

# Work Style Reform by Serverless Architecture

2018/11/16

**SIOS Technology, Inc.**  
**Engineering Department**  
**Noriyuki TAKEI**



# About me



# About me

## about me



# Noriyuki TAKEI

## Information

- SIOS Technology, Inc.
- Cloud Native Application
- Living in Chiba Prefecture

## Favorites

- Azure
- Squash
- Sweets

## blog

<https://tech-lab.sios.jp/>

# What is Serverless Architecture?

# What is Serverless?

Serverless is Application Infrastructure  
which has 3 advantages as below.



**No need for  
physical servers**



**Auto-Scaling  
if necessary**



**Pay  
as you go**

# What is Serverless Architecture?



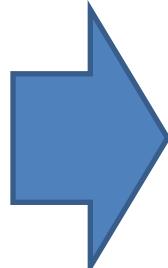
## No need for physical servers

You can reduce the time and effort when developing Applications



**before**

- Creating Virtual Machine
- Installing OS
- Installing Middleware(Apache and so on)
- Evaluating Performance
- Coding



**after**

All you have to do is only 2 things, inputting your code on your browser and saving it!!

# What is Serverless Architecture?



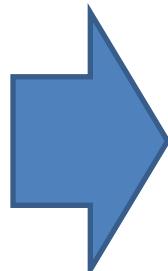
## Auto-Scaling if necessary

Serverless has unlimited scalability!!



**before**

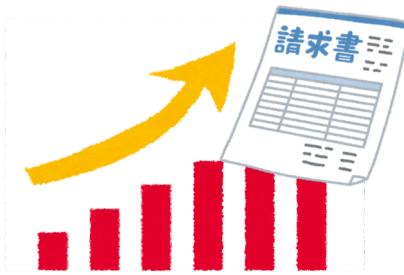
- Preparing Stress Test Tool
- Creating Stress Testing Scenarios
- Doing Testing
- Performance tuning
- Creating Stress Test Report



**after**

Serverless is infinitely scalable with no special configuration!!

# What is Serverless Architecture?

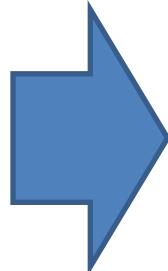


## Pay as you go!!

## Pay as you go!!



Virtual Machines will be charged even if no process invokes.



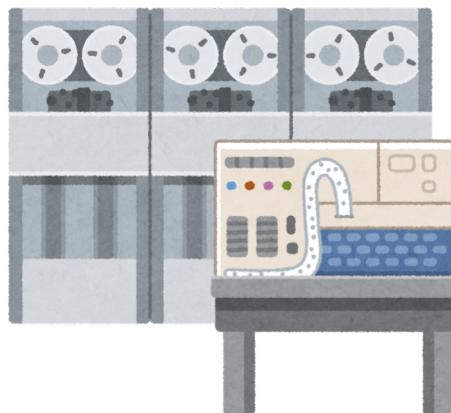
after

Serverless will be charged when any process is invoking, so it is very economical.

# What is Serverless Architecture?



Serverless provides  
Drastic Transformation  
to Application Development!!



# What is Serverless Architecture?



## Lift & Shift

### On-Premises

#### Physical Servers



•  
•  
•



### Public Cloud

#### Virtual Machines



•  
•  
•



# What is Serverless Architecture?



## Cloud Native

### On-Premises

#### Physical Servers



•  
•  
•



### Public Cloud



#### API Management



#### Azure Functions



#### CosmosDB

# What is Serverless Architecture?



## Lift & Shift

Lift & Shift is to migrate physical servers on On-premises to Public Cloud and keep their original. As a result, we can use OS, Middleware, Application in the same way before migration.

## Cloud Native

Cloud Native is to develop Applications with specific features of Public Cloud, for example, Azure Functions, API Management and so on. NoOps Environment provided by Serverless will save us from applying security patches and upgrading Middlewares.

# What is Serverless Architecture?



Serverless is the revolutionary way to develop applications by Cloud Native.

In the future, Cloud Native will be most frequently used when developing applications

We must change the way of designing systems.

# The difference between IaaS & PaaS & FaaS



# The difference between IaaS & PaaS & FaaS



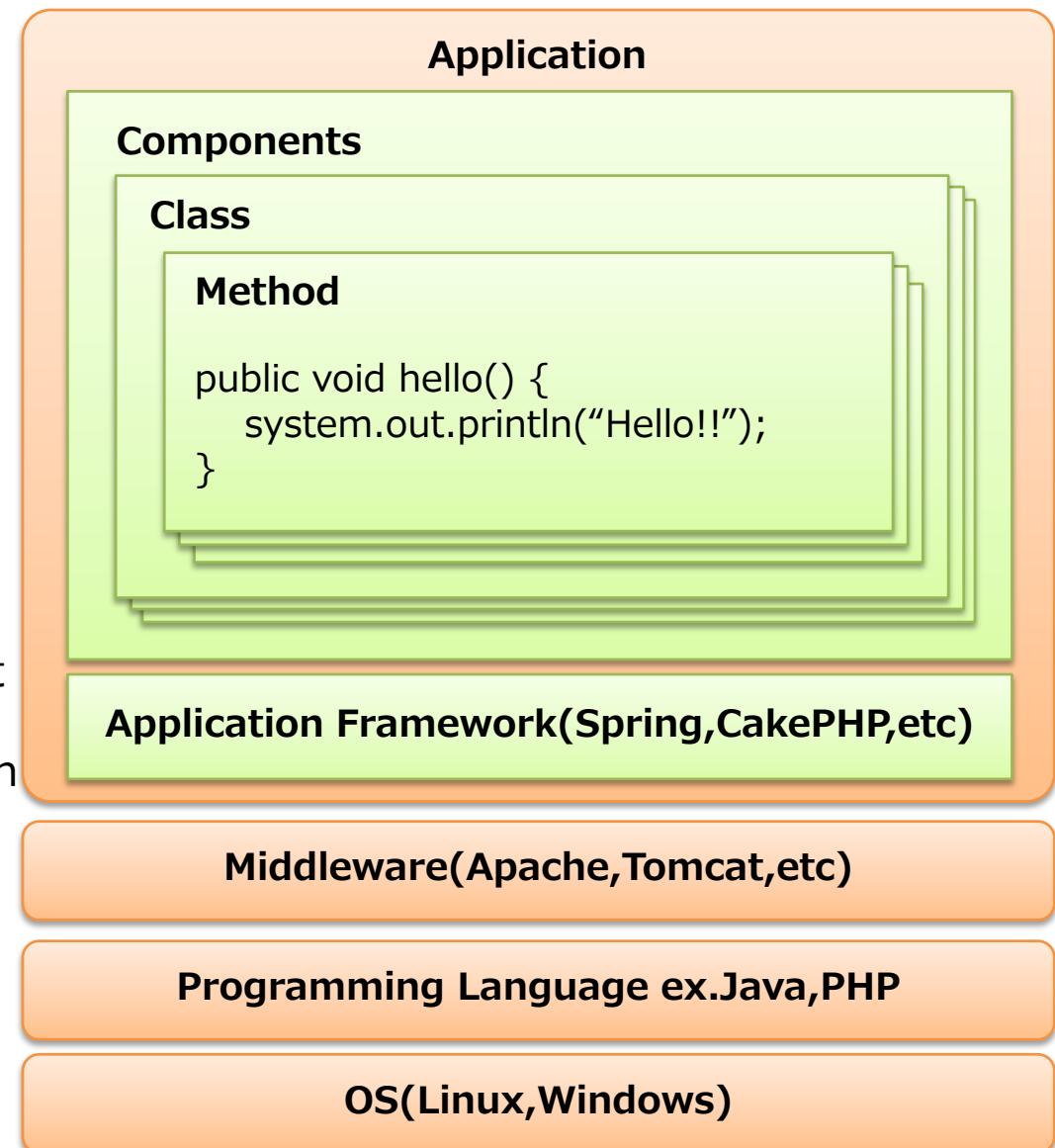
## IaaS

already OS installed Platform

System Administrator must manage the layer above OS.

System Administrator must protect Applications against security threats(vulnerabilities in Application Framework,OS,Middleware)

Any programming language and middleware are executable in IaaS.



# The difference between IaaS & PaaS & FaaS

## Development Process using IaaS

Create Virtual Machines.



Install Runtimes such as Java,PHP and so on.



Install Middlewares such as Apache,Tomcat and so on.



Develop Applications using Application Framework such as Spring Boot,CakePHP and so on.



Build Applications using Maven,gradle and so on.



Deploy Applications built by Build Tools such as Maven.

