

FUJITSU Cloud Service K5 CF Service Functional Overview

April 2018
Fujitsu Limited

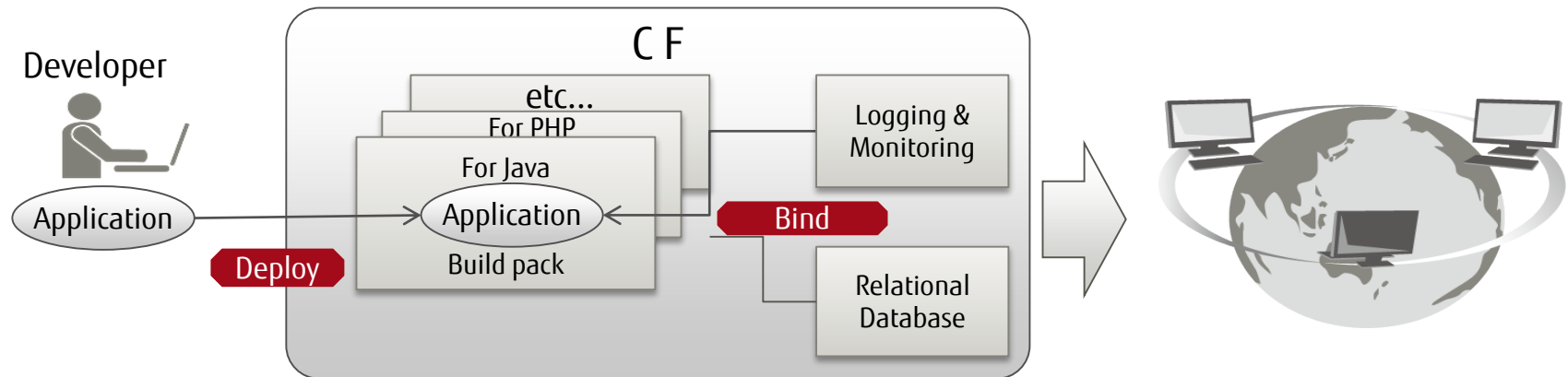
- Unauthorized copying and replication of the contents of this document is prohibited.
- The contents of this document may be changed without prior notice.

- About CF
- Features of CF
- Services Provided
- Overall View of Services Provided
- Function Overview
- Function Details
 - Application Execution Environment
 - Service Link
 - Scaling/ Load Balancing
 - Health Monitoring/ Auto-recovery
 - Operation View/ Development Tools
 - Build Packs
 - Services that Can be Combined
- Billing Model
- Restrictions and Notes

What is CF?

CF is an **application execution environment service** based on “Cloud Foundry” (open source). CF provides **an environment that helps its clients to rapidly release to market systems that produce new value.**

- Build packs help to **quickly deploy and publish applications.**
- The combination of various services enables **prompt and flexible development.**
- The CF dashboard helps to **control operations management.**



(1) Develop an application using the language of choice

(2) Deploy the application

(3) Publish the application on the internet

Execution of applications coded in various languages

CF users can select the most suitable development language, such as Java, Node.js, PHP, etc. They can deploy and publish applications developed using these languages.

Development and operation with service mashup

CF users can rapidly and flexibly implement new applications based on frequently changing ICT requirements by combining the prepared application development environment and various services.

Operation support via a dashboard

CF users can visualize the status of resources, and search and refer to application logs via a dashboard. CF supports stable operation by visually displaying operation and incident information.

Scale out / in based on operation status

CF users can start small when beginning to use the CF service because it is easy to scale applications out/ in as well as up / down.

Use of open technology

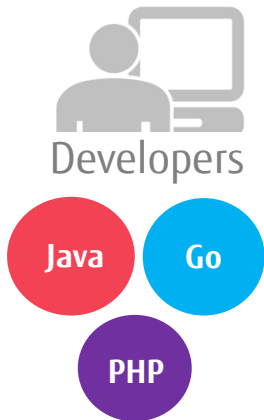
CF users can reduce vendor lock-in risk as the CF platform uses open source services which include contributions from many developers around the world.

- CF users can deploy and release applications developed using Java, Node.js, PHP, etc. **The development language most suitable to the system features can be selected.**

■ Use of a build pack

CF users can run several types of applications because they can select the appropriate build pack for the development language.

