fees are only charged when assessments are performed, no fees are charged during periods when assessments are not being performed, even while ScanMonster is still in use, or during periods such as late at night when no one is accessing ScanMonster. The actual monthly cost at NTT DOCOMO ranges from a few tens of yen to a few hundred yen (**Figure 6**).

2) Cross-account Access for Assessment of Multiple AWS Accounts

Simultaneous assessment of multiple AWS accounts is performed with cross-account access by

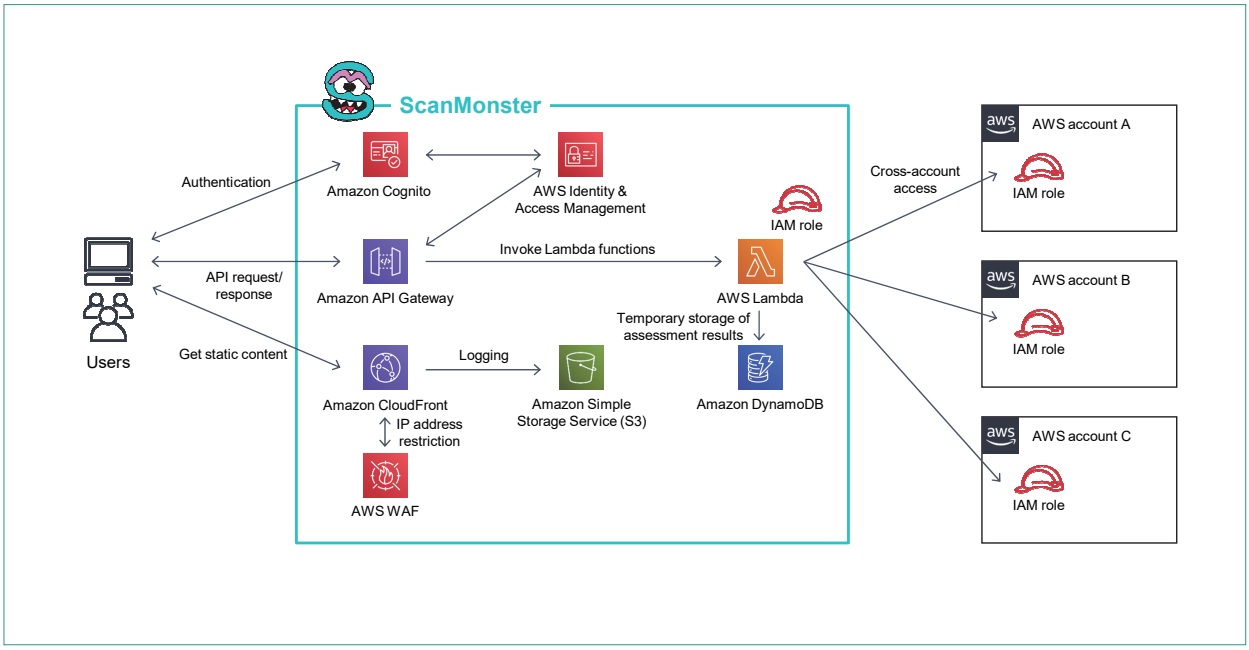


Figure 5 ScanMonster architecture diagram

*25 AWS Lambda: A type of FaaS provided by AWS that provides an execution environment for application code so that the user need only register created source code to run the application.

*26 Amazon S3: An object storage service provided by AWS. Designed to provide 99.999999999% data durability.

*27 Amazon DynamoDB: A NoSQL database service provided by AWS. It is designed to be able to handle large numbers of requests with low latency.

*28 Amazon Cognito: A PaaS offering provided by AWS. It provides authentication, authorization, and user management functions for web and mobile applications.

*29 Amazon CloudFront: A Content Delivery Network (CDN) service provided by AWS.

*30 AWS WAF: A firewall service for web applications provided by AWS.

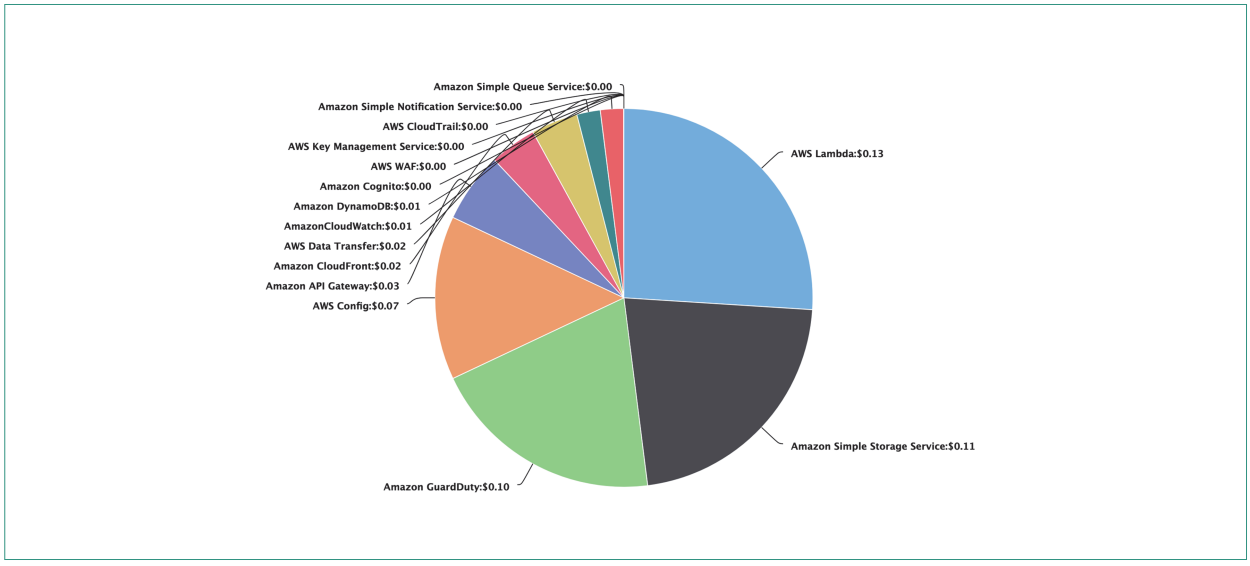


Figure 6 ScanMonster cost performance chart

assuming an Identity and Access Management (IAM) role*32. In the AWS account to be assessed, grant access to ScanMonster by setting up an IAM role that trusts the AWS account that deployed ScanMonster. This ensures that assessments are only accepted by AWS accounts that have explicitly granted access. The permissions of the IAM role are restricted to the minimum read permissions needed to perform assessments, so as to prevent unauthorized attacks from the AWS account where ScanMonster is deployed. Furthermore, if Amazon Cognito users are also managed within ScanMonster, it will also be possible to set up AWS accounts that can be accessed by each user.

3) Simple ScanMonster deployment using Infrastructure as Code (IaC)*33

All the AWS resources from which ScanMonster is built are written using AWS CloudFormation*34 templates, allowing it to be deployed instantly. This makes it possible to automatically create a new ScanMonster environment every time testing and quality control is performed during application development. ScanMonster is also a product offered to external customers, which makes it easy for customers to deploy ScanMonster within their own AWS accounts. This allows them to store and use highly confidential security information such as the assessment results of AWS environments without disclosing it to NTT DOCOMO.

5. Conclusion

We have described the basic concepts of cloud security and the efforts that NTT DOCOMO is making with regard to security control. Public cloud technology is still evolving, and the concepts of cloud security and attack methods in the wild are in constant flux. The important thing is that once a system has been created on a public cloud, it is not the end of the story. Security assessments

*31 Provisioning: The process of securing and configuring resources such as servers and networks to run applications.
*32 IAM role: One of the identity resources in AWS. It is used to delegate access rights to AWS resources to any authorized user, application or service.
*33 IaC: The process of describing and managing the configuration of infrastructure such as servers, networks, and storage by means of code statements. It can be used to automate configuration and provisioning tasks.
*34 AWS CloudFormation: An AWS service offering that implements IaC by providing provisioning and management functions for AWS resources described in a template.

should still be conducted on a daily basis, and the contents of assessments should be updated every day to keep up with the times. To perform updates, it is essential not only to use the services provided by cloud providers themselves, but also to create and operate organizations such as CCoE that are dedicated to promoting cloud use and controlling security in order to keep up with the rapid evolution of public cloud services.

NTT DOCOMO developed ScanMonster as a tool that enables autonomous security assessments of numerous internal projects. By using these tools together with a set of guidelines, we have established a system that allows public cloud environments to be used safely without restricting the speed of business. We are considering expanding ScanMonster in the future to include functions for customizing the details of assessments, linking assessments with guidelines, and providing support for multiple cloud systems including GCP and Azure in addition to AWS. We also aim to make improvements to our internal structures and security policies, such as consolidating the assessment results at the Information Security Department and studying mechanisms for performing assessments on a regular basis.

REFERENCES

[1] Bloomberg: "Capital One Says Breach Hit 100 Million Individuals in U.S.," Jul. 2019. https://www.bloomberg.com/news/articles/2019-07-29/capitalone-data-systems-breached-by-seattle-woman-u-s-says

[2] AWS: "Shared Responsibility Model." https://aws.amazon.com/compliance/shared-responsibility-model/

[3] H. Moriya, et al.: "NTT DOCOMO's Use of Public Clouds and Role of CCoE," NTT DOCOMO Technical Journal, Vol.23, No.1, pp.4–12, Jul. 2021.

Technology Reports (Special Articles) Public Clouds Operations Failure
Special Articles on Use of Public Clouds

System Operations on Public Clouds to Provide against Large-scale Failures

Innovation Management Department Hiroki Moriya

Public clouds as typified by AWS are being increasingly used not only by private enterprises but also by many organizations and groups such as government agencies and educational institutions. In this way, public clouds are becoming a social infrastructure, so the shutdown of a public cloud can have a major impact on society overall. As a result, system operations that make use of clouds are becoming increasingly important. NTT DOCOMO has been using public clouds on a large scale for many years and has accumulated system-design and operation know-how assuming the possibility of a large-scale failure. It has also developed tools that enable smooth information sharing within the company in the event of a large-scale failure. It has consequently become possible for NTT DOCOMO overall to perform system operations that make provisions for the large-scale failure of a public cloud.

1. Introduction

It's already been more than ten years since public clouds including Amazon Web Services (AWS)*1 began to flourish, and since then, many organizations and groups such as government offices and educational institutions in addition to private enterprises have undertaken the construction and operation of systems that use public clouds*2.

Public clouds are no longer simple cloud services

*1 AWS: A cloud computing service provided by Amazon Web Services, Inc.
*2 Public clouds: Cloud computing services that anyone can use over the Internet.

as they have come to function as an infrastructure supporting society overall. Once a large-scale failure occurs in a public cloud, a situation arises in which that failure can have a major impact throughout society.

NTT DOCOMO has experience in the long-term use of large-scale public clouds and has used that experience to accumulate know-how on system operations that leverage the features of public clouds and to develop tools that support those operations. In this way, we have been active in extending the use of public clouds throughout the company. This article describes these activities with the aim of providing a reference in the use of public clouds even for organizations outside NTT DOCOMO.

2. Approach to System Failures

2.1 High-availability System Configurations

Failures are inherent to systems. A system that never breaks down or a perfect system that suffers no failures does not exist. This principle applies to both systems constructed on-premise*3 and systems constructed on public clouds. It is therefore important when constructing a system to give it a configuration that can restore system-wide availability even if only slightly in the event that a failure occurs in an element making up the system. For example, in the case that a single piece of equipment such as a server stops functioning, a technique long used to handle such a problem is to maintain system functions by switching to backup equipment. Likewise, in the application domain, many measures have been implemented to enhance the fault tolerance of the entire system. These include the adoption of a loosely coupled system through asynchronous processing that limits the range of failure impact and the use of retry processing that assumes the occurrence of failures. Additionally, in contrast to configuring a system as a single monolithic*4 service, the trend in recent years has been to construct a system by dividing it into multiple microservices*5 and to improve availability of the entire system by limiting the range of impact when a failure occurs.