# The difference between IaaS & PaaS & FaaS



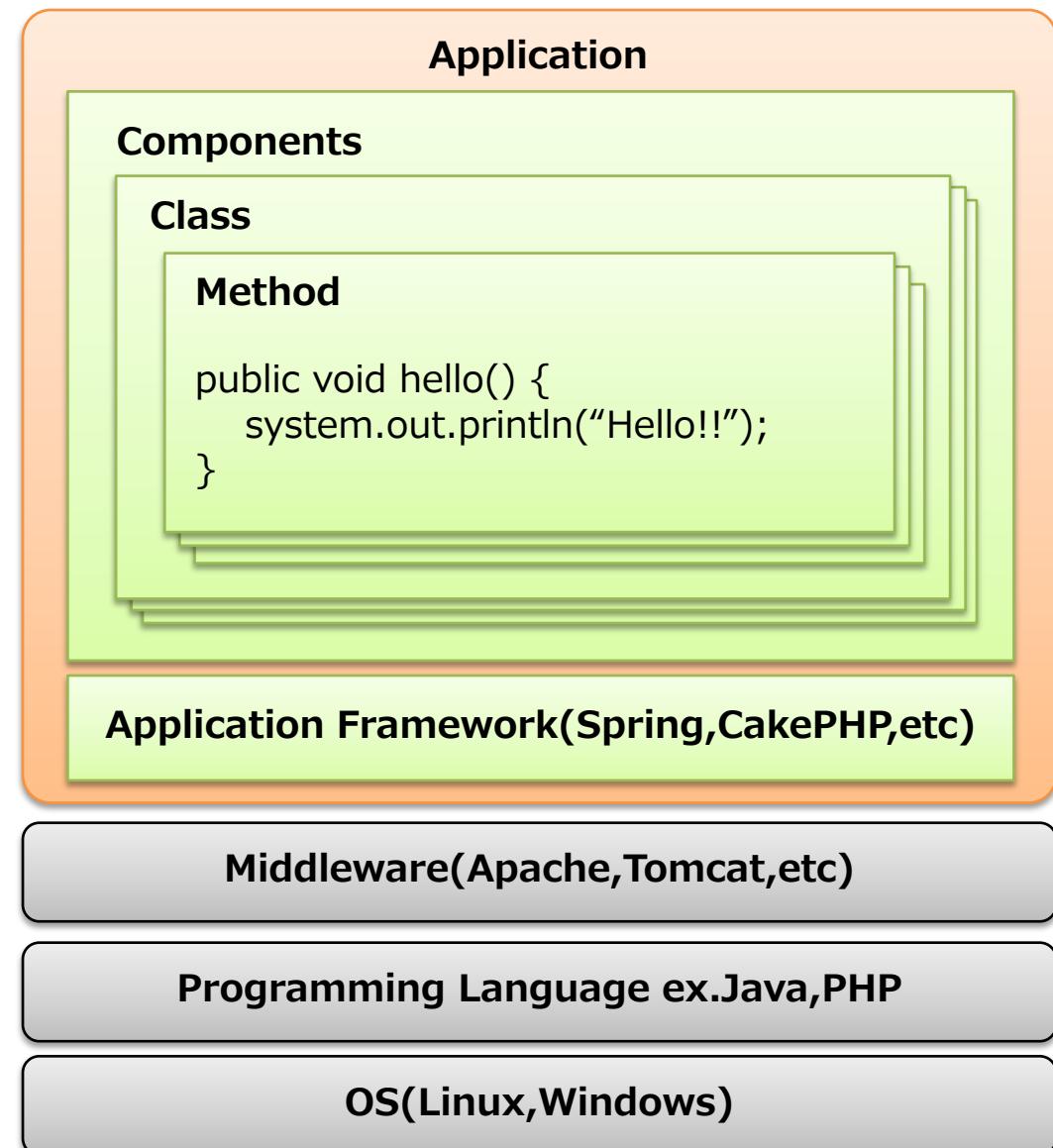
## PaaS

System Administrator is responsible for OS, Runtime and Middleware that Public Cloud was once responsible for.

Administrators of Public Cloud are responsible for Protection for vulnerabilities in Middleware and applying security patches for operating system

System Administrator is responsible for vulnerabilities in Application Framework such as CakePHP, Spring Boot and so on.

IaaS is not flexible as PaaS when developing applications. You can use Runtimes only provided by Public Cloud.



# The difference between IaaS & PaaS & FaaS



## Development Process using PaaS

Develop Applications using Application Framework such as Spring Boot,CakePHP and so on.



Build Applications using Maven,gradle and so on.



Uploading Built Applications compressed by ZIP via Administration Portal of Public Cloud.

# The difference between IaaS & PaaS & FaaS

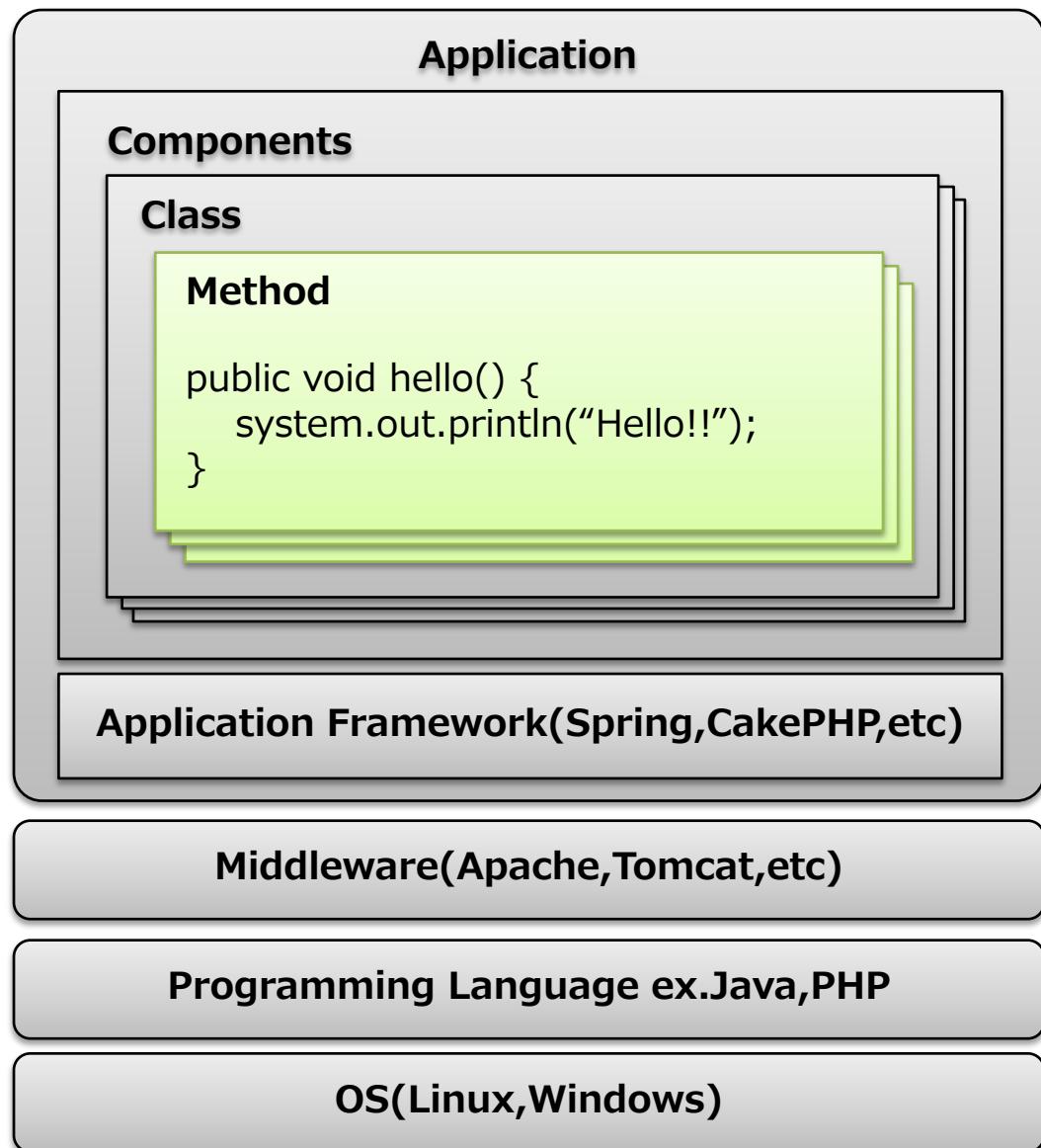


## Serverless Architecture

You do not need OS, Runtime, Middleware, Application Framework.

You don't have to do anything else but writing code to run Applications.

You don't have to protect for vulnerabilities in Middleware and applying security patches for operating system.