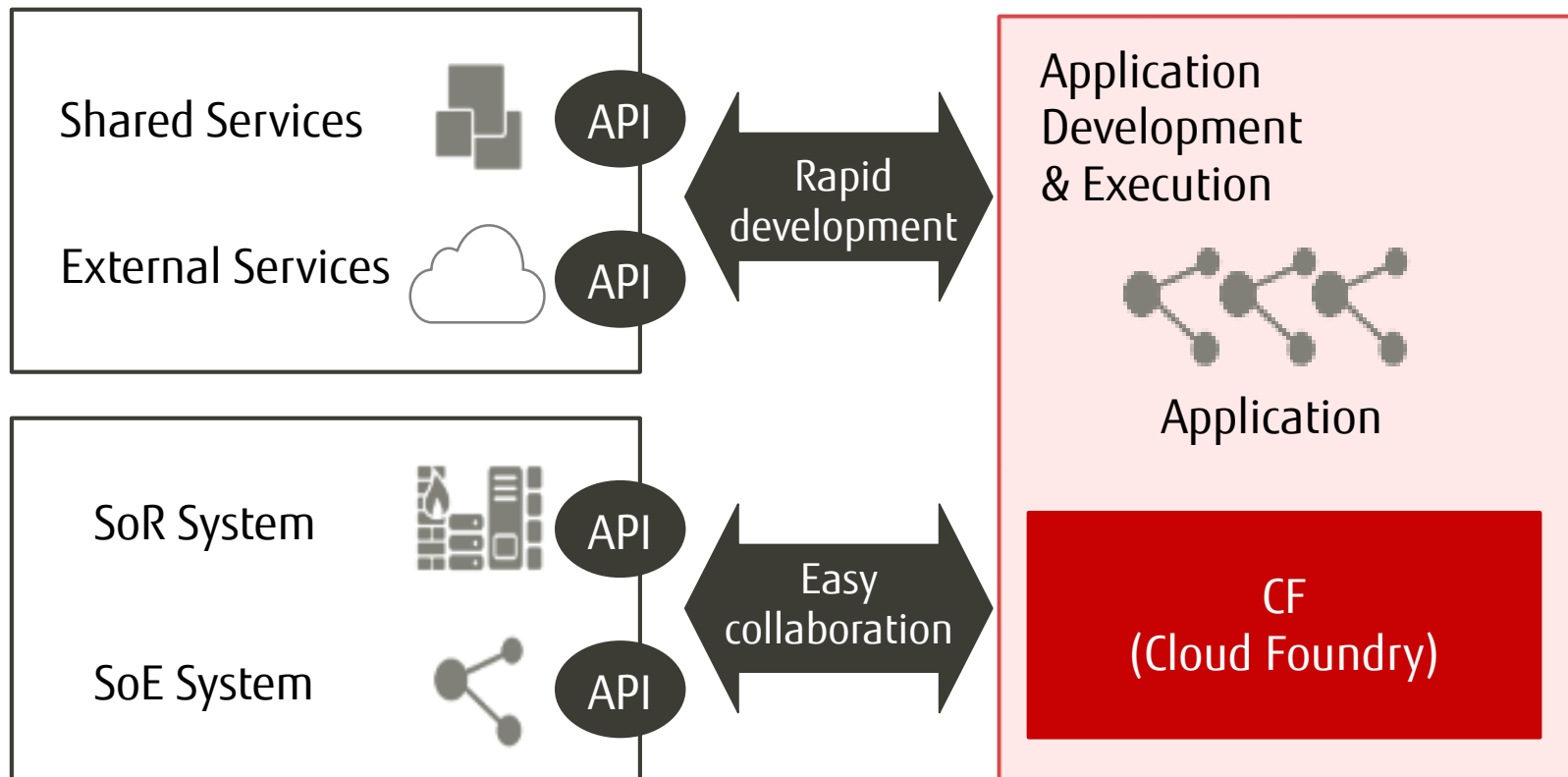
* A build pack can deploy the necessary framework and runtime environment for execution of the application.



The screenshot displays the 'FUJITSU Cloud Service K5 PaaS' interface. The top navigation bar includes 'HOME', 'Catalog', 'Service In Use', and 'Usage Fee'. The 'Catalog' section is active, showing a grid of build packs for the 'CF' service. Each build pack card includes a language logo, the language name, and a brief description of the service.

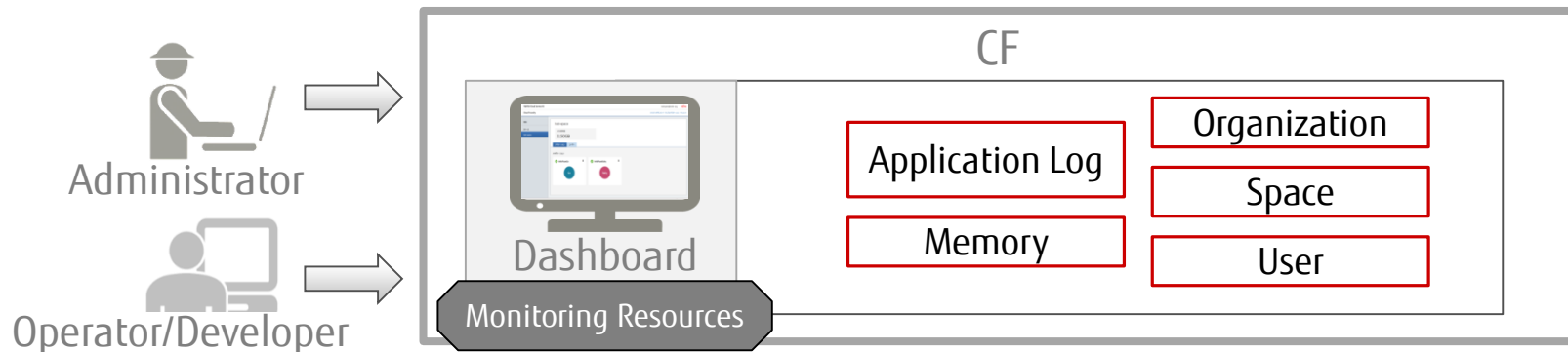
Language	Build Pack Description
Java	This service provides the buildpack for the execution environment of Java Web applications based on GlassFish 4.1 Web Profile/Java SE 8.
Go	This service provides the buildpack for the execution environment of the applications written in Go.
Node.js	This service provides the buildpack for the execution environment of the applications written in Node.js.
PHP	This service provides the buildpack for the execution environment of the applications written in PHP.
Python	This service provides the buildpack for the execution environment of the applications written in Python.
Ruby	This service provides the buildpack for the execution environment of the applications written in Ruby or using frameworks of Rack, Rails, o...
Staticfile	This service provides the buildpack for the execution environment of the Static HTML, JS, and CSS applications.