2.2 Shared Responsibility Model on Public Clouds

The "shared responsibility model [1]" has been widely adopted in the use of public clouds. This model clarifies the range of responsibility of both the cloud user and the public cloud operator and aims to secure the availability of the entire system through the activities of both parties. For example, in the case of virtual machines*6, the range of responsibility of the public cloud operator is generally the area up to the virtualization layer including the data center, physical servers, and networks. The operator therefore implements measures to maintain availability such as incorporating redundancy in infrastructure*7 components, physical equipment, networks, etc. On the other hand, it is the responsibility of the cloud user to implement a virtual machine OS and applications running on that OS in such a way that maintains availability. This example concerns the use of virtual machines, but cloud services other than Infrastructure as a Service (IaaS)*8 have become popular in recent years

*3 On-premise: An environment in which a company owns, maintains, and operates the hardware making up its system.
*4 Monolithic: A configuration that provides multiple functions as a single and large software unit.
*5 Microservices: A system development technique that creates a single service by combining a number of small services.
*6 Virtual machines: Computers such as servers constructed in a virtual manner by software.
*7 Infrastructure: Generic term for physical or virtual data centers, servers, networks, etc. for executing applications.
*8 IaaS: A service that virtually lends out hardware such as servers and networks. The user sets and runs an OS and application software on the borrowed servers or network.

such as Platform as a Service (PaaS)[*9] and Software as a Service (SaaS)[*10]. But the concept of the shared responsibility model is equally applicable to these services if only because the range of responsibility differs between the user and public cloud operator.

Many public cloud operators provide functions to support user designs that maintain system availability. They are also committed to disseminating information in the form of documents and white papers covering commonly used configuration patterns as best practices. With these functions, it has become possible for users to construct systems for efficiently achieving high availability. Moreover, since configurations that achieve high availability are often already in use by public cloud operators for cloud services like PaaS and SaaS, users have come to favor such services for constructing their systems.

3. Recent Large-scale Failures on the Cloud Operator Side

The following provides some examples of large-scale failures in public clouds that have recently occurred.

1) AWS Failure

At AWS, on August 23, 2019, a failure in the control system for air conditioning equipment prevented the cooling system from operating normally causing some servers to overheat. This, in turn, caused failures to occur in a single Availability Zone[*11] and specifically in Amazon Elastic Compute Cloud (EC2)[*12] and Amazon Elastic Block Store (EBS)[*13]. Additionally, as EC2 and EBS are used as platform services for configuring other services, it was found that Amazon Relational Database Service (RDS)[*14], Amazon Redshift[*15], Amazon ElastiCache[*16], Amazon WorkSpaces[*17], and other managed services[*18] were also affected. According to some cloud users, there were some cases in which adopting a redundant configuration across multiple Availability Zones beforehand was able to minimize the impact of this failure that occurred in a single Availability Zone, but some effects were nevertheless reported for some configuration settings.

Then, on April 20, 2020, failures such as an increase in processing errors and delays also occurred in the Tokyo Region in managed services such as Amazon Simple Queue Service (SQS)[*19] and AWS Lambda[*20]. It was confirmed that users of these services were impacted by these failures [2].

2) Microsoft Azure Failure

At Microsoft Azure[*21], authentication errors in Azure Active Directory (Azure AD) were observed around the world on September 29, 2020. Azure AD is a core service of Microsoft Azure for managing authentication and permissions in many services for developer and user access and inter-service operations. It was found that this failure impacted users by preventing them from using some services or applications such as Microsoft Azure and Office 365. As for the cause of this failure, it was announced that an update to an internal validation test was supposed to undergo deployment[*22] only after multilayer testing. However, this update actually bypassed the testing process owing to a latent bug in the Safe Deployment Process (SDP)

*9 PaaS: A service that lends out a platform including an OS and middleware for running applications on the cloud. The user creates and uses application software on the borrowed platform.
*10 SaaS: A service that lends out applications on the cloud. The user can start using those applications immediately.
*11 Availability Zone: A single or group of data centers. Individual Availability Zones are physically independent of each other.
*12 EC2: A type of IaaS offered by AWS providing virtual machines.
*13 EBS: A type of IaaS offered by AWS providing block storage.
*14 RDS: A type of PaaS offered by AWS providing relational database functions.
*15 Amazon Redshift: A type of PaaS offered by AWS providing a data warehouse.
*16 Amazon ElastiCache: A type of PaaS offered by AWS providing in-memory cache.
*17 Amazon WorkSpaces: A type of SaaS offered by AWS providing Windows or Linux desktop environments.

system[*23] and was directly deployed into the production environment as a result [3].

3) GCP Failure

A failure occurred on Google Cloud Platform (GCP)[*24] on March 27, 2020 in Cloud Identity and Access Management (Cloud IAM), an access control and management service. Cloud IAM is a service used in common by many GCP services for access control, so this failure had a large-scale impact on other services. The official announcement stated that the effects of the failure lasted for a total of 14 hours. Its cause, as announced, was that cache servers within Cloud IAM ran out of memory due to an unexpected high number of modification requests made to the service, which caused those requests to time out [4].

As reflected by the above incidents, unexpected failures impacting users have even occurred in services provided by cloud operators with a proven track record on a global level—there will never be a total absence of failures.

4. Things for Users to Consider

Things that users should consider in relation to the occurrence of system failures in public clouds are summarized below.

4.1 Assume the Occurrence of Failures

Throughout the world, failures occur even in public clouds with a proven track record and wide usage—failures will never be completely eliminated even with the further evolution of public clouds. It is extremely important that users keep this in mind. Using a public cloud without facing this reality makes it difficult to design and operate a system that assumes the occurrence of failures, which can lead to major losses. It is imperative from the start that business owners and management in addition to development and operation managers understand this fact.

4.2 Apply Best Practices

As described above, it's impossible to operate a system with zero failures, but it is possible to reduce the impact of failures. Most public cloud operators provide users with recommendations on system designs for maintaining availability and reliability while simultaneously providing a variety of functions and options to make it easy for users to implement those designs. It is important that users use these functions to implement a design that will enable the system to continue operating as much as possible even if a failure should occur in the public cloud. Fortunately, there are already many users of public clouds throughout the world, and as a result, design patterns that have been adopted in all sorts of use cases have been released as best practices. For this reason, there is no need for the user to design from scratch since adopting those best practices in design and operation can reduce risk.

In addition, functions like PaaS and SaaS have grown in recent years, and services and platforms designed beforehand by public cloud operators based on best practices have come to be energetically adopted by users. This trend in constructing systems with high failure resistance is now gaining

*18 Managed services: Cloud services whose resource provisioning, operation, etc. are mostly the responsibility of the cloud operator. Among cloud computing services, these refer to PaaS and SaaS, for example.
*19 SQS: A type of PaaS offered by AWS providing a message queuing function.
*20 AWS Lambda: A type of FaaS offered by AWS providing an execution environment for application code. The user can execute an application by registering created source code.
*21 Microsoft Azure: A cloud computing service provided by Microsoft Corporation.
*22 Deployment: Installing applications by placing them in their execution environments.
*23 SDP system: A system used by Microsoft to manage the process of safely deploying software.
*24 GCP: A cloud computing service provided by Google LLC.

momentum. The "serverless" design pattern is a good example of this trend. It is a technique for designing and operating systems in which the user need not be concerned about physical servers or even virtual servers typical of IaaS. This article, however, omits details on serverless computing. Using such design techniques and services is advantageous not only in strengthening failure resistance but also in reducing the burden placed on users in system operation. There is no doubt that this trend will accelerate in the years to come.

4.3 Design Operations to Provide against Failures

Even if risk can be reduced through some set of measures, it will still be impossible to completely eliminate failures or their impact. Moreover, while it may be possible to develop measures that can sufficiently deal with failures that can be envisioned beforehand, there are many failures that originate in unexpected cases or events. It is therefore important to quickly identify the cause of a failure at the time of its occurrence and to design operations to enable a speedy recovery to be made. Several specific points in this regard are given below.

1) Enhance System Observability

System monitoring is extremely important even when using public clouds. The composition of an infrastructure particularly when using public clouds often spans IaaS, PaaS, and SaaS, and in the case of applications, the use of containers*25 and microservices is increasing. As a result, systems that are distributed over many types of environments are coming to be deployed, which is making it more important than ever to enhance system observability. This does not simply mean life/death monitoring. Rather, it means the adoption of distributed tracing that tracks individual requests*26 processed by applications and processes at the method*27 level and of a platform service that can consolidate logs and tracking results in an integrated manner and visualize and analyze this information. These mechanisms come in various forms and may be provided by cloud operators or third party*28 services or may be released as Open Source Software (OSS)*29, so it is necessary to select and use the ones that fit the user's current objectives.

2) Lower the Risk of Unexpected Failures

Minimizing the risk of unexpected failures as much as possible is also important for improving operations. Making it a practice of routinely performing failover*30 tests for the system on the cloud and recovery tests to provide against an outage of a specific cloud service is effective in lowering the risk of unexpected failures.

The following introduces chaos engineering as a technique for reducing the risk of failures characteristic of public clouds. Chaos engineering intentionally causes failures to occur in the system's production environment and measures their impact on the entire system. In this way, it becomes possible to observe whether effects that are not expected to be system wide at the time of a failure behave as such and to therefore improve the system's failure resistance. This technique is ideal for a cloud that can manage an infrastructure via an Application Programming Interface (API) and restructure it in any number of ways. Netflix in

*25 Containers: As one type of computer virtualization technology, a method for creating a dedicated area called a container on one host OS and running necessary application software within that container.

*26 Requests: Operation requests made to an application.

*27 Method: An HTTP method such as GET, POST, PUT, and DELETE.

*28 Third party: Refers to a third party manufacturer or developer.

*29 OSS: Software whose source code is released free of charge for anyone to reuse or modify.

*30 Failover: A mechanism for automatically switching over to a redundant standby system when a failure occurs in the main system.

the United States has put into practice such operation on a daily basis to lower the risk of suffering effects from unexpected failures [5]. Additionally, since chaos engineering is applied in a production environment (actual operation environment), it is not a technique to be executed blindly. Rather, it should be applied after putting the documented system design into practice and after sufficient confidence in the failure resistance of a system has been obtained. This point requires special attention.

3) Understand the Configuration of Individual Cloud Services

Of unexpected importance when using a public cloud is the need to understand the configuration of individual cloud services. Of course, details on the inner configuration of a cloud service in relation to security and compliance are not released, and depending on the cloud service, neither are details on the locations themselves of data centers. On the other hand, there are services whose logical configuration at least in part is released so that the user can take that information into account at the time of system design and operation. Obtaining a good understanding of that configuration can aid in identifying the cause of a failure and achieving a smooth recovery.

For example, in the case of AWS, a group of data centers is called an Availability Zone that is physically independent from any other Availability Zone. Understanding this concept in itself is essential to high-availability design. Additionally, in terms of PaaS/SaaS, it is often the case that a service will be rolled out to users after it is configured by AWS itself across multiple Availability Zones from the same perspective as a user. Understanding this makes it possible to grasp whether the occurrence of a failure in a particular Availability Zone will impact individual cloud services. Moreover, on the user side, if an option is provided for cutting off traffic to a specific Availability Zone at the time of a failure, deciding to do so can minimize the impact of that failure. In addition, the EC2 service providing virtual servers has become the foundation for many PaaS/SaaS services, and understanding this makes it possible to predict whether a failure in EC2 "holds the possibility of impacting other services" and to at least stand ready for such an outcome. In addition to the above, there are cases in which the configurations of certain public cloud services are actually released, so responses to the occurrence of failures can be smoothly put into practice the more that these configurations are understood.

5. Support Visualizer Development and Provision

Finally, this section describes a measure that NTT DOCOMO has put into practice in the event of a failure when using AWS, the most used cloud service within the company. NTT DOCOMO manages more than 900 AWS accounts (as of December 2020) that use AWS under various workloads*31. In 2019, however, a failure in the AWS Tokyo Region had not a small impact on NTT DOCOMO and a number of problems came to light as a result. As a measure taken to solve these problems, we constructed a system called Support Visualizer

*31 Workload: An indicator of the size of a system's load, such as the CPU utilization rate. In particular, in a public cloud environment, the workload may represent the system itself, including the OS and application code running on the cloud. In this article, we use the term in this latter sense.

that consolidates information on AWS support cases (described later).

5.1 Problems Identified from AWS Large-scale Failure

As described above, NTT DOCOMO uses AWS under a variety of workloads. Needless to say, system requirements as well as the number of application users, data traffic, etc. differ depending on the workload, so it is not rare for system design and operation to be system dependent and for different types of cloud services to be used. Consequently, if a failure should occur in a cloud service, the extent to which the effects of that failure will be felt will depend on the system. Up to now, there have been systems that suffered no effects at all from the occurrence of a large-scale failure as well as systems in which effects were felt by cloud service users and application users. The following problems came to light with respect to the management of individual systems and the company overall during the occurrence of such failures.