# The difference between IaaS & PaaS & FaaS



## Development Process of Serverless Architecture

Access Public Cloud Portal.



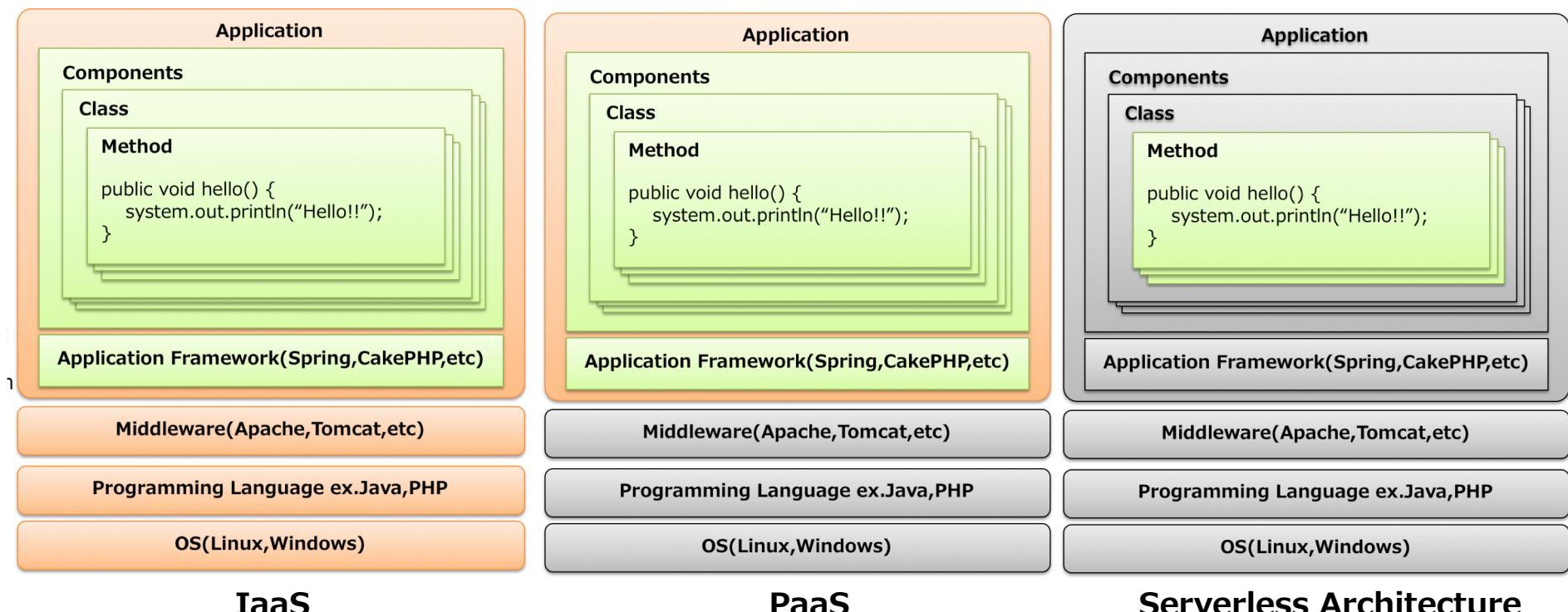
Write Code on your Browser.

# Only these!!



# The difference between IaaS & PaaS & FaaS

I arranged the images introduced earlier. As you can see, you'll find out that the range to manage by system Administrator becomes narrow in the order of IaaS→PaaS→Serverless Architecture



# Advantages and disadvantages of Serverless Architecture

# Advantages and disadvantages of Serverless Architecture



Serverless Architecture is not suitable for all the applications.



# Advantages and disadvantages of Serverless Architecture



## Many Constraints



Stateless



Difficulty of designing systems



Migration Ability

# Advantages and disadvantages of Serverless Architecture



	IaaS	PaaS	Serverless Arichitecture
Maintenancibility	△	○	◎
Flexibility	◎	○	△
Man-hours	△	○	◎
Cost	○	○	◎
Scalability	△	○	◎
Migration ability	◎	○	△

# Advantages and disadvantages of Serverless Architecture



## Maintenancibility

	IaaS	PaaS	Serverless Architecture
Maintenancibility	△	○	◎

Maintenancibility means the efficiency of the system to keep Applications work well.

To keep Maintenancibility efficient,we must apply patches,protect our systems from vulnerabilities,and so on,moreover those problems will occur irregularly.

Serverless Architecture has a great advantage!!

# Advantages and disadvantages of Serverless Architecture



## Maintenability

IaaS

System Administrators must manage systems above Operating system layer, apply patches and protect systems from vulnerabilities.

PaaS

Public Cloud Administrators are responsible for the layer below Application, but System Administrators must manage Application, for instances, apply patches for Application Framework such as CakePHP, Spring Boot and so on.

 Serverless Architecture

System Administrators can be free from managing infrastructures such as OS, Middleware, Network and so on.

# Advantages and disadvantages of Serverless Architecture



## Flexibility

	IaaS	PaaS	Serverless Architecture
Flexibility	◎	○	△

Flexibility means the flexibility to choose Runtimes,to design Network and Storage.

IaaS has a great advantage!!

# Advantages and disadvantages of Serverless Architecture



## Flexibility



### IaaS

System Administrators can install any languages, choose any Storage Protocols (CIFS, NFS, WebDav, iSCSI, etc), in the same way as On-Premise Environment.

### PaaS

System Administrators can manage only the layer above Runtimes, so they are not so free as IaaS. Only Runtimes provided by Public Cloud are executable.

### Serverless Architecture

System Administrators can choose Runtimes provided by Public Cloud, but the execution unit of code being "function" prevent them from designing systems with advantages of Application Framework and useful libraries.

# Advantages and disadvantages of Serverless Architecture



## Man-hours

	IaaS	PaaS	Serverless Architecture
Man-hours	△	○	◎

Man-hours is the time required to develop Applications, including the time to install OS, Middleware, Runtime and so on.

Serverless Architecture has a great advantage!!

You should not ignore the time required to learn about the technologies around Serverless.

# Advantages and disadvantages of Serverless Architecture



## Man-hours

### IaaS

To develop Applications requires many tasks to do such as designing and building Middleware, investigating Frameworks and so on.

### PaaS

Fewer tasks than IaaS is required, for example, choosing Application Framework.etc above application layer.

### Serverless Architecture

The only one thing to do for developing Applications is to write code on your browser via Public Cloud Portal.

# Advantages and disadvantages of Serverless Architecture



You should not ignore the time required to learn about the technologies around Serverless.



# Advantages and disadvantages of Serverless Architecture



The Constraint of "Stateless" keeps it difficult to design systems.



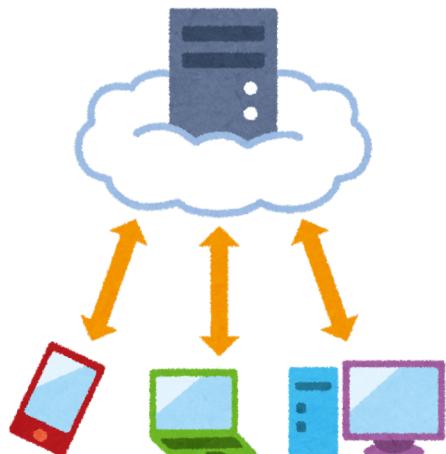
# Advantages and disadvantages of Serverless Architecture



What is “Stateless”?