- CF users can **rapidly and flexibly implement an application even when ICT requirements are changing frequently** as they can combine the application development environment with various prepared services.



Operation Support via a Dashboard

- **Application status and user privileges can be managed** through a dashboard. The **status of resources**, such as memory used by applications, can also be monitored.
- Administration of applications
Starting/ stopping of each application and a list of application statuses (including activation)
- Monitoring of the organization and space, and administration of user privileges
Display of user access privileges and quotas with regards to organization and space.
(*) Organization and space is a unit of user administration. CF provides one organization per a contract number in each region. It controls application instances and service instances. A CF user (one organization) can generate multiple spaces.
- Visualization of resource status
CF visually displays the resources used by applications.
- Searching/ referencing of application logs
CF users can search and refer to application logs via the dashboard and analyze incidents.



Scale Out / In Based on Operation Status

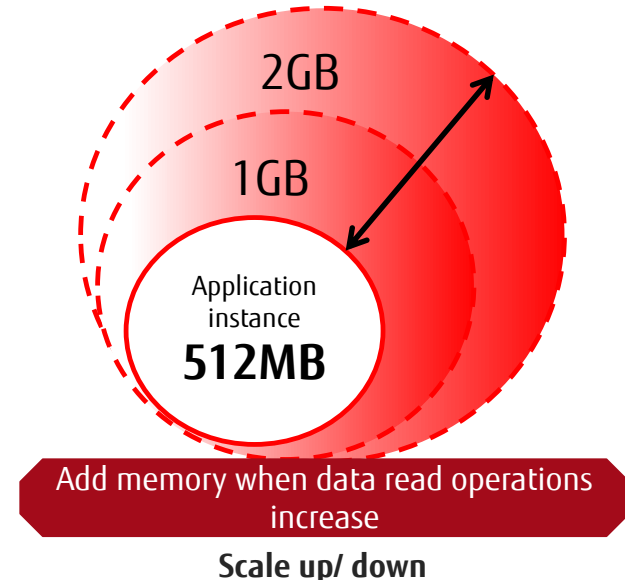
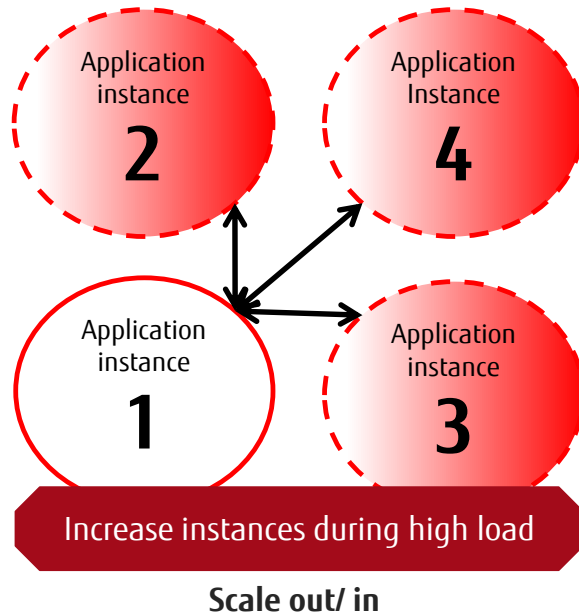
- CF users **can start with minimum resources** as applications can be easily scaled out / in and up / down.

- Scale out / in

CF users can manually increase/ decrease the number of instances of an application. CF can automatically dispatch the requests to each application in each instance even when the number of instances changes.

- Scale up / down

CF users can increase/ decrease the memory size available for an application.