1) Collecting Information from the Entire Company

The impact of a failure on individual systems is not necessarily the same, so it is incumbent on the company to determine without delay the extent of that impact on individual systems and to make appropriate announcements. In actuality, however, consolidating information quickly is difficult given that the operation of each system is independent and that on-site personnel at the time of a failure are busy trying to track down the cause and perform recovery operations. Of course, speedy restoration of the system and minimizing the impact on users should be given top priority, but amid all this, it is also important that the company disseminate appropriate information as their social responsibility. Up to now, it's been very difficult to satisfy both of these needs at the time of a large-scale failure, so the problem of collecting information from the entire company remained.

2) Identifying and Dealing with the Range of Impact

Problems in individual systems also came to light. Since system failures themselves are not limited to the use of public clouds and can occur at any time, a large-scale failure will likely be dealt with in the same way as a system failure. At this time, however, it is not known whether the problem is caused by a failure in the cloud itself or is peculiar to that individual system, so identifying the cause and dealing with it appropriately takes time.

In relation to this phenomenon, most public cloud operators release a service status, so checking on that status at the time of a failure can provide information on that failure and its approximate range of impact. On the other hand, our experience with cloud failures to date has revealed that such service status data does not necessarily list all failures that have occurred. In actuality, there are cases in which a failure becomes known from reports submitted by cloud users and cases in which the cloud operator reports a failure after resolving it.

5.2 Support Visualizer

In AWS, if a user's system happens to be affected by a failure in a service provided by AWS itself, the user may issue a report to AWS in the form of an inquiry ticket[*32] called a "support case."

*32 Inquiry ticket: A unit for managing individual inquiries and their replies.

This operation makes it possible for AWS on its side to collect information on a failure and to eventually isolate its range of impact.

As a user, however, NTT DOCOMO manages each AWS account independently and cannot, as a result, grasp the impact of a failure on other projects on the basis of a support case. There is therefore a need for projects to exchange information directly with each other. However, as described above, on-site personnel are working as hard as possible to recover their system at the time of a failure, so the reality is that they have little time to exchange information. The same can be said of consolidating information from the entire company. In response to this problem, we developed Support Visualizer as a system that consolidates information on support cases issued within the company and that anyone in the company can use to peruse that information.

1) Architecture

The architecture of the Support Visualizer system is shown in **Figure 1**. Although support-case information in AWS accounts is independent of each other, AWS provides an API that can obtain that information. Support Visualizer uses this API to consolidate support-case information, which is indexed and stored in a database to enable it to be searched and analyzed by Amazon's Elasticsearch service. Here, the Kibana*33 dashboard*34 can be used by the Cloud Center of Excellence (CCoE)*35 to peruse and analyze this consolidated information, and in the event that a highly urgent support case

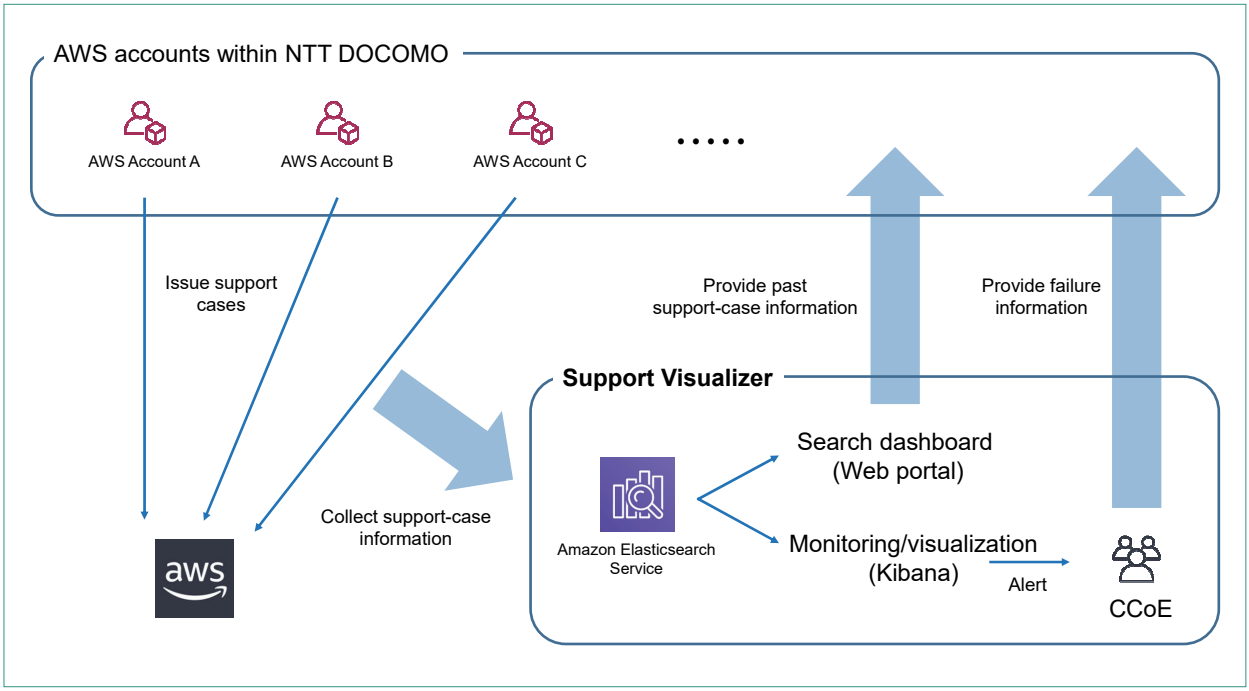


Figure 1 Architecture of Support Visualizer

*33 Kibana: An open-source data visualization tool developed by Elastic.
*34 Dashboard: A screen that consolidates information.
*35 CCoE: An exclusive team within an enterprise that establishes best practices and creates essential systems and governance to make cloud usage successful.

comes to be issued, the CCoE can be notified using Slack*36 or a similar tool.

We have also deployed a portal to enable users within the company to peruse this in-house consolidated support-case information and to filter that information by keywords or services, degree of urgency, update time of support-case information, etc. (**Figure 2**).

2) Problems for which Solutions Can Be Expected

Given the occurrence of a system failure on the cloud and the issuing of support cases from separate systems, this deployment of Support Visualizer enables even other users within the company to check the content of those support cases. This is especially effective at the time of a large-scale failure on the cloud—even if system operators are busy responding to the failures in their own systems, support cases issued by those systems will still be consolidated automatically. In this way, the company can easily obtain an overall view of the extent to which a failure is impacting its systems. At the same time, operators of individual systems can determine whether the effects on their systems are separate phenomena or a phenomenon occurring on the cloud overall. Additionally, if a similar phenomenon has already occurred on other systems,

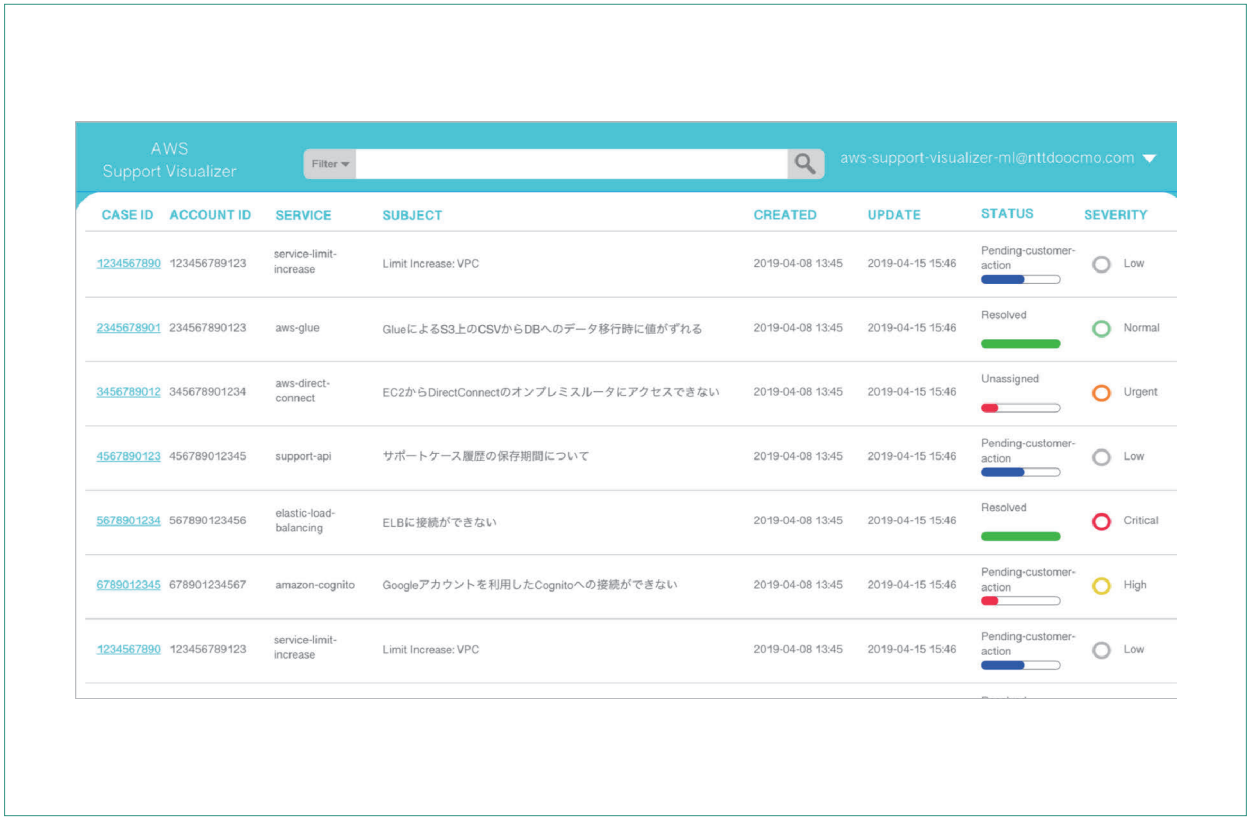


Figure 2 Screenshot of dashboard for Web portal of Support Visualizer

*36 Slack: A business chat tool provided by Slack Technologies, Inc.

it may be possible to check what measures were taken to solve that problem. NTT DOCOMO manages more than 900 AWS accounts of various workloads, so we can expect Support Visualizer to enable users to grasp the impact of a failure at actual sites, which is something that cannot be obtained only on the basis of status information released by the cloud operator.

3) Side Benefit

A side benefit of Support Visualizer is that technical inquiries on development and operation in everyday use of a cloud come to be consolidated, which means that technical know-how on using a cloud from within the company can also be consolidated. Going forward, we aim to incorporate a mechanism for analyzing trends in consolidated support cases and actively consolidating know-how from the results of that analysis. In this way, we can feedback analysis results to each system and promote more efficient cloud usage.

6. Conclusion

This article described system operations that anticipate the occurrence of large-scale failures in public clouds. Despite the ongoing evolution of technologies and architecture such as clouds, containers, and microservices, it is impossible to construct a system with zero failures. We will continue in our efforts to minimize the impact of failures as much as possible and to cultivate best practices so that NTT DOCOMO can make productive use of public clouds throughout the company.

REFERENCES

[1] Ministry of Internal Affairs and Communications: "Cloud Characteristics and Security," (In Japanese). https://www.soumu.go.jp/ict_skill/pdf/ict_skill_2_3.pdf

[2] AWS: "Summary of the Amazon EC2 and Amazon EBS Service Event in the Tokyo (AP-NORTHEAST-1) Region," Aug. 2019. https://aws.amazon.com/jp/message/56489/

[3] Microsoft Azure: "RCA – Authentication errors across multiple Microsoft services and Azure Active Directory integrated applications (Tracking ID SM79-F88)," Sep. 2020. https://status.azure.com/status/history/

[4] Google Cloud: "Google Cloud Infrastructure Components Incident #20003." https://status.cloud.google.com/incident/zall/20003

[5] Netflix: "The Netflix Simian Army," Medium, Jul. 2011. https://netflixtechblog.com/the-netflix-simian-army-16e57fbab116

Technology Reports Cloud MEC Application Design Patterns

Application Design Patterns in MEC

Innovation Management Department Yoshikazu Akinaga

5G network construction has progressed in recent years, and NTT DOCOMO launched 5G services in March 2020. 5G has three major characteristics - high-speed/capacity, low latency, and simultaneous connection to many devices. These high-speed/capacity and low latency characteristics enable services that have been difficult to achieve with conventional mobile technology, and thus hold great promise. For this reason, MEC is attracting attention as an approach to building systems that take advantage of high speed/capacity and low latency. Also, since June 2020, NTT DOCOMO has also been offering cloud server called "docomo Open Innovation Cloud" as well as a service to connect it directly into the mobile network called "Cloud Direct". To take advantage of these services, we propose application design patterns in MEC, which are architectural templates used for functions required to build applications when considering architecture. In this article, we describe some typical patterns and their effects.