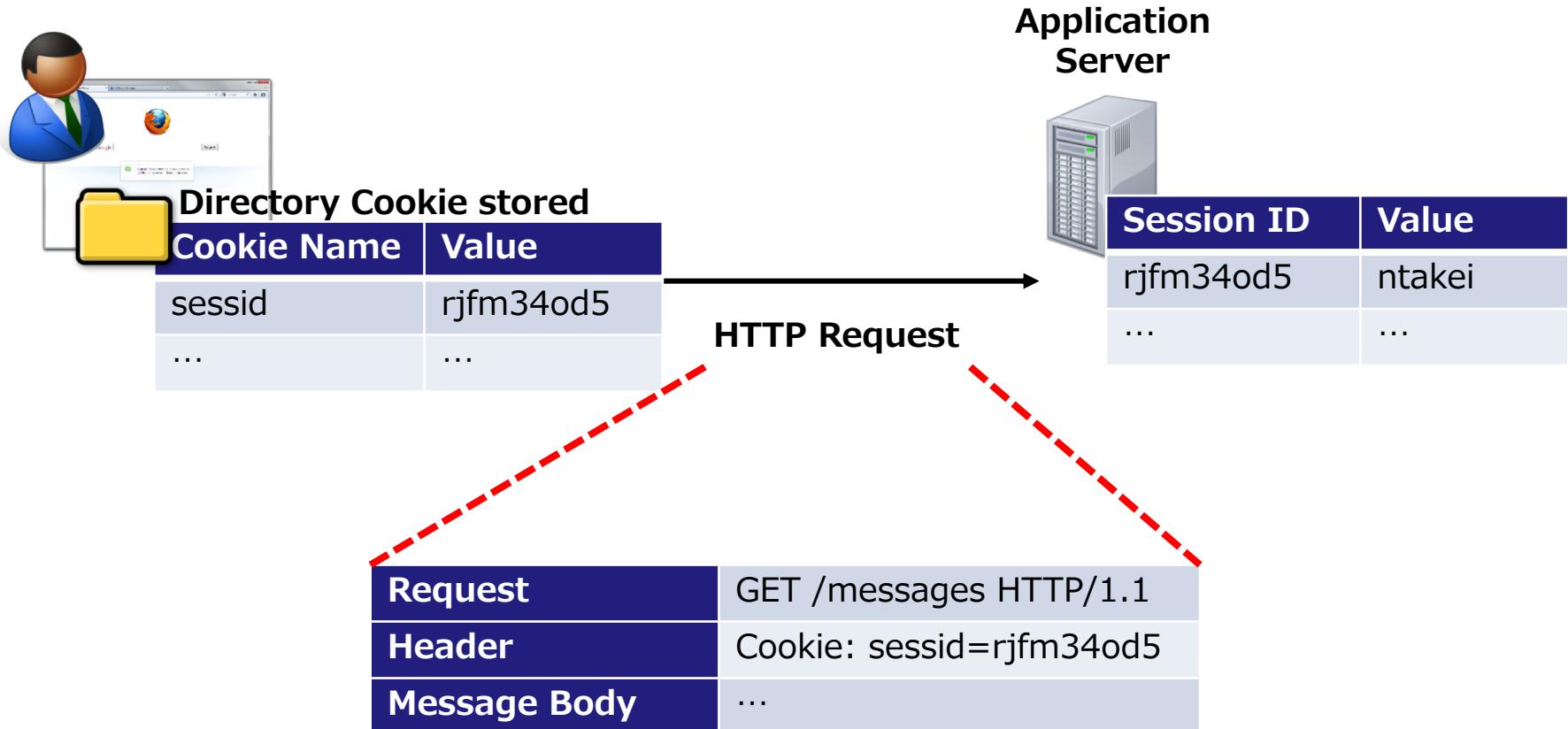


Server has no state.



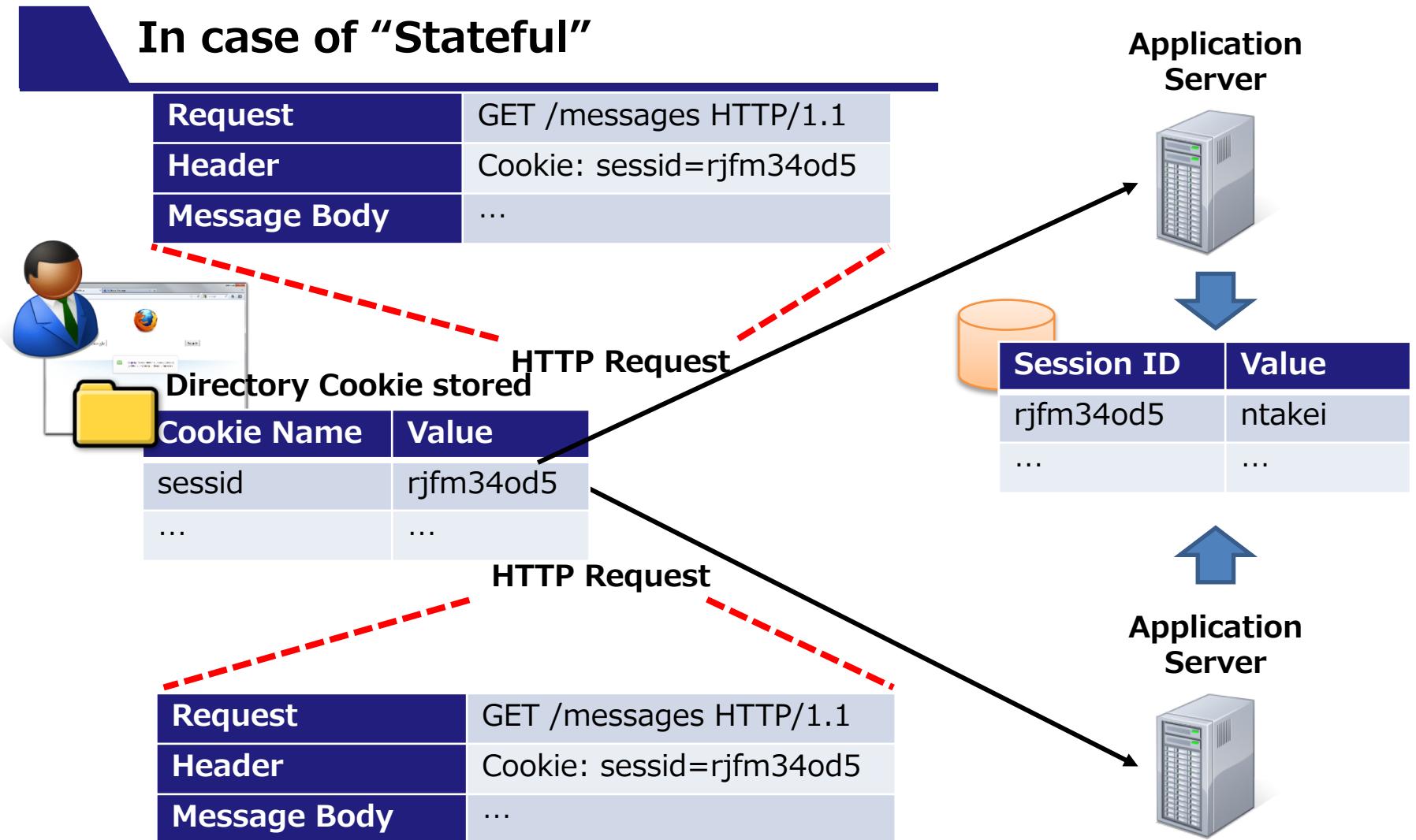
# Advantages and disadvantages of Serverless Architecture

## In case of “Stateful”



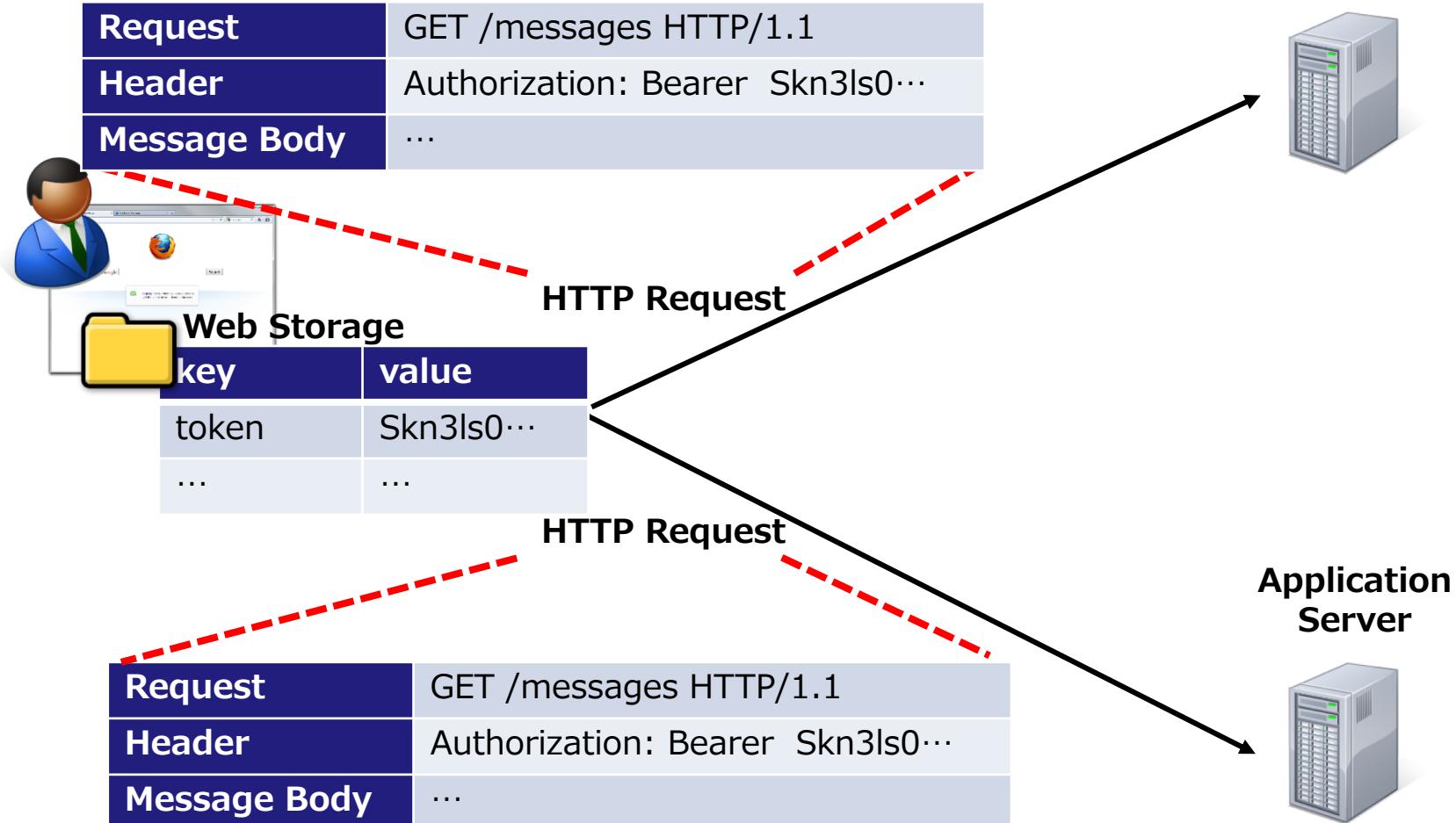
# Advantages and disadvantages of Serverless Architecture