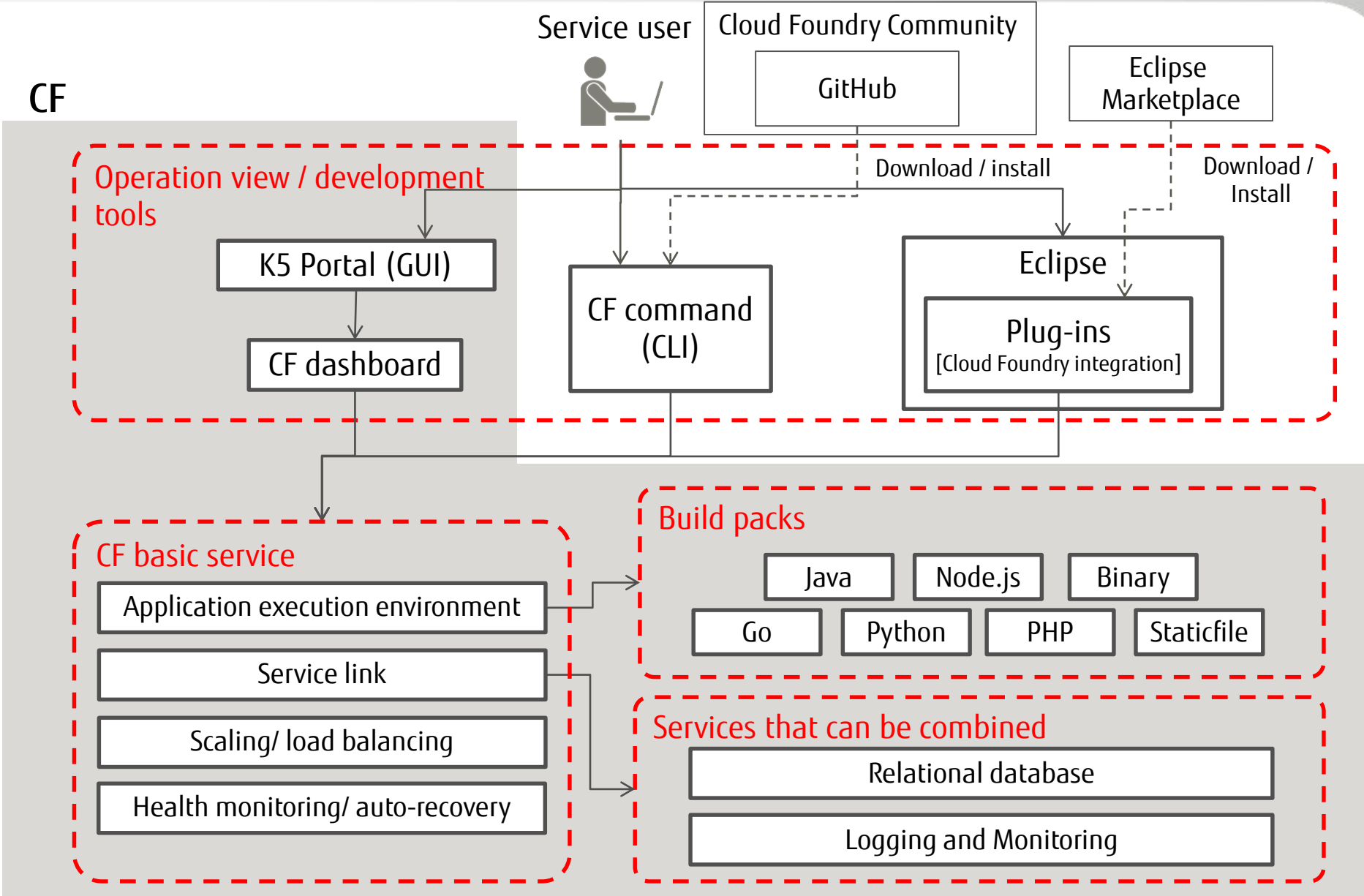


- CF users **can reduce vendor lock-in risk** as CF provides its users with the **latest PaaS** based on open sources where input includes contributions from many developers around the world.
- PaaS Software: “Cloud Foundry”
As “Cloud Foundry” is open source, it provides application portability without silos between cloud vendors, public/ private clouds, etc.
Many vendors participate in the Cloud Foundry Community and develop and provide the latest PaaS which is enhanced by the technology of each vendor.
Supporting companies: Fujitsu, IBM, NTT, Pivotal, HP, VMware, SAP, etc.

CF provides the following services:

- CF basic service
- Operation view/ development tools
- Build packs
- Services that can be combined

CF Service Map



■ CF Basic Services

Function	Description
Application Execution Environment	CF users can execute applications developed with various languages such as Java, Node.js, etc.
Collaboration with Other Services	CF users can call other services from their applications by selecting and registering the services using commands as CF generates service instances.
Scaling/ Load Balancing	CF users can scale up/ down instances of an application. Access requests to applications can be automatically allocated to each instance even after the number of instances increases/ decreases.
Monitoring of Alive/ Dead, Auto Recovery	CF monitors the status of applications. When an application instance ends abnormally, CF automatically recovers the instance.

■ Operational View/ Development Tools

Function	Description
CF Dashboard	CF provides users with application management functions via a dashboard.
CF Command	This is a command program that allows CF users to download from the CF community and install onto client devices to manipulate CF via commands.
Eclipse Plugin	This enables an operational view of Eclipse and is used to deploy applications created in the Eclipse development environment to CF.

■ Build packs

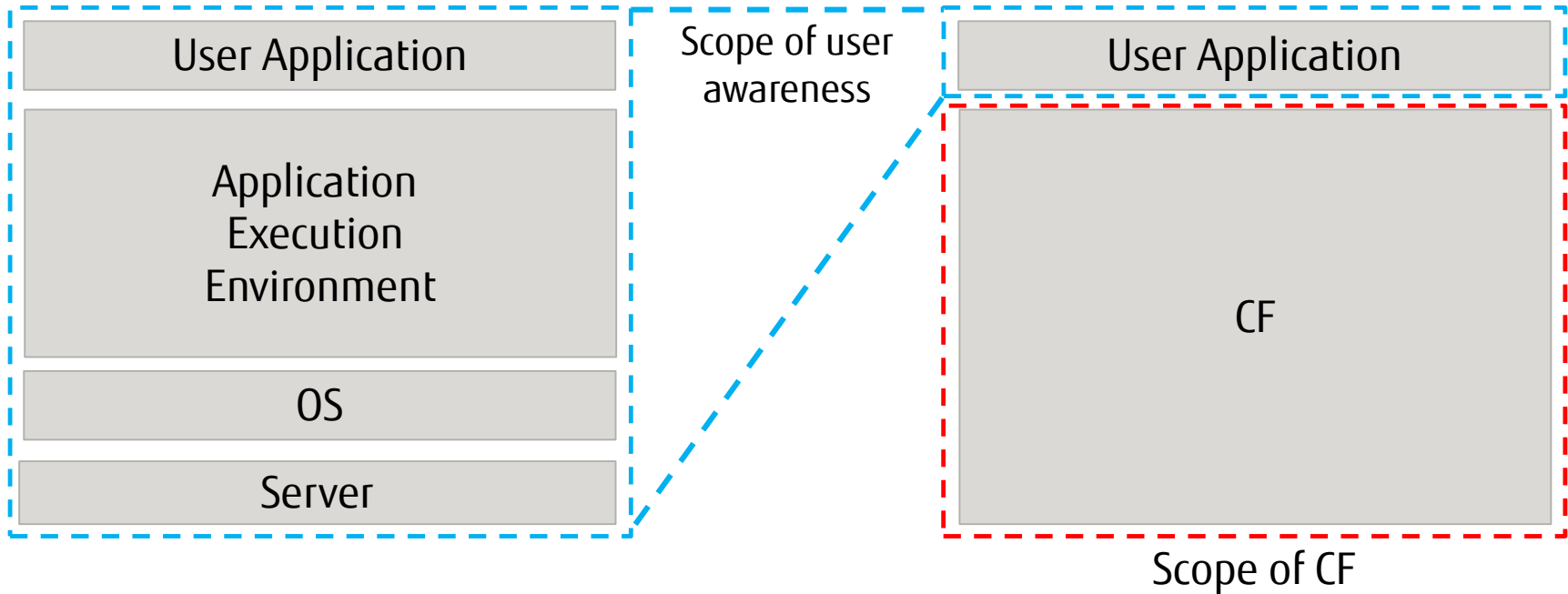
Function	Description
Java	Java Web application execution environment based on GlassFish 4.1Web Profile/ Java SE 8
Java	Java Web application execution environment based on Tomcat/ OpenJDK
Go	Application execution environment for Go.
Node.js	Application execution environment for Node.js
PHP	Application execution environment for PHP
Python	Application execution environment for Python
Ruby	Application execution environment based on Rack, Rails, Sinatra or an application coded in the Ruby language.
Static file	Execution environment for static HTML, JavaScript and CSS applications.