1. Introduction

In recent years, construction of networks for 5th Generation mobile communication systems (5G) has been progressing. NTT DOCOMO launched its 5G services in March 2020. 5G has three major characteristics - high-speed and high-capacity with enhanced Mobile Broad Band (eMBB), Ultra-Reliable and Low Latency Communications (URLLC), and simultaneous connection of multiple devices with massive Machine Type Communications (mMTC), which are expected to have an impact on various

applications, solutions and industries. With these high-speed, high-capacity and low-latency characteristics, services that were previously difficult to achieve with mobile devices are now expected to become a reality.

For this reason, the positioning of clouds on networks, called Multi-access Edge Computing (MEC), is attracting attention as an approach to building systems that can take advantage of high-speed, high-capacity, and low latency. NTT DOCOMO has also been offering cloud servers called "docomo Open Innovation Cloud" as well as a service to connect it directly into the mobile network called "Cloud Direct" since June 2020. These services hold promise for the provision of a variety of applications and solutions that take advantage of 5G characteristics.

To take advantage of these services, we propose application design patterns in MEC. Referring to these patterns will enable developers to design applications with optimal placement of functions in the cloud and MEC. In this article, after describing application design patterns and MEC, we describe some typical patterns and their effects.

2. What is an Application Design Pattern?

An application design pattern is an architectural template used for functions required for building applications when considering architecture, and is a collection of generalized ideas based on typical use cases available for use as application architecture.

Similar to general design patterns, application design patterns are highly reusable because they are discussed as parts rather than as whole applications, which makes it easier for application architects to immediately consider and incorporate new functions.

As an example, **Figure 1** shows an application design pattern on a public cloud*1 for building Web

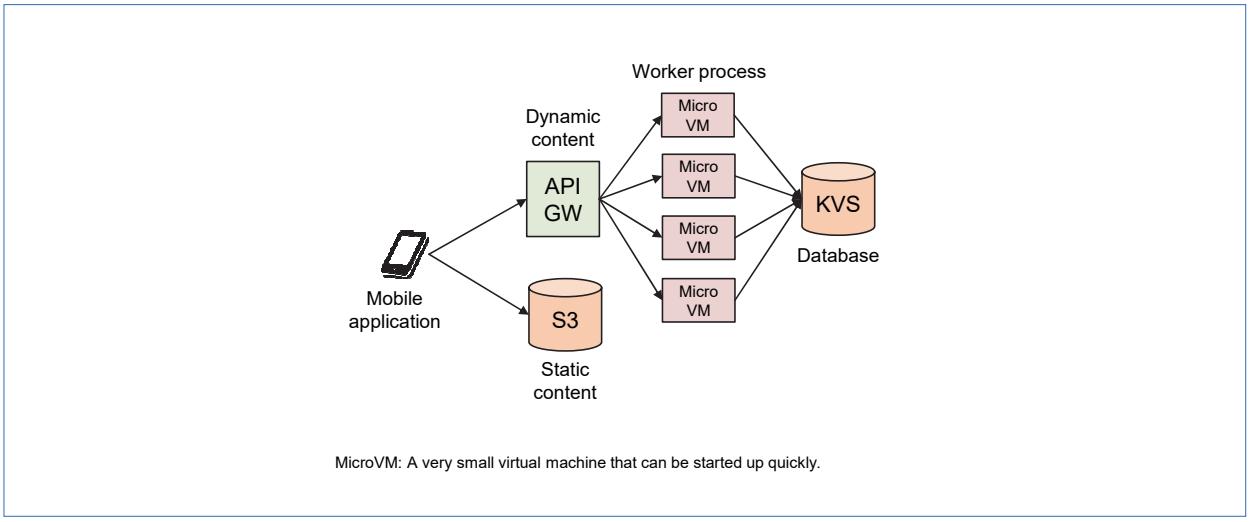


Figure 1 Example application design pattern (serverless)

*1 Public cloud: A cloud computing service that can be used by anyone via the Internet.

services. Fig. 1 shows an application design pattern for a typical serverless application commonly used in public clouds. It is easy to see at a glance that dynamic and static contents are stored separately, that worker processes*2 are finely arranged for scaling out, and that a Key Value Store (KVS)*3 is used to enable access to avoid database bottlenecks. Designers can implement scalable mobile applications by following these design patterns to design applications. These organized patterns of design know-how are application design patterns.

3. What is MEC?

Although there is growing momentum to take advantage of low latency on 5G networks, low latency cannot be achieved just by making radio sections 5G. To achieve low latency as a system, the entire network must be considered, and its sending and receiving distances and the number of devices using it must be reduced. Therefore, NTT DOCOMO is attempting to solve this problem by having computing resources within the 5G network. This is MEC. MEC is achieved by placing the computing resources (virtual machines) as close as possible to radio sections (**Figure 2**). The closer locations are to radios (closer to base stations), the shorter the latency can be, although this would dramatically increase the number of base stations and inevitably increase the number of required virtual machines which would be economically impractical. In contrast, it's possible to attempt aggregation if locations are close to the Internet, but because this increases latency, it is important to allocate resources appropriately based on the balance between supply and demand.

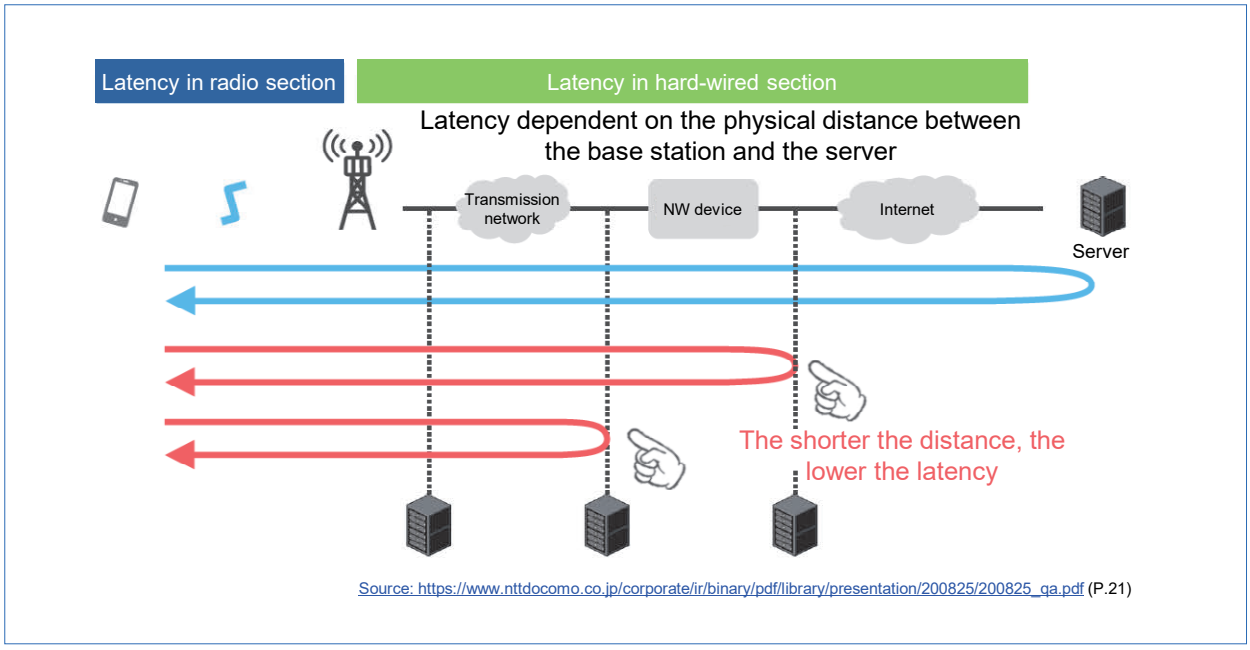


Figure 2 Location of MEC on the network

*2 Worker process: A single program that is launched with a certain role and terminates when the role is completed.

*3 KVS: A high-speed database that specializes in simple management of data with sets consisting of a key and a value. These are often used to avoid processing bottlenecks because they support distributed processing on multiple servers and can handle huge amounts of input and output simultaneously.

3.1 Four Benefits of MEC

From the user perspective, MEC promises the following four main characteristics.

(1) Traffic optimization, terminal-to-terminal communications

Peer to Peer (P2P) communications between terminals can be done with very low latency by accessing or passing traffic through a server that is geographically close to terminals and the server can send large amounts of data. This holds promise for games and other applications with strict response time requirements.

(2) Low latency, low jitter

Accessing servers that are geographically close makes it possible to connect with low latency and take advantage of the characteristics of 5G. Also, unlike the Internet, provision over a closed network reduces the causes of jitter and hence holds promise for streaming and other applications that are vulnerable to such fluctuations.

(3) Secure and private network

Secure connections can be provided because there is never any connection to an external network such as the Internet. Private connections with SIM authentication will be possible and enable networks that handle highly confidential information.

(4) Leveraging edge computing resources and complementary terminal functions

Edge computing[*4] enables processing of large loads such as advanced AI processing and the generation of high-definition 3D images, which cannot be processed by individual terminals, and is anticipated for use in games, 3D Computer Aided Design (CAD) and XR[*5] among others.

Incorporating these MEC characteristics into specific applications is discussed with application design patterns.

3.2 Preconditions for Application Design Patterns in MEC

Because of the nature of servers distributed across regions, MEC differs from systems operating in centralized data centers in terms of fault tolerance and robustness. Therefore, a method called "design for failure" is incorporated into the applications on MEC. This is a method of building applications assuming they could fail in public cloud computing. For example, data durability in MEC is not guaranteed, because it requires a significant amount of redundancy, which is difficult to obtain in a distributed system. Therefore, it is preferable to implement data durability in a public cloud or other storage service rather than MEC.

Other preconditions in MEC include the following.

- Robustness and availability of MEC are both low.
- MEC provides Infrastructure as a Service (IaaS)[*6].
- MEC does not provide Platform as a Service (PaaS)[*7] which has high durability data storage
- On the other hand, MEC could provide functions to improve portability, such as container[*8] management services.

*4 Edge computing: Technology that distributes edge servers closer to the users to improve response and reduce latency.

*5 XR: A general term for technologies such as VR, AR, and MR that provide new experiences through the fusion of virtual space and real space.

*6 IaaS: A online service in which servers and networks are provided virtually. The user sets up an OS and application software on the online server or network and uses it.

*7 PaaS: A online service that provides a platform including an OS and middleware to run applications on the service. The user creates and uses application software on the platform.

- MEC also provides Domain Name System (DNS)[*9] services.

4. Application Design Patterns in MEC

The following discusses some specific application design patterns and their use cases.

4.1 A Pattern for High Speed, Large Upload

A function that applications can take advantage of the high speed and high capacity of 5G is uploading, which has traditionally been a weakness of mobile networks. Significantly higher upload speeds make it possible to consider utilization, although due to issues such as increased latency, jitter, and reduction in throughput due to latency in Transmission Control Protocol (TCP)[*10] ACKnowledgement (ACK)[*11] responses due to such increased latency and jitter, it is not possible to obtain sufficient speed in communications to the Internet. However, MEC makes high-speed, large uploading possible.

1) Challenges

To speed up large uploads using the high-speed 5G network, and avoid the dependency of the upload process on the latency of TCP ACK packets, jitter and the influence of servers on the internet, and make the upload process stable. There is also the challenge of accepting a large number of large uploads at the same time.

2) Design Pattern

To take full advantage of 5G upload capabilities, uploads should be terminated at an MEC local server and temporarily stored. Then, stored contents are queued[*12], retrieved by a worker process from a separate temporary upload destination, and perpetuated in public cloud storage. It is also possible to run separate processes (e.g., image/video processing by AI) while data is in temporary storage. The DNS lookup service helps search for nearby servers to enable connection to the closest upload server. Also, since temporary storage is effective in load balancing, it can be used to handle mass uploads from the same location (e.g., simultaneous uploads from spectators in a stadium) when there is not enough bandwidth on the Internet. Since only the upload process can be implemented separately, it is easy to integrate into existing systems. In addition, a Load Balancer (LB)[*13] should be included in MEC to duplicate the upload server and ensure availability (**Figure 3**).

4.2 A Pattern for Ultra-low Latency Messaging and Status Management

The low latency of 5G enables multi-player games to be synchronized across multiple devices, simultaneous XR experiences and message exchange, etc. KVS Pub/Sub functions[*14] in MEC that achieve the synchronization provide very versatile messaging services.