## In case of “Stateful”



# Advantages and disadvantages of Serverless Architecture

## In case of “Stateless”



# Advantages and disadvantages of Serverless Architecture



In case of Stateless,it's impossible to authenticate user by using Cookie.

You need to change Authentication methods such as JSON Web Token,OpenID Connect and so on.In this case,Authentication State is carried by the client.

You must learn about the technologies around Serverless.

# Advantages and disadvantages of Serverless Architecture



## Cost

	IaaS	PaaS	Serverless Architecture
Cost	○	○	◎

The Cost is one for running Azure Services such as Virtual Machines, MySQL and so on.

Serverless Architecture has a great advantage!!

# Advantages and disadvantages of Serverless Architecture



## Cost

### IaaS

The cost always increase if applications in a virtual machine are running or not, therefore it's very wasteful.

### PaaS

It's the same as IaaS.

### Serverless Architecture

The cost will arise only when functions are running.

# Advantages and disadvantages of Serverless Architecture



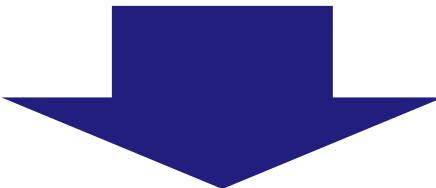
## Comparison result between IaaS and Serverless

IaaS

¥9,821

 Serverless  
Architecture

¥694.4



**Serverless has a great advantage!!**

# Advantages and disadvantages of Serverless Architecture



## Scalability

	IaaS	PaaS	Serverless Architecture
Scalability	△	○	◎

Scalability is an attribute that describes the ability of a process, network, software or organization to grow and manage increased demand.

Serverless has a great advantage!!

# Advantages and disadvantages of Serverless Architecture



## Scalability

IaaS

In order to achieve the scale-out,to build a Load Balancer,configuration of scale-out and so on are required.In addition,Scale-out is limited to what is preconfigured.

PaaS

It's the same as IaaS.

 Serverless Architecture

Serverless Architecture has an unlimited Scalability!!

# Advantages and disadvantages of Serverless Architecture



## Migration Ability

	IaaS	PaaS	Serveless Architecture
Migration Ability	◎	○	△

Migration Ability is an attribute that describes the ability of moving Applications to other public cloud.

IaaS has a great advantage!!

# Advantages and disadvantages of Serverless Architecture



## Migration Ability



### IaaS

Because System Administrators can manage all the layers from OS to application, applications can be migrated to other public cloud as it is.

### PaaS

Compared with IaaS, PaaS has some constraints when migrating. For instances, Runtimes to use are limited to ones provided by other public cloud.

### Serverless Architecture

Implementations of Serverless Architecture strongly depend on the way provided by public cloud, so it's very difficult to migrate to other public cloud.

# Advantages and disadvantages of Serverless Architecture



What is the most useful case to use Serverless Architecture?



# Advantages and disadvantages of Serverless Architecture



There is a possibility that many sudden and unexpected requests may occur.

To save time to apply security patches and update libraries(such as OpenSSL)

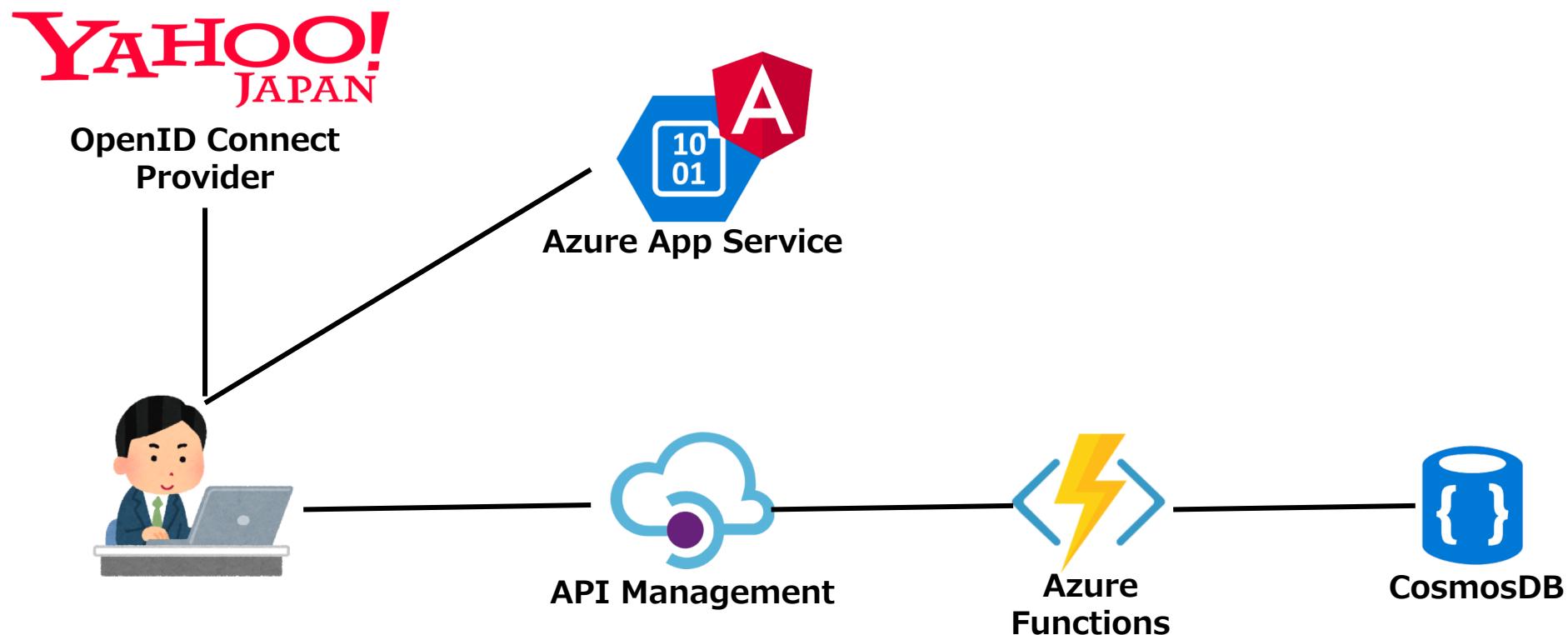
To reduce the cost of Public Cloud.