■ Services that can be combined

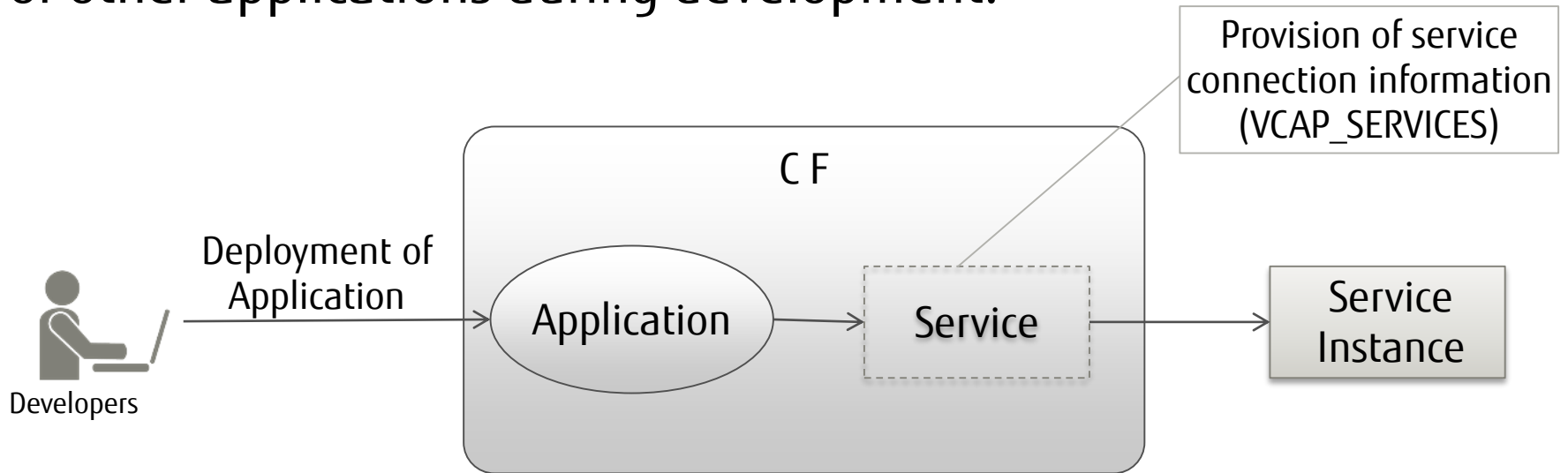
Function	Description
Relational Database	This provide CF users with the option of an "open standard" PostgreSQL database service.
Logging and Monitoring	CF users can search, delete and reference logs output by applications. CF users can display logs on the CF dashboard.

Function Details

- CF provides users with an execution environment suitable for their applications. CF users do not need to be aware of details relating to the server, the OS or the application execution environment.