1) Challenges

To achieve low-latency synchronization by putting ultra-low latency KVS and its Pub/Sub functions in MEC. To achieve state synchronization between terminals in competitive and open-world games and state synchronization of IoT devices with ultra-low latency. To also provide services with synchronization latency below a certain level to

*8 Container: A type of computer virtualization technology in which a dedicated area (the container) is created on a host OS, and the necessary application software is run in the container.
*9 DNS: A system for mapping host names and IP addresses on an IP network.
*10 TCP: A higher-level Internet protocol that is used as a standard. It plays a complementary role to IP by confirming the connection destination and data arrival, controlling the flow of data, and detecting duplicate or missing data to realize highly reliable communications.
*11 ACK: A reception confirmation signal to notify the transmitting node that the receiving node has received (decoded) the data correctly.
*12 Queuing: The creation of a queue and temporarily storing the order of processing and the content to be processed in it.

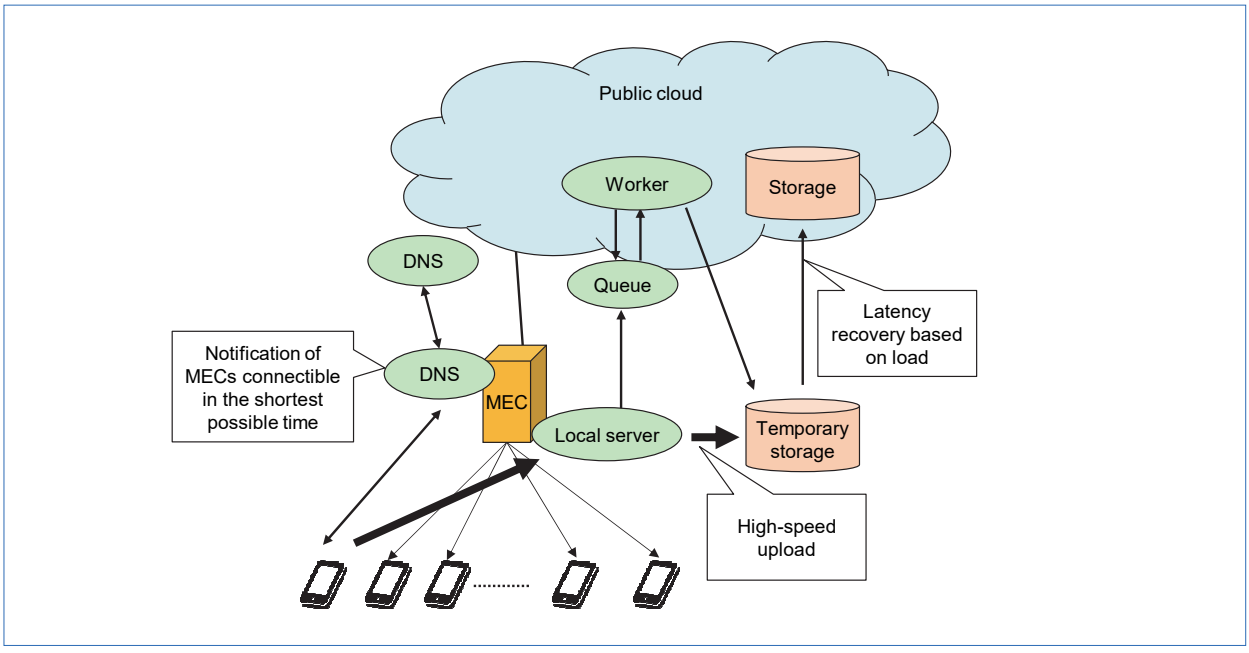


Figure 3 High-speed large upload design pattern

eliminate impaired gameplay, such as warping*15.

2) Design Pattern

An in-memory DB*16 such as Redis should be installed in MEC to perform messaging, temporary status management and synchronization of application at ultra-high speed between terminals. These in-memory DBs can be distributed and reference information in the order of nanoseconds. Here as well, a neighboring in-memory DB lookup service in DNS can be used to connect to neighboring in-memory DBs. Using the Pub/Sub function, messages can be exchanged between neighboring terminals with ultra-low latency, and similarly, KVS can be used for application state management, etc. Using these in-memory DBs makes it possible to aggregate game scores for the regions in which players are playing at very high speed and aggregate game rankings (real-time leaderboards*17). These can also be used as service buses*18 for coordination of multiple systems, state synchronization between players in network games, and synchronization of remote robots (**Figure 4**).

3) Points to Note

In-memory DBs have cases where authentication does not exist, so message encryption and network isolation, etc. are necessary when they are implemented.

4.3 A Pattern for Web Service Caching

One major method of real-time updating on the Web is to use Web sockets*19. When implementing such services, major issues include latency and load balancing of traffic. MEC is also effective in solving these issues.

1) Challenges

To update dynamic content with as low latency

*13 LB: A device that balances the load of communication traffic.

*14 Pub/Sub function: A function that allows asynchronous message exchange between Publish and Subscribe functions. Since the Subscribe side distributes to everyone what is sent by the Publish side, this function is suitable for distributing 1: N messages.

*15 Warp: A phenomenon in which a player suddenly disappears and reappears from a different location due to mis-synchronization of location information in a game in which two or more players are participating. This phenomenon is often detrimental to gameplay, therefore game providers try to avoid it if possible.

*16 In-memory DB: A database that enables fast response to information acquisition by expanding and holding data in memory for processing.

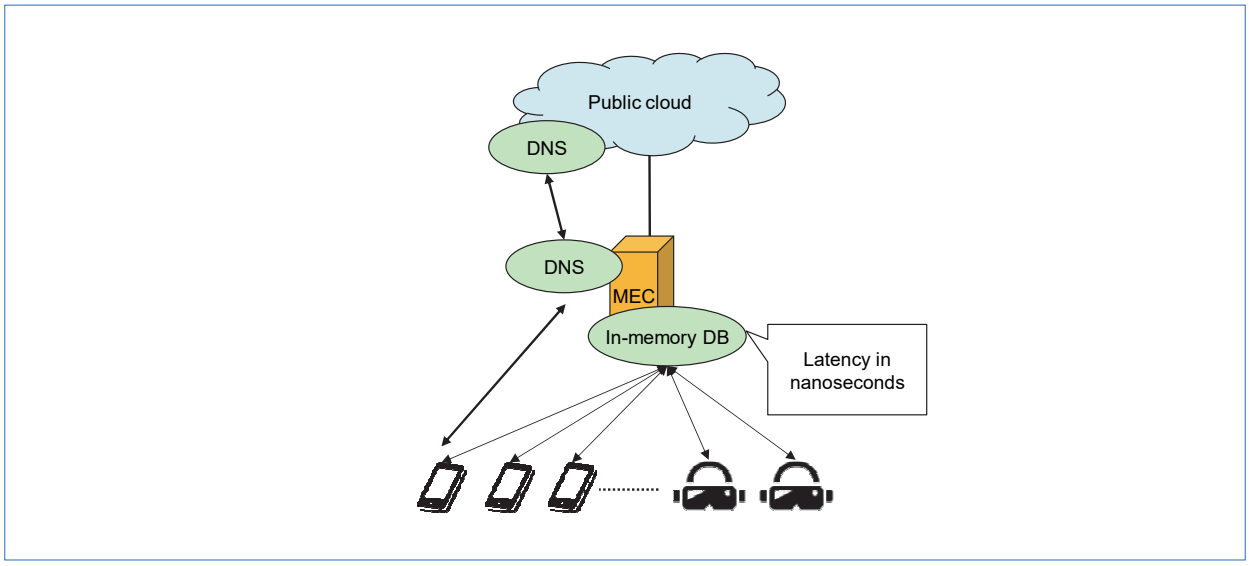


Figure 4 Ultra-low latency messaging and status management design pattern

as possible (e.g., Web sockets) and to achieve highly responsive services. There are also issues such as reducing the load of reading static content on the Web.

2) Design Pattern

A caching mechanism for Web services in MEC enables the presentation and application layers of the three-tiered Web structure*20 to be placed in MEC to achieve higher speeds. The database layer can also be sped up by having read replicas*21 and a cache mechanism in MEC. It is important to note that if the database layer is not deployed in the public cloud, data persistence becomes difficult, and consistency cannot be maintained over a wide area. Therefore, the cache is placed near the application layer to achieve higher speed.

On the database layer, the read replicas of MySQL*22 and multi-master*23 are also effective for load balancing. However, in anticipation of low latency, care must be taken to balance the latency of the database layer with the latency of the network, because the latency of the database layer is significantly greater than the latency of the network (**Figure 5**).

3) Points to Note

To automatically deploy the application layer close to the customer, schemes for containerization, etc. should be implemented in consideration of portability. Also, in-memory caching (KVS) is used for Web caching to enable ultra-fast response times. The data persistence issue needs to be considered for writing in this case.

This pattern is easy to implement, but care must be taken because management costs may increase as the number of MEC servers increases. In addition, using container management solutions and other methods to package implementation while deploying the minimum number of applications required is an important point in keeping the system efficient and cost effective.

*17 Leaderboard: A board that lists the top rankings and one's order in a game, etc.

*18 Service bus: A mechanism for exchanging messages to mutually link statuses or pass on processing to the next system when multiple systems are linked.

*19 Web Sockets: A technical standard for exchanging two-way messages over the Web, defined as Request For Comments (RFC) 6455.

*20 Web three-layer structure: A method of dividing the components of a Web system into three layers (presentation layer, application layer, and data layer) and designing them as independent modules.

*21 Read replica: A read-only copy of a database. Such a copy of a main database is always kept to reduce the load of reading and searching.

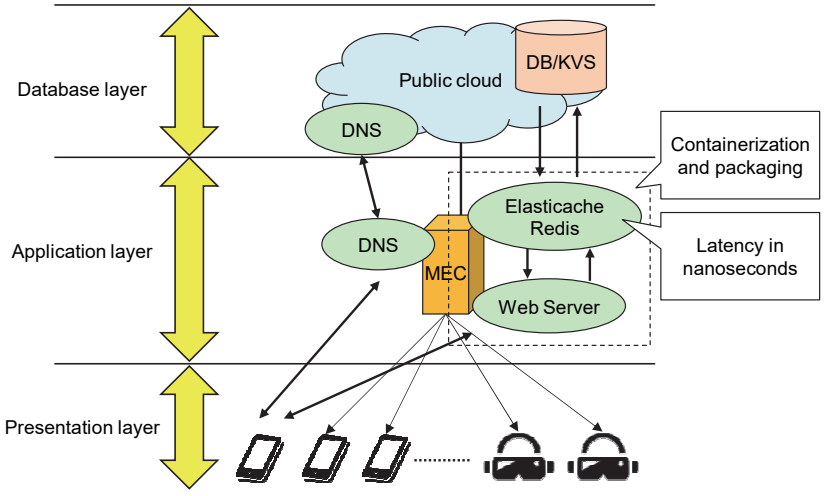


Figure 5 Web service cache design pattern

4.4 A Pattern for Lightweight Protocols of IoT Devices

The problem of security in IoT devices is solved by blocking access from the Internet and keeping low-load and low-security protocols within the carrier network. This is highly convenient because it enables isolation from the Internet and handling of processing from many devices with low latency.

1) Challenges

Since some IoT devices are too small to implement Secure Sockets Layer (SSL)[*24] and other similar protocols, there is the challenge of adopting lightweight protocols while maintaining security. There are some examples of implementing Message Queuing Telemetry Transport (MQTT)[*25] with Transport Layer Security (TLS)[*26]. However, because communications encryption is too heavy for such devices, the challenge is to use lightweight protocols with less authentication and encryption to reduce device battery consumption, yet maintain security.

2) Design Pattern

Lightweight protocols for IoT devices, such as MQTT, can be safely terminated within the mobile network and processed and encrypted when they leave MEC to enable secure management of IoT devices. This prevents attacks from the Internet and enables secure device management systems to be built (**Figure 6**).

4.5 Other Design Patterns

Other design patterns that have been proposed are shown in **Table 1**. Going forward, we would like to increase the number of design patterns as specific methods of effectively using MEC.

*22 MySQL: One of the most popular open-source Relational Database Management Systems (RDBMS).

*23 Multi-master: A method of improving the reliability and performance of a database. Refers to a system that can have multiple master servers. Even with multiple connections to such servers, the data is the same, and there are no restrictions on the functions that can be used.

*24 SSL: A protocol for encrypting communications and detecting data tampering between applications on a network, primarily between WWW browsers and WWW servers.

*25 MQTT: A lightweight message queue protocol of the Pub/Sub type, used to exchange messages between various devices and servers on the IoT.