# Usage Scenarios



# Usage Scenarios

## Single Page Application



# Usage Scenarios

## Single Page Application

### Goal

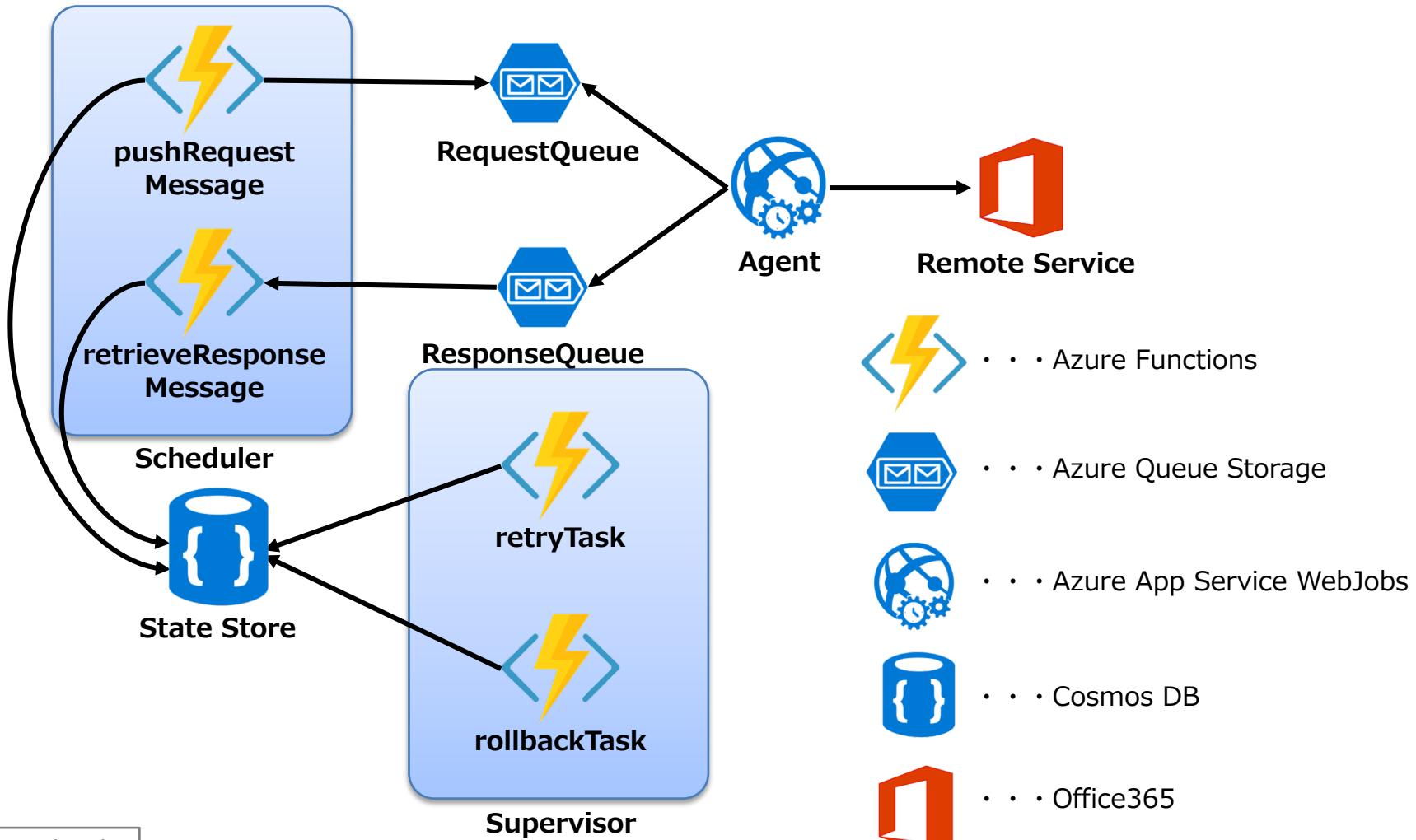
To achieve "LessOps" by reconstruction of applications which consists of 3 tier(Web,App,DB) using Serverless

### Implementation

View is Java Script Framework "Angular" which achieves SPA, stored by Azure Blob Storage, Authentication Method is OpenID Connect, Authentication of API is "API Management", Application Logic is "Azure Functions", Data Store is CosmosDB.

# Usage Scenarios

## Scalable NoOps Batch Infrastructure



# Usage Scenarios

## Scalable NoOps Batch Infrastructure

### Goal

To aim improvement of scalability and NoOps using Serverless Architecture as Application Infrastructure and Azure storage queue as intersystem communications.

### Implementation

Application Logic is "Azure Functions", Data Store is CosmosDB, intersystem communication is "Azure Queue Storage", request method to remote service is "Azure App Service".

# Durable Functions



# What is Durable Functions?

Extension of Azure Functions

Executing a sequence of functions in a particular order.  
Often the output of one function needs to be applied to  
the input of another function.

Timer-triggered Functions and Stateful Workflow.

## Limits of Serverless



Stateless

timeout for functions is 5 minutes



**It's impossible to develop applications which consists of complex processes.**

# Durable Functions is So great Solution!!

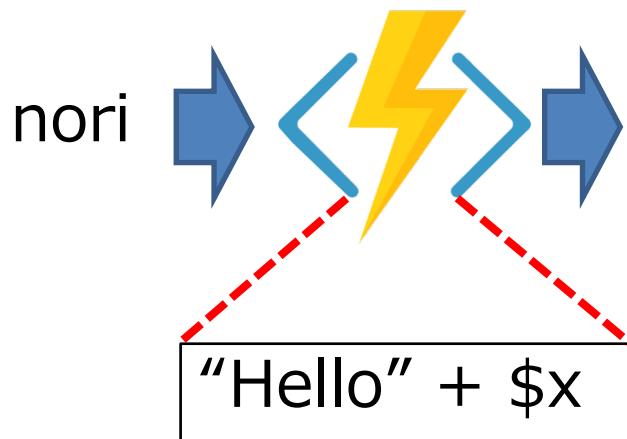


# Durable Functions

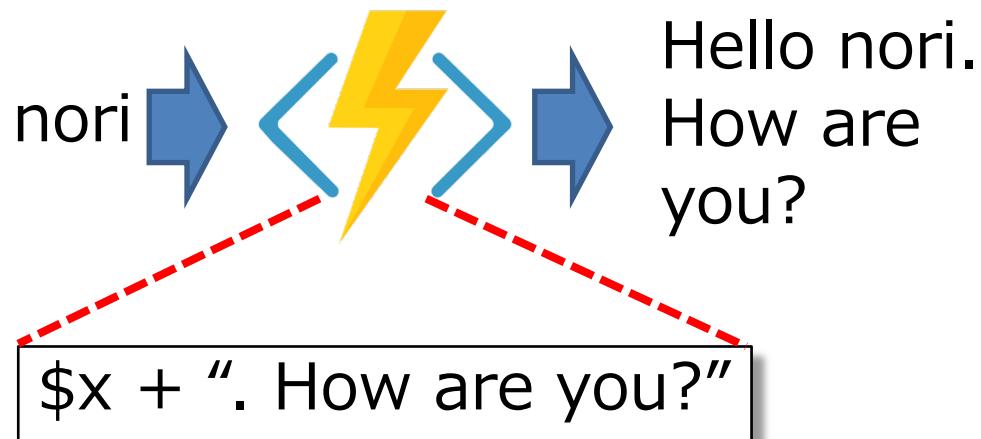
## Function chaining

Function chaining refers to the pattern of executing a sequence of functions in a particular order. Often the output of one function needs to be applied to the input of another function.

### sayHello Function



### sayHowAreYou Function



## Function chaining(Implementation)

```
const df = require("durable-functions");

module.exports = df.orchestrator(function*(ctx) {

    const x = yield ctx.df.callActivity("sayHello","nori");
    const y = yield ctx.df.callActivity("sayHowAreYou", x);

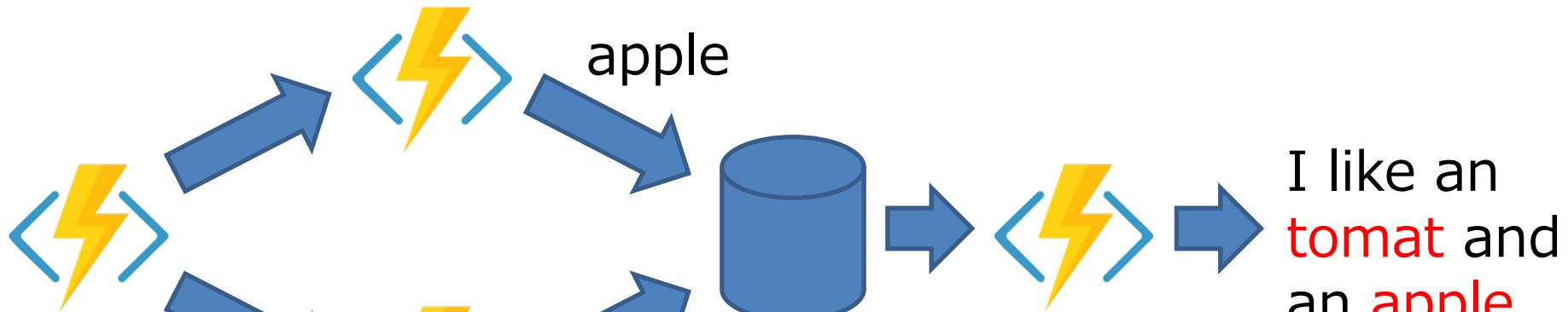
    return y;
});
```

# Durable Functions

## Fan-out/fan-in

Fan-out/fan-in refers to the pattern of executing multiple functions in parallel, and then waiting for all to finish. Often some aggregation work is done on results returned from the functions.