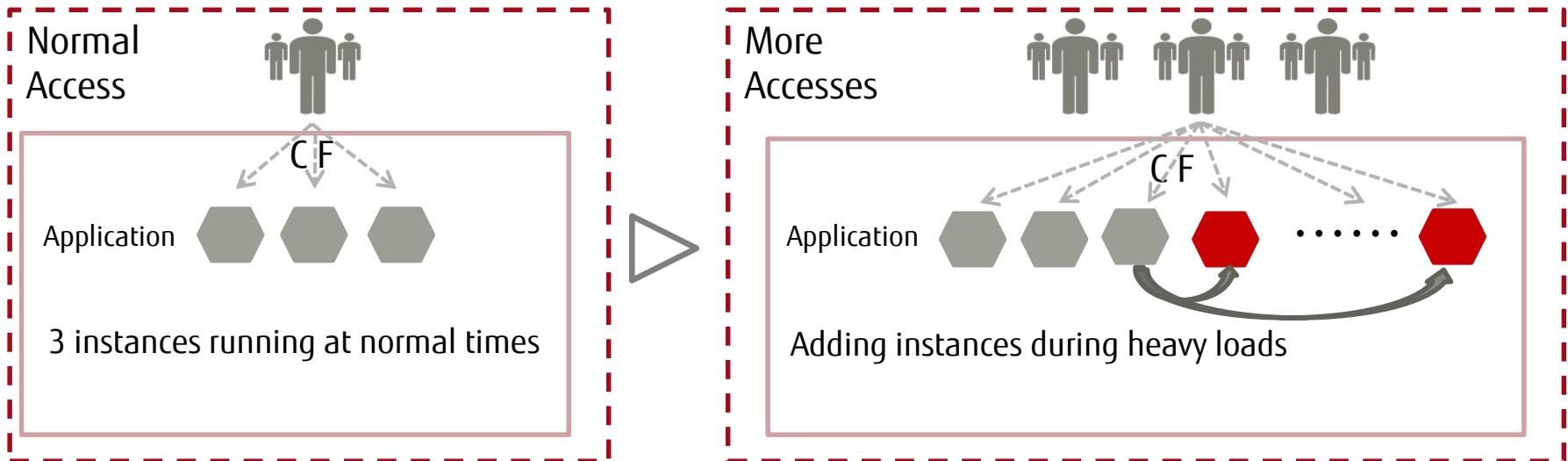
- Applications on CF can be combined easily with other services.
- Other services are coupled loosely with CF applications. Therefore, applications can be developed without recognizing the existence of other applications during development.



Information required to connect with other services are defined by environment variables called `VCAP_SERVICES`. CF applications can connect with other services using `VCAP_SERVICES`.

■ Scaling

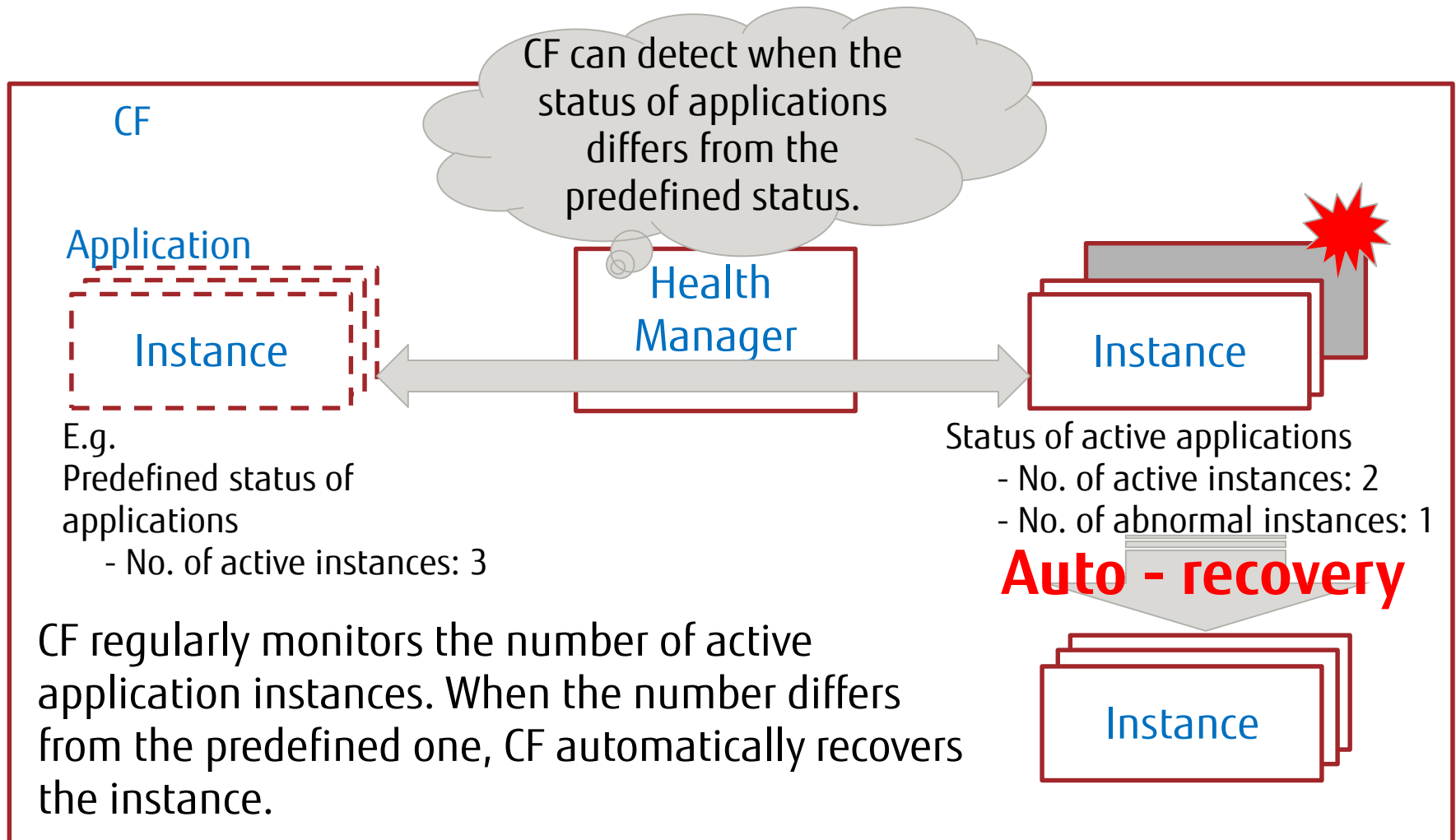
CF users can increase or decrease the number of instances based on service workload.