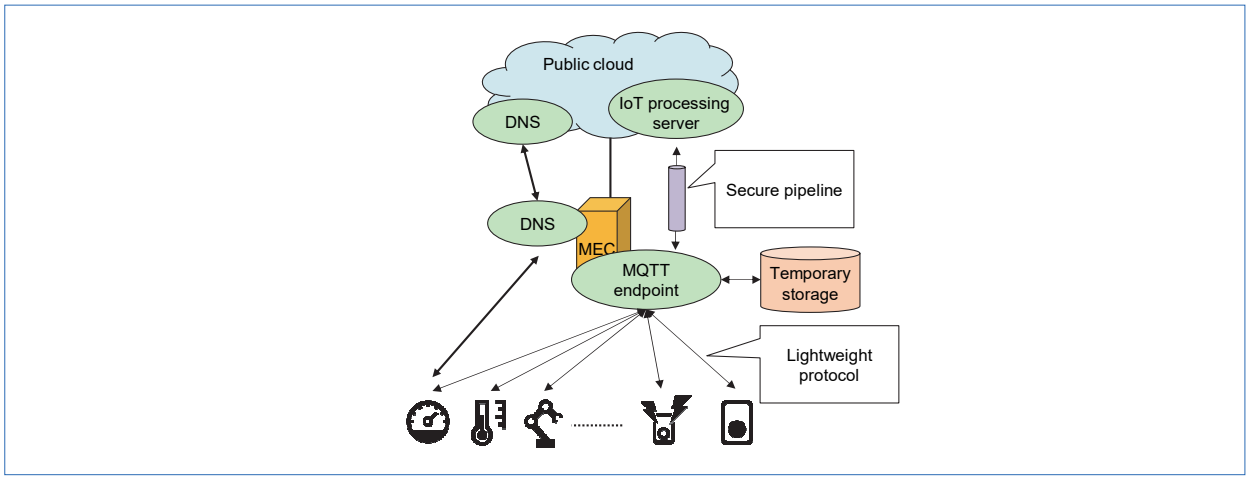


Figure 6 Implementation pattern of lightweight protocols for IoT devices

Table 1 Other proposed application design patterns in MEC

Pattern name	Challenge	Design pattern
CDN	To reduce the load on the origin server (delivery server) for video and streaming services. To prevent quality degradation due to network jitter to the origin server. When services are provided at ultra-high speed and ultra-large capacity, the higher the network level the more the load concentrates, and quality tends to deteriorate. The challenge is to provide as much video as possible in real time.	The closest server can be determined by having the DNS return the closest location, just as in an ordinary CDN system. It can retrieve the cache of the closest location. Each server retrieves the latest information from the origin server (e.g., Web server in the public cloud, streaming server) when there are changes. The same logic can be applied when using HLS, etc. for streaming, which is highly versatile.
P2P traffic optimization by reflection	To optimize P2P traffic and provide stable and high-quality services through loopback communications on the local edge network. For example, to optimize video calls by making them limited locally and looping them back.	The shortest possible routing within the network of the loop back traffic over the 5G/4G local network enables terminal-to-terminal communications with low latency. Placing a signaling server in MEC will make it possible to implement WebRTC and VoIP (SIP), etc. If the signaling server is deployed in a wide area, it does not have to be in MEC.
Local secure network	To realize a local network between terminals and build a secure network without using VPNs (e.g., file servers).	Placing VPN servers in MEC enables creation of any private network. In addition, because this can achieve high speed and large capacity, access to file servers, etc. can be built safely and easily.
Security using geographic constraints	To prevent access from outside of certain areas and provide services more securely.	Placing a file server in MEC only in a specific region enables the system to access servers only from systems in the specific region. The regional MEC server is identified with DNS and that server is accessed. Access to MEC is with information only accessible in the region or using an authentication method. Storage or DB is deployed on a public cloud, and its robustness and availability are ensured. To ensure availability, LBs should be deployed in MEC to create redundant systems.

CDN: Content Delivery Network
HLS: HTTP Live Streaming
SIP: Session Initiation Protocol
VoIP: Voice over Internet Protocol
VPN: Virtual Private Network
WebRTC: Web Real-Time Communication

*26 TLS: A protocol that regulates SSL as an Internet standard technology and ensures its extended security. It has extended cryptographic algorithms and error message regulations compared to SSL.

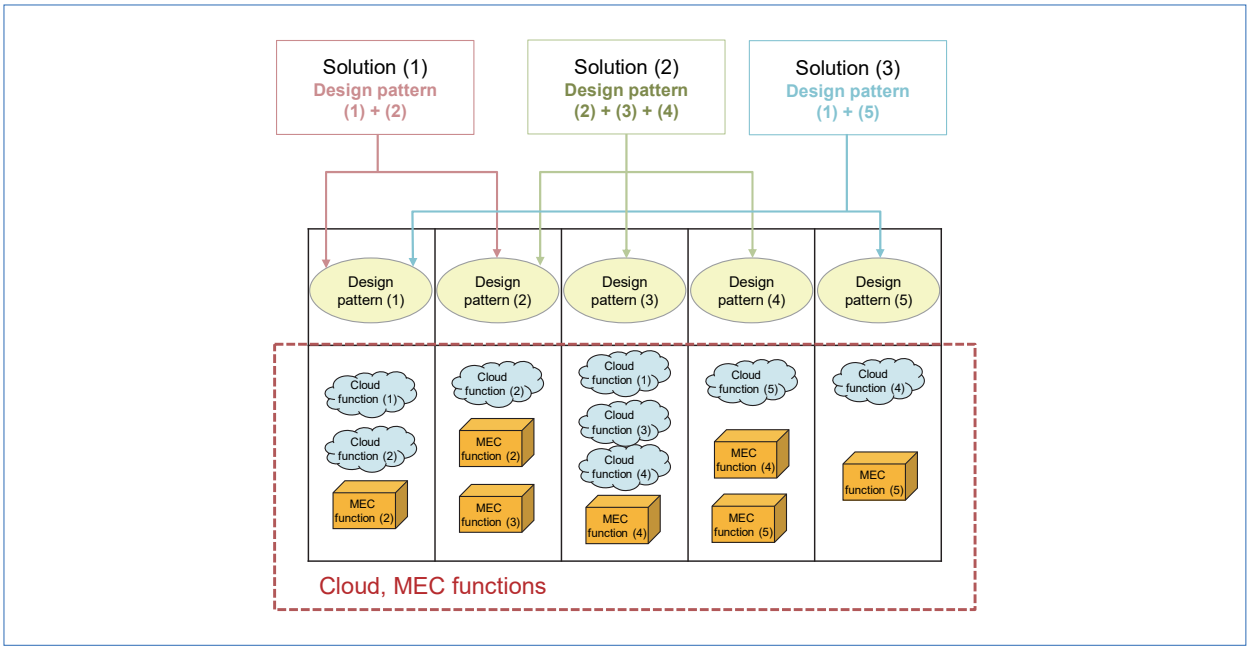


Figure 7 Relationship of application design patterns and solutions

5. Relationship between Application Design Patterns in MEC and Solutions

Application design patterns in MEC only indicate useful usage patterns (best practices) to achieve specific functions to use MEC effectively. These design patterns are combined to build an application or solution. The relationship between patterns and solutions is shown in **Figure 7**.

6. Conclusion

In this article, we explained the importance of application design patterns to make MEC more useful. For effective use of MEC, we believe that the accumulation of these best practices and the reuse of this knowledge will lead to the creation of new value-added services in the 5G era. Going forward, as policy for the provision of functions on MEC, we will strive to increase the variety of design patterns so that they can be applied to a wider range of applications. We also want to study and provide PaaS functions (functions that make it easy to use functions to achieve design patterns without having to implement them yourself) on MEC to make it easier to use common design patterns.

Technology Reports

AI | Voice Recognition | Dialogue

"AI Phone Service" to Automate Telephone Reception and Monitoring

Service Innovation Department Tomoko Kawase
5G & IoT Business Department Shin Oguri
Service Design Department Yuki Saito

The number of businesses implementing contact centers is increasing with the spread of cloud-based contact center systems, but the shortage of human resources for operators has become a problem. To automate routine telephone answering and post-answering office work, NTT DOCOMO has developed an automated telephone answering service called "AI Phone Service," which features identity verification through voice recognition. This service makes it possible to automate tasks such as accepting reservations and applications, and monitor the elderly.

1. Introduction

In improving customer satisfaction, telephone contact points such as contact centers are significant as they are customer contact points that do not require IT literacy. In recent years, cloud-based contact centers*1 have become popular, and the number of businesses implementing contact centers is increasing. However, to respond to an increasingly diverse range of customers in an easy-to-understand and prompt manner, and to improve customer satisfaction, in addition to business knowledge, operators need communication skills as well as IT skills to carry out post-call operations, but there are not enough human resources to fill these positions. Also, staffing must be flexibly adjusted during busy and off-season periods. As well as that, the number of people in contact center offices must be reduced to prevent the spread of the novel coronavirus. Against this backdrop, there are growing demands for automated telephone answering services using Artificial Intelligence (AI). Overseas, services using AI to support telephone operations have been available since

around 2018, and in Japan, many cases implementing AI telephone application acceptance were announced in 2020. Using AI to take over routine answering duties conventionally performed by operators means operators can focus on non-routine answering duties. In addition, AI can limit the IT skills required of operators by taking over post-call tasks, which will help reduce the shortage of IT-literate operators.

NTT DOCOMO has already implemented a voice recognition Interactive Voice Response (IVR)*2 [1] function for IVR mechanisms for general inquiry counters (general centers). These systems automatically connect users to the appropriate specialist center to handle their inquiries, and utilize the spoken dialogue service know-how accumulated through the AI agent services “Shabette Concier*3” and “my daiz*4” [2]. Voice recognition IVR implementation has had the effect of reducing waiting time before being connected to an operator, reducing the time spent answering the phone at the general centers, and reducing the amount of work transferring to specialist centers.

NTT DOCOMO has developed a new cloud service called “AI Phone Service” to provide customers with a solution for improving the efficiency of their telephone answering services with voice interaction technology. This article provides an overview of the AI Phone Service and its mechanism, and describes approaches to speech recognition technology to support various use cases.

2. Service Overview

The AI Phone Service is designed to be used by businesses such as local governments, retailers, restaurants and companies with call centers. As shown in **Figure 1**, use cases with received phone

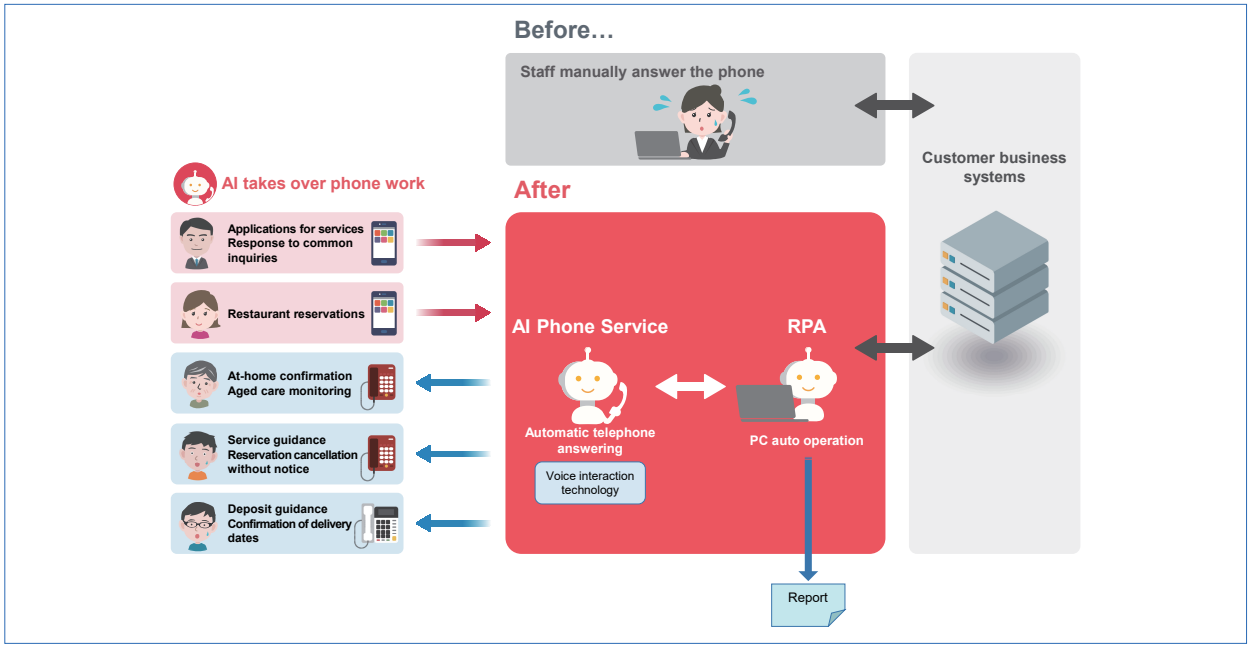


Figure 1 Overview and use cases for AI Phone Service

*1 Cloud-based contact center: A system for responding to customers over the phone that operates using servers on a network, rather than servers owned by a company.