### **favoriteFruits Function**



### **favorite**

### **Vegetables Function**

## Fan-out/fan-in(Implementation)

```
const df = require("durable-functions");

module.exports = df.orchestrator(function* (ctx) {

    const parallelTasks = [];

    parallelTasks.push(ctx.df.callActivity("favoriteFruits"));
    parallelTasks.push(ctx.df.callActivity("favoriteVegetables"));

    const results = yield ctx.df.Task.all(parallelTasks);

    ctx.log('I like an ' + results[0] + ' and an ' + results[1]);
});
```

# Durable Functions

## Human interaction

Many processes involve some kind of human interaction. The tricky thing about involving humans in an automated process is that people are not always as highly available and responsive as cloud services. Automated processes must allow for this, and they often do so by using timeouts and compensation logic.



# Durable Functions



## Human interaction(Implementation)

```
const df = require("durable-functions");
const moment = require('moment');

module.exports = df.orchestrator(function* (context) {
    yield context.df.callActivity("gatherCode");

    const expiration = moment.utc(context.df.currentUtcDateTime).add(90, 's');
    const timeoutTask = context.df.createTimer(expiration.toDate());

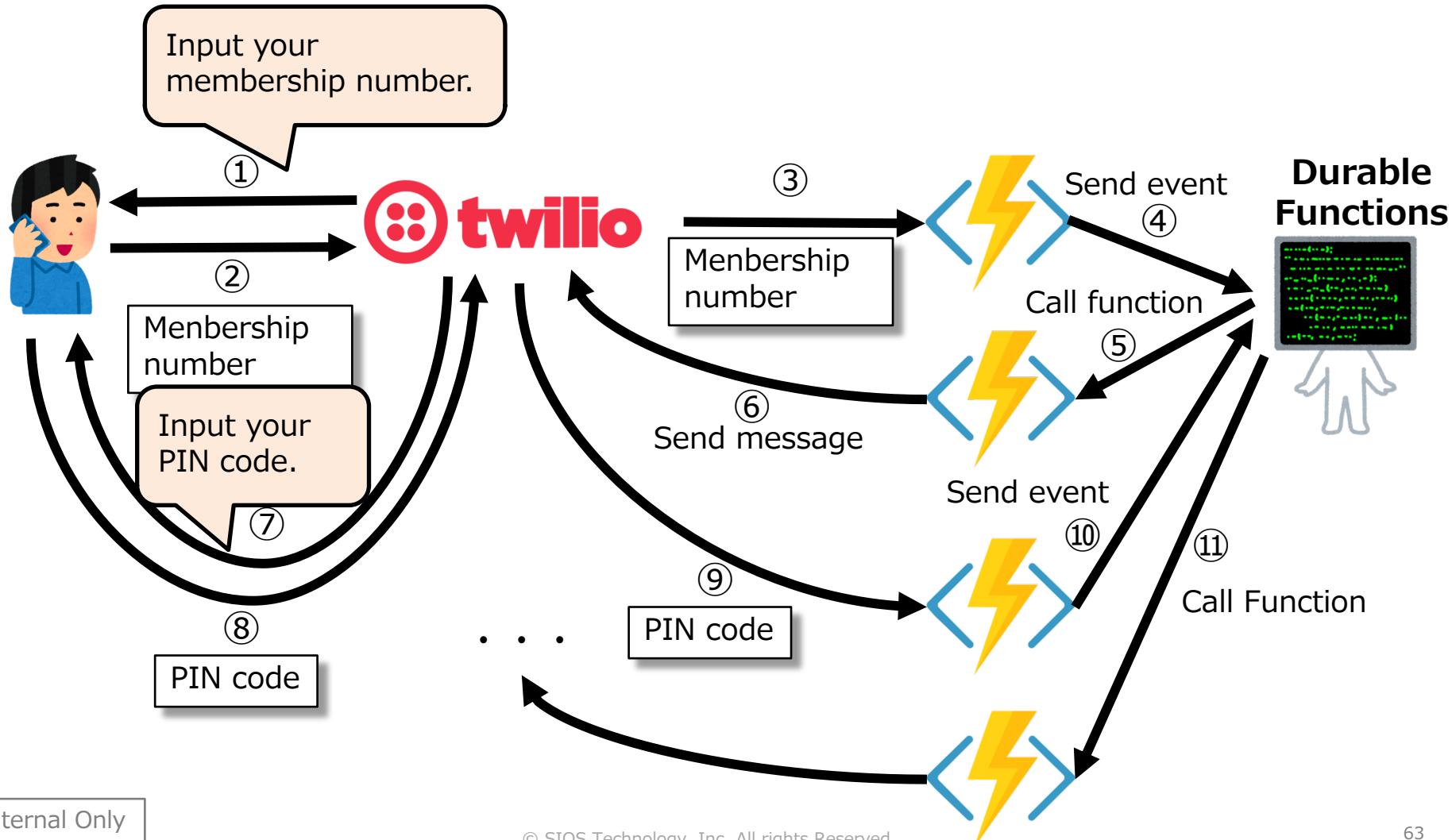
    const challengeResponseTask = context.df.waitForExternalEvent("Response");

    const winner = yield context.df.Task.any([challengeResponseTask, timeoutTask]);

    if (winner === challengeResponseTask) {
        // Do Smething
    }
});
```

# Durable Functions

## Usecase (Interactive Voice response)



# Alarm Clock by Durable Functions

## Work Style Reform Law

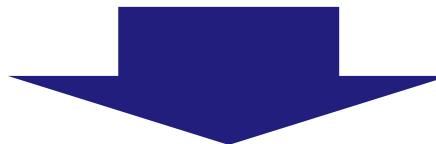
The Law principally addresses issues related to work hours and to the disparity in treatment between regular and nonregular employees.

This Law consists of 3 important points as below.

- Upper limit to permitted overtime hours
- Fair treatment of employees, regardless of employment status
- Introduction of “highly skilled professionals” exemption

## Work Style Reform Law

"Work Style Reform Law" allows everyone to work freely in accordance with their own life, so everyone can go to the office when they want. To provide an alarm device which promote natural and refreshing awaking is required.



I developed an alarm clock by durable functions which prevent you from going back to sleep!!

## Overview

At your specified time, the announcement to your specified phone number will go out.



Enter the 10 digit number announced by alarm-clock.



The Flow will be finished if you enter the valid number, or not , the announcement of entering the 10 digit number to your specified phone number will go out again.

## Technologies

### Azure Functions

Azure Functions is a solution for easily running small pieces of code, or "functions," in the cloud.

### Durable Functions

Durable Functions is an extension of Azure Functions and Azure WebJobs that lets you write stateful functions in a serverless environment.

### Twilio

Cloud communications platform for building SMS, Voice & Messaging applications on an API.