■ Load Balancing

CF's Load Balancing function allocates access requests using the round robin method.

- CF constantly monitors the status of applications. When applications end abnormally, CF recovers them automatically.



- CF users can use the following tools to modify the configuration of CF (organization, space, etc.), to deploy applications, to activate or suspend applications, and so on.

- CF Dashboard

CF users can manipulate and enjoy CF's basic services via a browser using a GUI. In addition, CF users can monitor infrastructure resources (CPU, Memory, Disk, etc.) that are used by applications.

- CF Command (*)

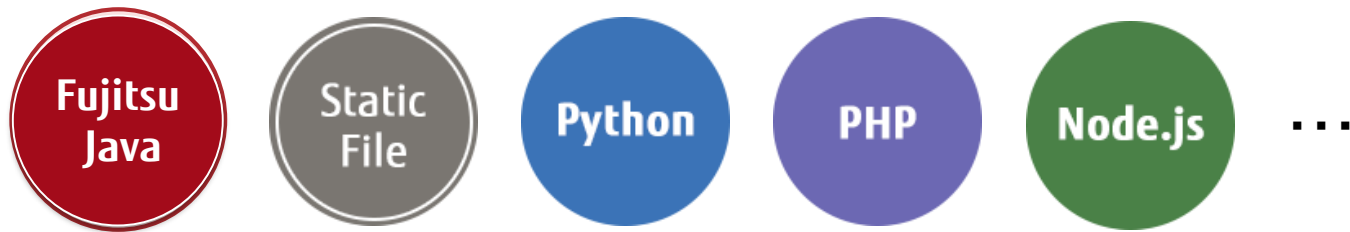
CF users can manipulate and enjoy CF's basic services using CLI via the client device where the CF command is installed. CF users can automate some manipulation tasks by creating scripts.

- Eclipse Plugin (*)

CF users can integrate application development and CF manipulation using the Eclipse GUI, Integrated development environment.

(*) Please download and install separately the CF command and the Eclipse plugin from the public sites.

Multiple web application development languages such as Java, Node.js, and PHP are supported as standard.



- Build Packs can provide the necessary runtime environment and framework for the execution of applications on CF.
- CF users can use various Build Packs embedded in CF.

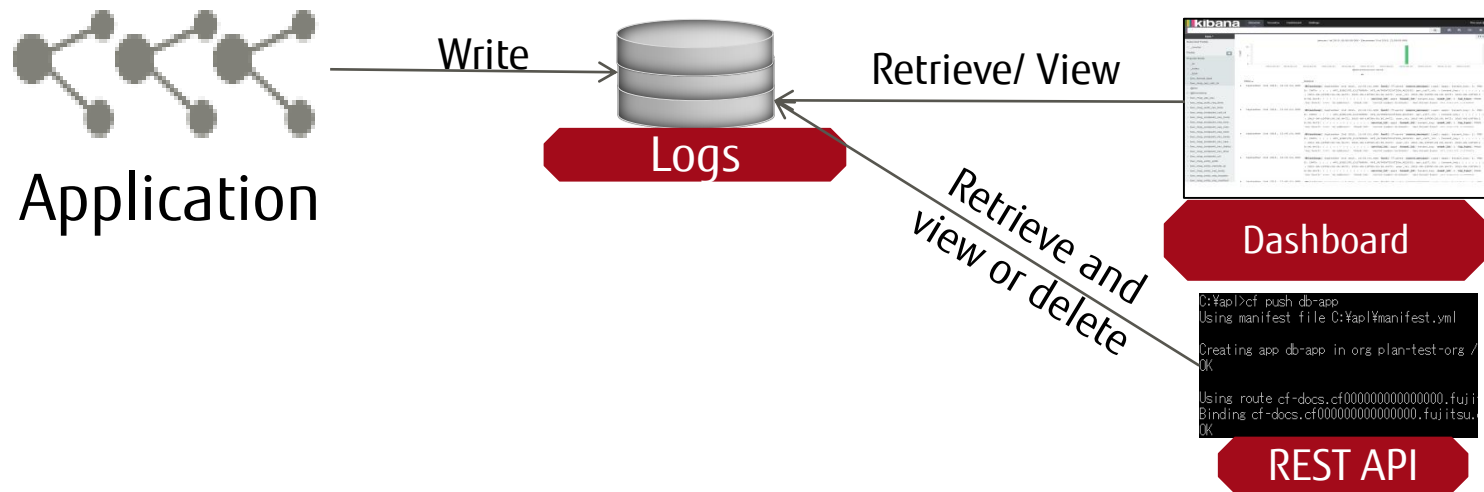
■ Relational Database

Open standard database: PostgreSQL

- Automatic configuration of a PostgreSQL service
When a specific PostgreSQL service is connected to a Java web application, the PostgreSQL JDBC connection is automatically configured during activation of the application.

■ Logging and Monitoring

- Logs output by applications can be retained, and then retrieved and viewed or deleted.
 - Using REST API, logs can be retrieved and viewed or deleted.
 - Logs can be retrieved and viewed from the dashboard.
 - Logs can be retained for up to 99 days.



- Logs can also be viewed using the `cf logs` command. Reference: Tutorial “Debugging applications” <https://cf-docs.jp-east-1.paas.cloud.global.fujitsu.com/en/manual/tut/tut/topics/c-dev-debug.html>
- No more logs can be retained after the log storage capacity is reached. Ensure logs are deleted at regular intervals or adjust the log output frequency and the retention period.
- When an error is generated in the logging and Monitoring service, viewing retained logs may become disabled. Ensure that required logs are retrieved and back them up at regular intervals.