*2 IVR: A system that provides voice guidance over the phone, such as “For XX, please press Y.”

*3 Shabette Concier: A speech dialogue agent that runs on smartphones or tablets, providing conversation with characters, making phone calls through conversation, setting alarms, searching for transfers, and fortune telling.

*4 my daiz: A speech dialogue agent that runs on smartphones and tablets, providing a wide range of information suited to the user.

calls include service application/modification, responding to common inquiries, and accepting reservations for restaurants and vehicle dispatch. This system can be used not only for receiving calls but also for making calls and can be applied to a wide range of applications beyond contact center operations, such as checking up on the elderly in the homes, monitoring their health, providing customers with service information, and confirming and reminding customers of reservations and payment information. In addition, the system not only automates the routine tasks that operators formerly performed, but also automates post-call tasks by linking spoken dialogue technology with Robotic Process Automation (RPA)*5. For example, it is possible to automatically create reports based on dialogue content logs and link with the customer business systems. Implementing the AI Phone Service will not only help solve the problem of securing human resources but will also enable 24/7 support.

3. System Configuration

The AI Phone Service system configuration is shown in **Figure 2**. The automatic telephone answering service is realized by linking the "AI phone core application" to the "docomo AI Agent Application Programming Interface (API)*6" [2], which provides the NTT DOCOMO dialogue technology, and the "Amazon Connect" cloud-based call center service. As telephones are used, voice interface functions are provided by a voice recognition engine.

The docomo AI Agent API provides functions that enable AI to respond to the user according to

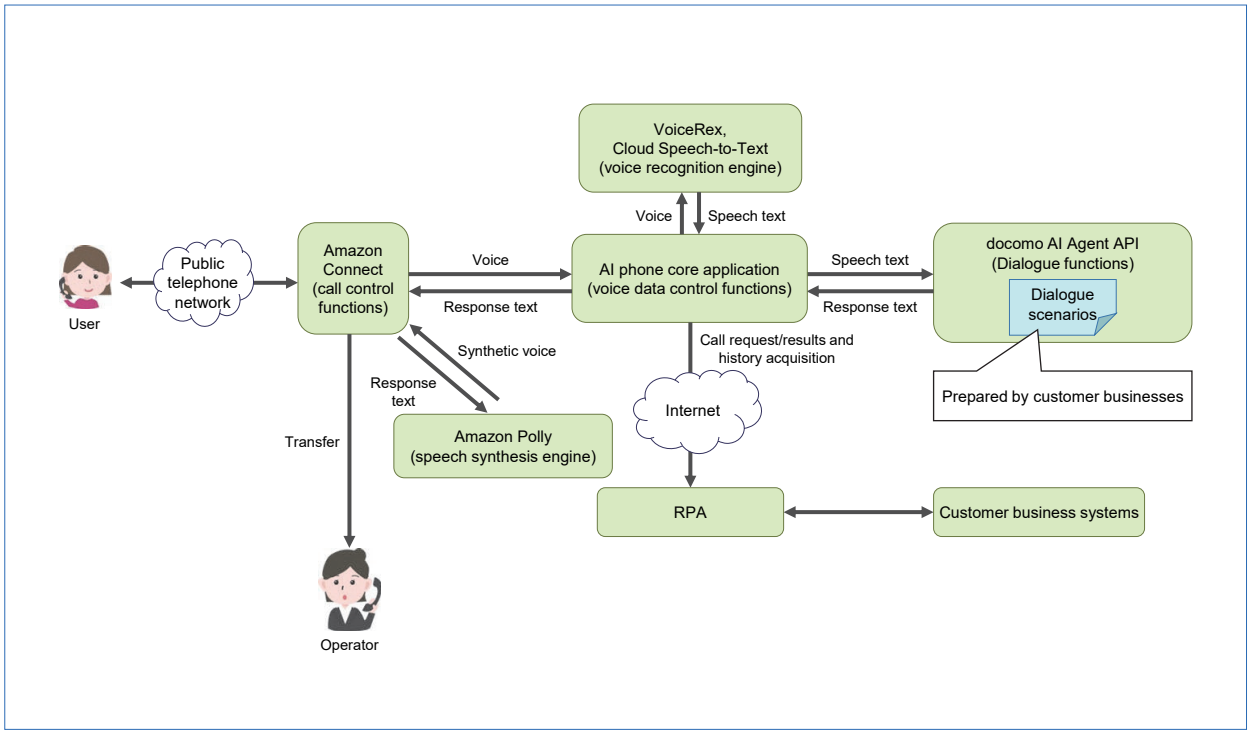


Figure 2 AI Phone Service system configuration

*5 RPA: A mechanism to automate routine tasks.

*6 API: An interface that enables the functions of software to be used by other programs.

predetermined dialogue scenarios, and notably, customer businesses can flexibly create their own dialogue scenarios.

Amazon Connect provides call control functions that give customer businesses using the AI Phone Service the advantages of easy expansion and not having to manage call control servers. There is also a function to transfer calls to operators when cases arise that are not easily handled automatically by AI. However, one of the limitations of using Amazon Connect is that the only text-to-speech[*7] engine that can be used is "Amazon Polly." Amazon Polly cannot reproduce the synthesized voice of a specific person to use as the AI voice, and in the case of Japanese language, there is only a choice between one male and one female speaker. Nevertheless, by tuning the speaking speed, pauses, and volume within the range of the selected speaker's voice, it is possible to make the AI speak slowly and loudly for important words, for example.

In sequence, the AI phone core application feeds the user's voice obtained from Amazon Connect to the voice recognition engine[*8] and then receives the speech text as the recognition result. Once the voice recognition engine detects the end of the user's speech, the AI phone core application works with the docomo AI Agent API and RPA to perform the subsequent processing.

The system uses NTT "VoiceRex[*9]" and Google Cloud Speech-to-Text as voice recognition engines. It mainly uses the former, but it can also select the latter each time it is a user's turn to speak. Since each engine has different areas of strength, it is possible to specify an engine to utilize its advantages in the scenario in advance, depending on what is to be heard. For example, when a person needs to be identified over the phone, it is necessary to listen to his or her name. VoiceRex can output voice recognition results in both kanji and kana and can also output information that the word is classified as "first name" or "last name," which is useful for accurately recognizing Japanese names.

4. Approaches to Speech Recognition Technology for Various Use Cases

In the AI Phone Service, VoiceRex is applied with the NTT DOCOMO original language model[*10], which has been proven in Shabette Concier, my daiz, and voice recognition IVR. To apply VoiceRex to automate telephone answering in a variety of use cases, we worked on the following three things.

4.1 Speech Recognition for Names

In use cases such as service application/modification and taking reservations, it is important to complete the identity verification scenario, which requires accurate voice recognition of the caller's name. Therefore, we tuned the language model by adding names as training data to the aforementioned language model. Real names cannot be used due to restrictions on personal information, so fictitious Japanese names were generated and used as training data. We evaluated name recognition performance before and after tuning and found that the error rate fell to less than 70% of the error rate before tuning.

4.2 Speech Recognition for Scenario-specific Words

Dialogue scenarios differ with each customer business, and user speech assumed in each scenario

*7 Text to speach: Technology for artificially creating speech data from text and verbally reading out text.
*8 Voice recognition engine: Equipment that takes voice data as input and converts it to text of what was spoken.
*9 VoiceRex: A voice recognition engine developed by NTT Media Intelligence Laboratories.
*10 Language model: A model that represents the frequency of word order.

also differs for each dialogue scenario. If words, which are frequently used in a particular scenario, are rare in general dialogue, it is not uncommon for AI agents to misrecognize them. For example, in the use case of restaurant reservations, users might frequently utter the word “private room.” However, since the frequency of occurrence of “koshitsu (private room)” is not so high inside the aforementioned language model, it may be misrecognized as words such as “hoshitsu (moisturizer)” or “koushitsu (imperial family)” (similar sounding words in Japanese). Also, in principle, service names that are not included in the language model cannot be recognized. Therefore, it is generally necessary to tune the language model in advance, as mentioned above. However, tuning the language model and implementing it in the voice recognition engine every time a dialogue scenario is added is not practical from either the computational or operational perspectives. Therefore, VoiceRex has a function that enables specification of a list of expected words for each speech recognition request, which makes it easier to output specified words without tuning the language model. This function enabled improved speech recognition performance.

4.3 Appropriate Timing of Responses Based on Speech Content

In spoken dialogue with AI agents, the speed of the response, i.e., the replay of AI speech a short time after user speech finishes, is a factor that leads to better user experience. However, when a user pauses (to breath, etc.) while saying such things as an address, a sequence of numbers or an open-ended response, if AI regards the pause as the end of the speech, it might not hear the speech after the pause, or it may interrupt the user’s speech and start responding. In other words, there is a trade-off between the success rate of listening to pause-containing speech and response speed.

Therefore, in the AI Phone Service, with each user’s turn to speak in a scenario, an allowable pause length is set according to the expected speech content, which is dynamically specified by the AI phone core application to the voice recognition engine. For example, after the AI asks, “Is your name Taro Tanaka?”, short user answers such as “Yes” or “No” are expected, so the allowable pause length should be set to a few hundred milliseconds. In contrast, after the AI asks, “Is there anything you are taking care of for your health?”, the user is expected to think and speak for a long time, so the allowable pause length should be set to longer than one second. This eliminates the aforementioned trade-off and allows the system to respond to short user speech at a good tempo while still being able to hear user speech that contains pauses until the end.

5. Verification Experiments

Before providing the AI Phone Service commercially, NTT DOCOMO established a test environment and conducted verification experiments for two use cases.

5.1 Accepting Applications with Identity Verification

To confirm the effectiveness of reducing the telephone answering workload of businesses that provide monthly services, we conducted a verification experiment with the use case of accepting

an application over the phone. Since the application process here involves identity verification linked to a user database in a customer service system, we designed and applied the scenario shown in **Figure 3**. If the user can be uniquely identified by simply searching the name in the user database, acceptance is complete. Even if the user cannot be uniquely identified by name confirmation, customer number confirmation or address confirmation, acceptance is complete if the user can be uniquely identified by combining them with the confirmation of a fee payment. In this verification experiment, we obtained an acceptance dialogue completion rate of 77%. In the verification experiment, dialogue was carried out using voice input only. However, commercial systems also support dial key input which holds promise for dialogue completion rate of 88% when used in combination with voice input.