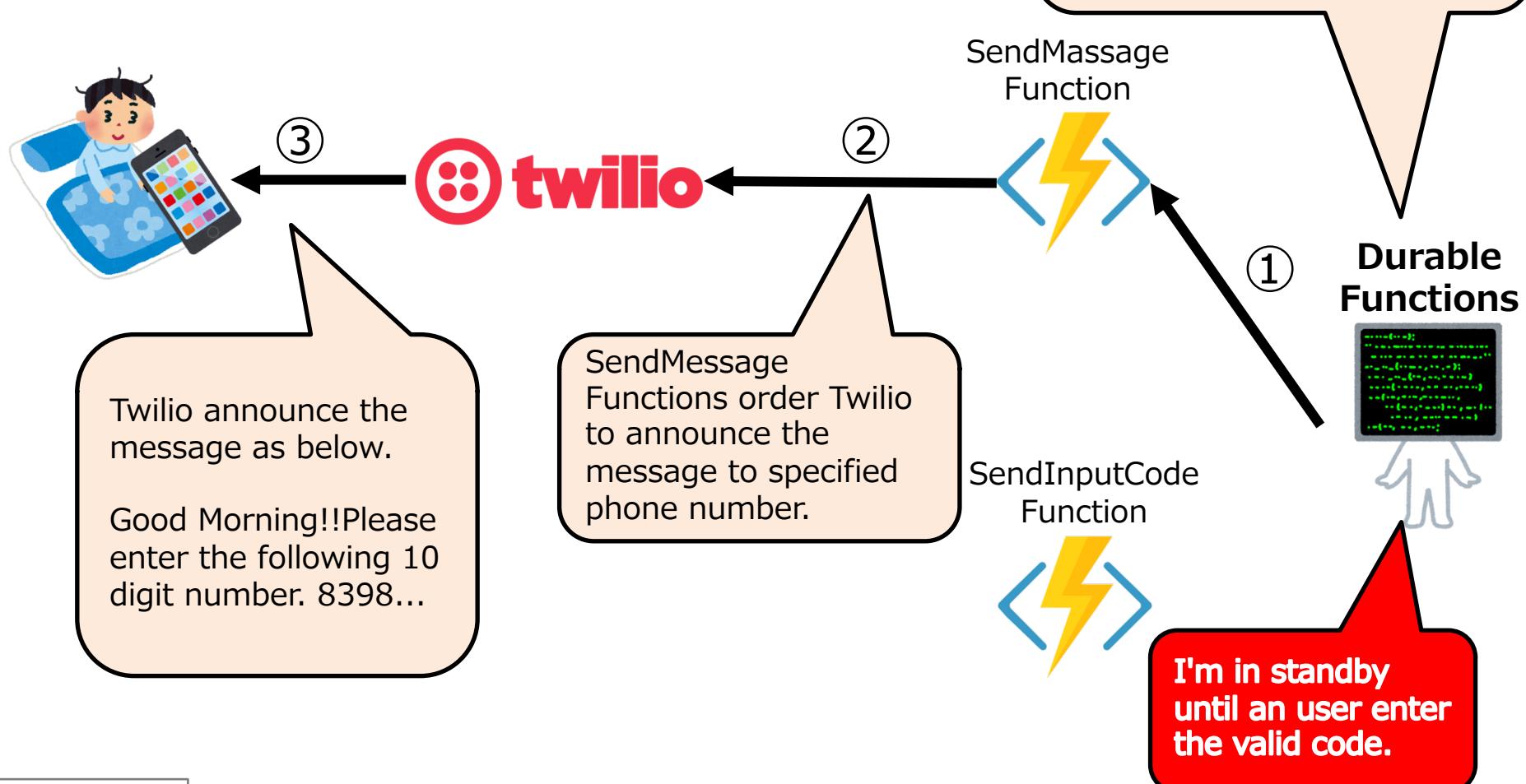
### Open JTALK

Open JTALK is a Japanese text-to-speech system.

# Alarm Clock by Durable Functions



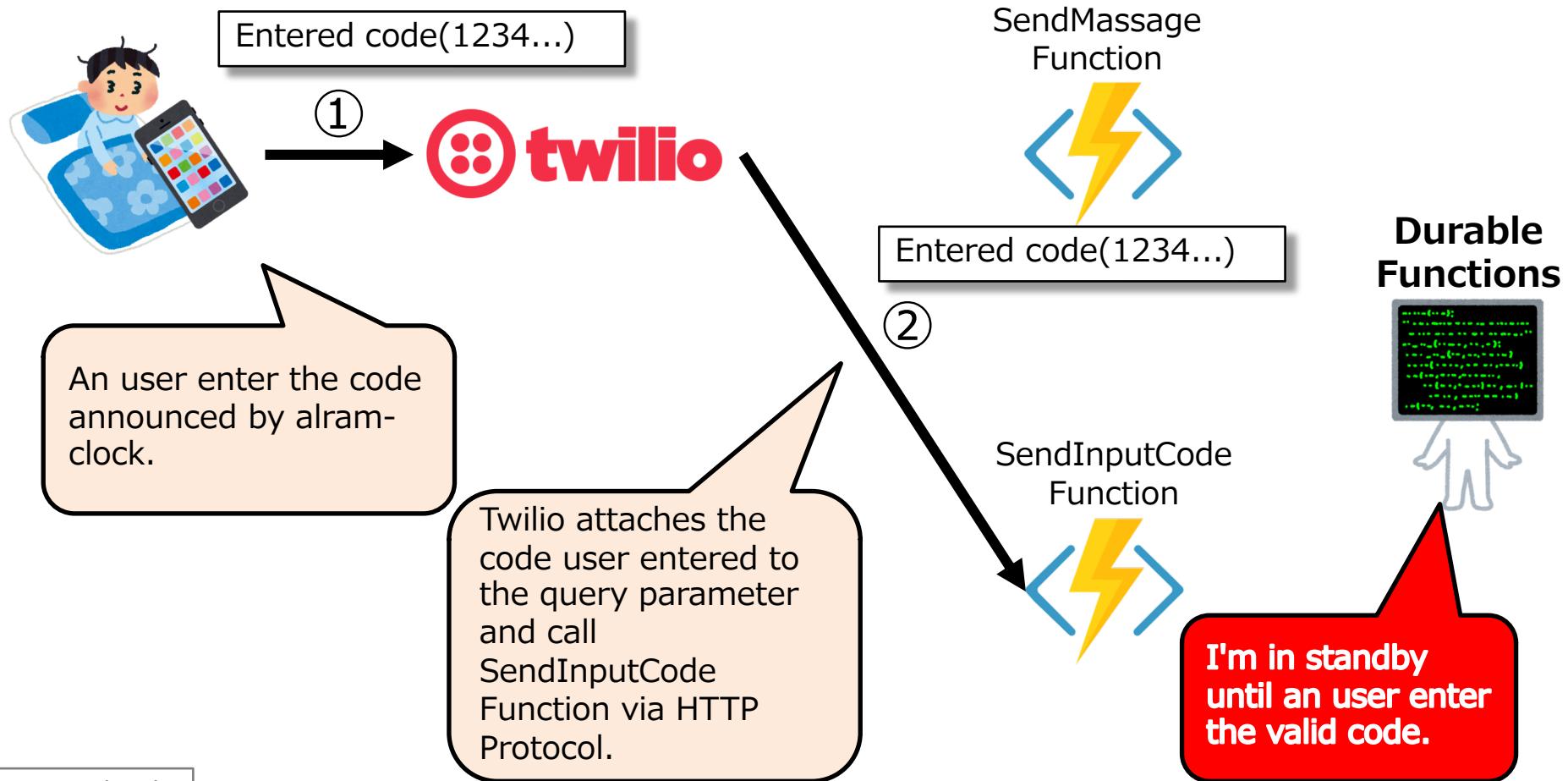
## Sequence



# Alarm Clock by Durable Functions



## Sequence



# Alarm Clock by Durable Functions



## Sequence



SendInputCodeFunction send an external event to Durable Functions with code an user entered.

SendMassage  
Function



I received an  
external event, so  
I'm in active again.

Durable  
Functions



SendInputCode  
Function



Entered code(1234...)

External  
Event

# Alarm Clock by Durable Functions



## Sequence



Durable Functions check the code is valid. If invalid, Durable Functions call SendMassage Function again, and order an user to enter the valid 10 digit code by making a phone call to an user.

SendMassage  
Function



Durable  
Functions

SendInputCode  
Function



# Serverless Architecture



You can see the details of this content at the blog as shown below

“SIOS TECH.LAB”

<https://tech-lab.sios.jp/>

The screenshot shows the homepage of the SIOS TECH.LAB website. At the top, there is a navigation bar with links for '技術ブログ', 'GUEST.CH', 'RED HAT.CH', 'NGINX.CH', 'OSSよろず相談室CH', 'Tech Lab' (with a GitHub icon), and 'ブラウザで読む'. Below the navigation, a banner displays the date '[2018-07-12] Tomcat がハングした' and the category '技術ブログ'. A featured image shows a heart made of colorful LEGO bricks. The main content area has two visible posts:

- Raspberry Piで一緒に「Lチカ」しまよう (中編)**  
Published on 2018-07-24 with 0 comments. The post discusses a simple electronic circuit project using a Raspberry Pi.
- オープンソースカンファレンス 2018 Hokkaido 参加レポート**  
Published on 2018-07-13 with 0 comments. This is a report from the Open Source Conference 2018 Hokkaido.

On the right side of the page, there is a sidebar with a '新着記事' section and a '人気記事ランキング' section. The '人気記事ランキング' section includes a thumbnail for an article about 'Zabbixの監視情報をGrafanaでグラフィカルなダッシュボード'.

## The related posts

### ■ Serverless Architecture

<https://tech-lab.sios.jp/archives/10856>

### ■ OpenID Connect

<https://tech-lab.sios.jp/archives/8651>

### ■ JSON Web Token

<https://tech-lab.sios.jp/archives/7576>

### ■ API Gateway

<https://tech-lab.sios.jp/archives/10606>

### ■ Single Page Application

<https://tech-lab.sios.jp/archives/7553>

Thank you  
for your time  
and  
attention.