5.2 Monitoring the Elderly

Elderly people who live alone tend to have less communication with others and need support such as daily calls, but local support organizations do not have enough staff to take care of each elderly person. Thus, to verify whether AI telephone support can solve the problem of monitoring elderly people living alone and the burden on support organizations, we conducted a verification experiment involving automatically calling elderly people living alone at regular intervals to check on their health and safety. Over the phone, the AI asks the questions shown in **Table 1** and converses with the elderly person. Since it was difficult to quantitatively

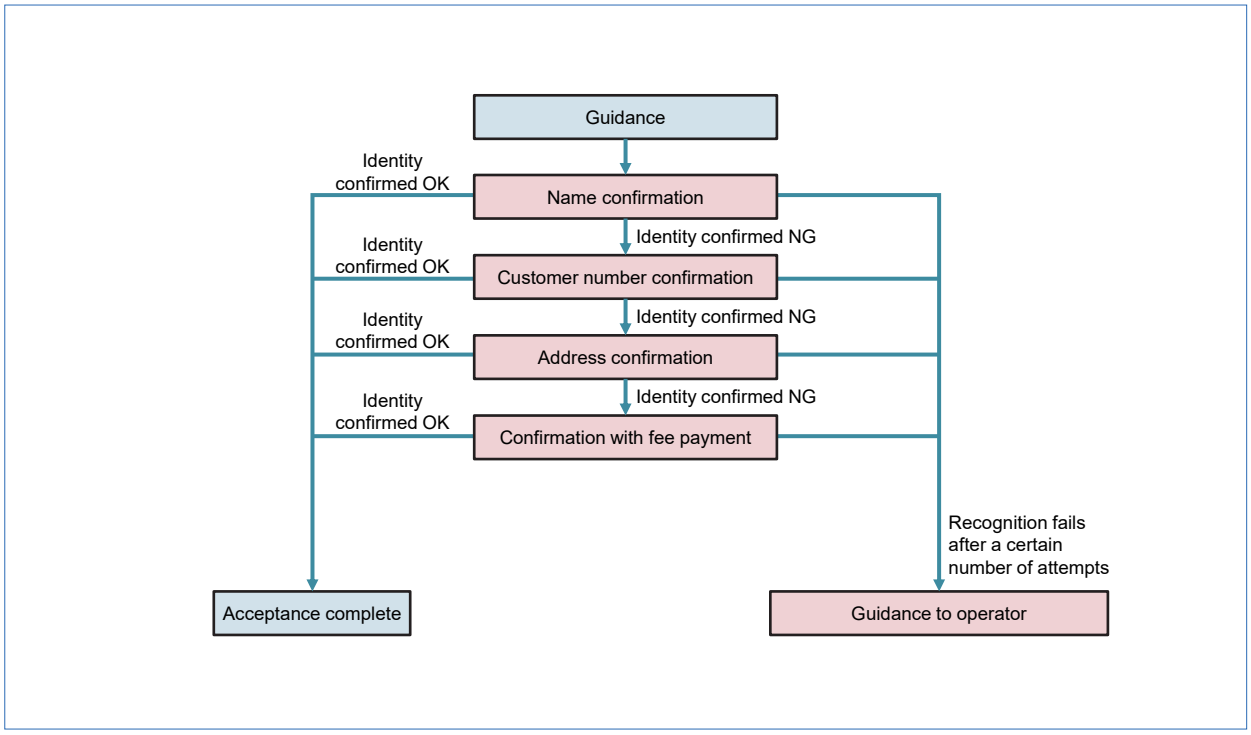


Figure 3 Accepting an application with an identity verification scenario

Table 1　Elderly monitoring questions

Item	Question
Sleep	Did you sleep well last night?
	What time did you go to bed last night?
	Did you wake up during the night?
	Is there anything else that is bothering you about your sleep?
Meals	Did you eat three meals yesterday?
	Have you eaten any protein this week, such as meat, fish, or eggs?
	Do you have an appetite?
	Is there anything else about your diet that concerns you?
Activities	Did you go out yesterday?
	Do you have any plans to go out today?
	Have you talked to your family and friends?
	Is there anything you would like to try next week?
	What would you like to do?
Physical condition	How are you feeling today?
	Have you had a bowel movement?
	Have you been to hospital recently?
	Is there anything particular that you do for your health?
	What kind of things do you pay attention to?
Body care and grooming	Did you take a bath yesterday?
	Do you soak in the bathtub?
	Do you take care of your teeth every day?
	Do you have any other concerns about body care and grooming?

measure the effects of monitoring the elderly with AI, we conducted interviews with the people subjected to the experiment. Based on the results of these interviews, we have been conducting a second verification experiment since February 2021.

6. Conclusion

In this article, we discussed the AI Phone Service to automate phone answering. These services have the potential to alleviate the problem of the shortage of human resources engaged in telephone

answering services. NTT DOCOMO began providing commercial AI Phone Service in December 2020 and plans to continue verification experiments and officially launch services for the use cases of accepting applications and reservations, and monitoring the elderly. Going forward, we will continue to work on the technological challenge of further improving speech recognition performance.

REFERENCES

[1] T. Hashimoto, et al.: “Improving Customer Satisfaction and Operator Efficiency in Call Centers Using AI - Speech Recognition IVR -,” NTT DOCOMO Technical Journal, Vol.19, No.4, pp.4-10, Apr. 2018.

[2] T. Oba, et al.: “docomo AI Agent Open Partner Initiative,” NTT DOCOMO Technical Journal, Vol.20, No.3, pp.4-9, Jan. 2019.

Event Reports | 5G | Open House | Exhibition Report

docomo Open House 2021 —The society of the future begins here. Hello, Transformation. —

R&D Strategy Department Masahiro Tamaoki†

"docomo Open House 2021 —The society of the future begins here. Hello, Transformation.—" was held online for four days from February 4 to 7, 2021. This article introduces the event and explains the details of the main exhibits.

1. Introduction

For four days from February 4 to 7, 2021, NTT DOCOMO held "docomo Open House 2021 —The society of the future begins here. Hello, Transformation. —" online.

This article describes details of the main exhibits at this event.

2. Online Exhibition on a Web Page

In light of the recent social situation, this event was held online. 233 exhibits from NTT DOCOMO and its partners were presented under the title of "Tech Showcase." Various lectures and seminars titled "docomo Open House TV" were held in conjunction with the exhibition, with the participation of not only NTT DOCOMO executives but also celebrities from outside the company. A rich variety of 74 contents were released for those not familiar with technologies such as the 5th Generation mobile communication system (5G) and AI to enjoy, such as lectures on solving future social issues and creating new value. During the four days of the exhibition, the total number of visitors exceeded 90,000, making it a great success (**Figure 1**).

In Tech Showcase, the details of technologies and their user benefits were introduced in individual

† Currently, Communication Device Development Department.

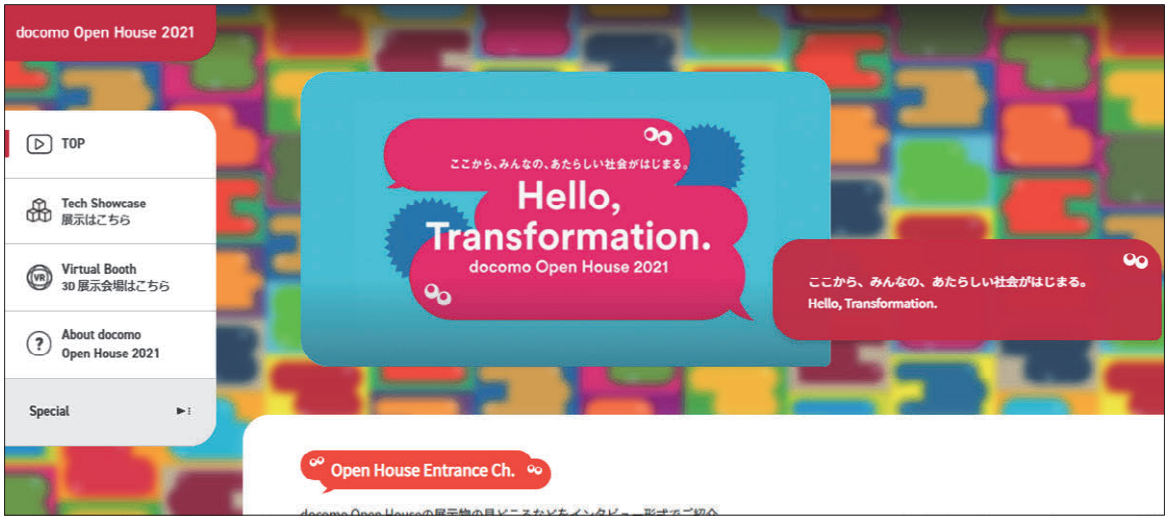


Figure 1 docomo Open House 2021 entrance page

template Web pages. Some exhibits offered interactive functions such as viewing live demonstrations through a Web conference system, chatbots, comment posting from visitors, and exchange of business card information using registered information. In addition, compared to exhibitions held in real venues, we were able to promote contents through a range of videos without the limitations of conventional displays and exhibition stands (**Figure 2**).

Similarly, we were able to provide lectures and seminars without being restricted by venue size or time. By delivering seven videos simultaneously, users could watch various lectures and seminars with the sense of zapping between them. In addition, lectures and seminars already completed could be delivered as so-called "missed deliveries," which do not restrict user viewing time. In addition, as some performances are possible with online distribution, NTT DOCOMO President and CEO, Motoyuki Ii appeared via Volumetric Capture*1 in his welcome speech (**Figure 3**). Meanwhile, in his speech titled "Digitization Creating a New Society," Executive Vice President and Executive General Manager of R&D Innovation Division, Naoki Tani introduced NTT DOCOMO initiatives to contribute to the realization of a society where the new normal*2 is an environment in which people can work and live safely and securely through the evolution of advanced technologies and ecosystems. Also, to make these NTT DOCOMO initiatives more enjoyable, a variety of lectures were held on themes that transcend the boundaries of partners, including technology commentary program by celebrities called "What's This Tech?".

*1 Volumetric Capture: A technology that converts images captured by a camera, etc., into three-dimensional digital data and reproduces the images in 3D space.

*2 New Normal: A state in which a new common sense has irreversibly taken hold as a result of changes in the social environment and circumstances.

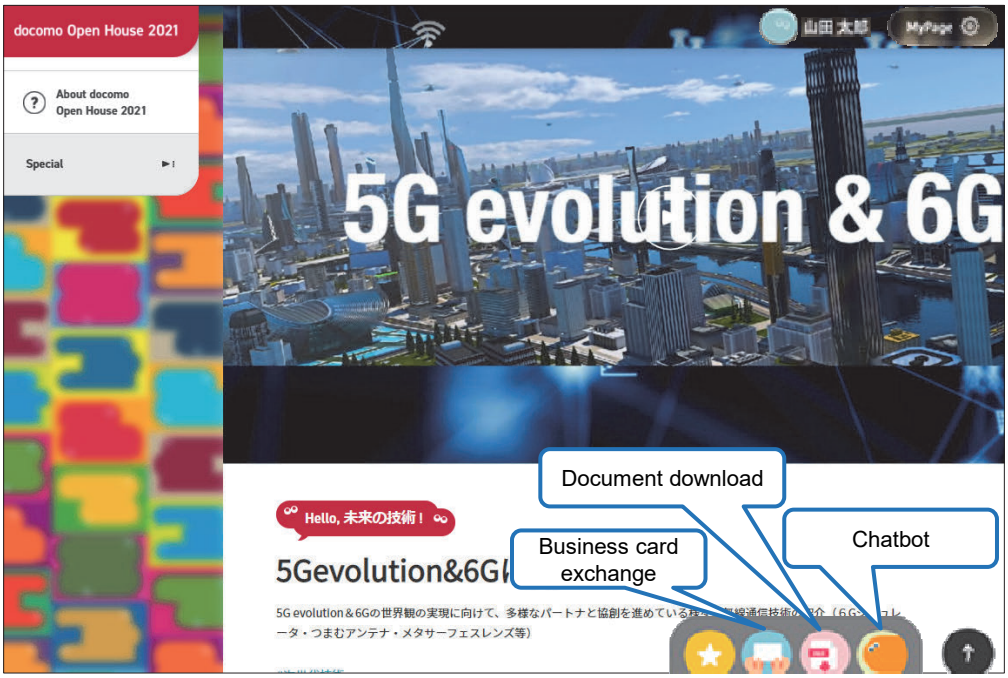


Figure 2 docomo Open House 2021 Tech Showcase

Figure 3 Welcome speech

3. VR Exhibition on a Smartphone App

At the same time as the aforementioned Web page exhibition, we also held a VR exhibition entitled "Virtual Booth" in which visitors could experience immersive contents using NTT DOCOMO technologies and solutions using the Virtual Event Platform developed by NTT DOCOMO (**Figure 4**).

In this Virtual Booth, volumetric video technology delivered rich and realistic limited-edition content via a smartphone app of 3D productions adding objects and models, etc., to images of celebrities and badminton players (**Figure 5**). This technology not only faithfully reproduces 3D models of people and objects photographed from all angles using special equipment in VR space, but also digitizes

Figure 4 Virtual Booth

Figure 5 Main booth of Virtual Booth

the movements of the subjects with high precision, enabling viewing of realistic VR content from any 360-degree viewpoint.

4. Conclusion

This article introduced the "docomo Open House 2021 —The society of the future begins here. Hello, Transformation. —" event held for four days from February 4 to 7, 2021, and described its exhibits.

NTT DOCOMO will create fun and surprising services that will revolutionize user lifestyles and communications for the new society of the future. We would also like to work on solving social issues with the aim of realizing growth and a prosperous society in Japan.

NTT DOCOMO
Technical Journal Vol.23 No.1

Editorship and Publication
NTT DOCOMO Technical Journal is a quarterly journal edited by NTT DOCOMO, INC. and published by The Telecommunications Association.

Editorial Correspondence
NTT DOCOMO Technical Journal Editorial Office
R&D Strategy Department
NTT DOCOMO, INC.
Sanno Park Tower
2-11-1, Nagata-cho, Chiyoda-ku, Tokyo 100-6150, Japan
e-mail: dtj@nttdocomo.com