Services that Can be Combined (3)

■ Custom meters

- Custom meters can be created and samples* registered and displayed.
- Each sample registered for a custom meter is retained for 14 days.
- A custom meter will be automatically deleted when the custom meter has no samples registered for 14 days (no samples registered for the same custom meter name).

* A sample refers to any data that is specified and obtained by the user through a custom meter.

Reference: Log Service Provision Plan

Plan	L1 (Light)	M1 (Medium)
Log storage size (GB)	10	50
Log retention period	99 days	
Custom meter retention period	14 days	
Supported speed (estimation)	Approx. 50 logs/second per application	
	Approx. 50 logs/second per service	Approx. 100 logs/second per service
Data storage	1 location	2 locations (by replication)
Operation from the CF dashboard	-	<ul style="list-style-type: none">• Search/view logs• Check the log storage size• Check custom meter samples

Billing Model

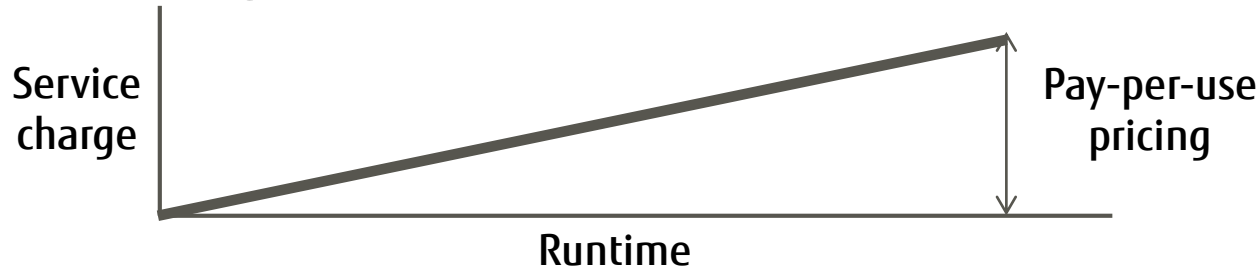
Billing Types

Category	Function	Pricing	Comment
Build pack	Various build packs	Pay-per-use (amount of memory allocated x runtime)	Billing is based on runtime and the amount of memory allocated when the application is running.
Service combination	Relational database	Pay-per-use (runtime)	Billing is based on the combined capacity of CPU, memory and storage, and usage time.
	Logging and Monitoring	Pay-per-use (runtime) + Pay-per-use (no. of custom meters)	Billing is based on the combined storage capacity of the CPU, memory and logs, usage time, and the number of custom meters.

Category	Service name	Plan	Comment
Services combination	Relational database	Small	1vCPU / 4G memory / 20GB storage
	Logging and Monitoring	L1 (light)	Log storage size 10GB
		M1 (medium)	Log storage size 50GB

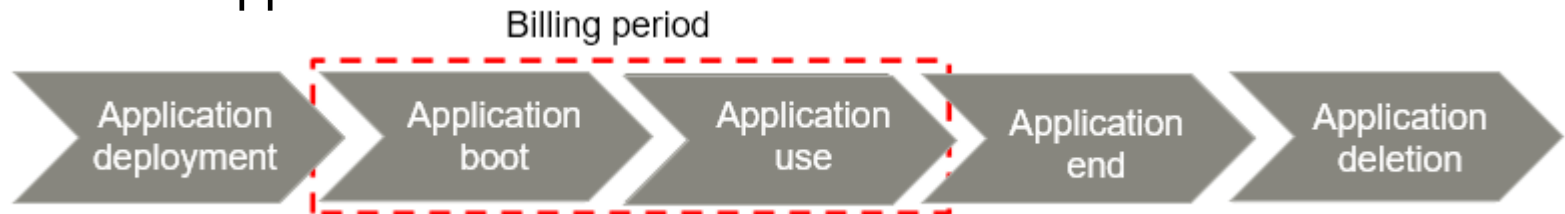
- When multiple functions are being used under one contract, the billing is calculated for each function in use and invoiced together.

- Pay-per-use billing, based on the amount of memory allocated x runtime. (Gigabytes per unit of time)



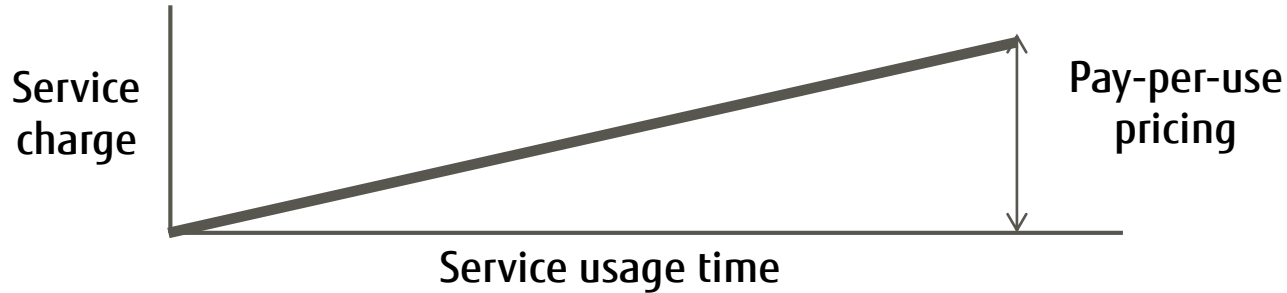
- The 'amount of memory allocated' noted above refers to the total number of application instances.
 - The 'gigabytes per unit of time' in the above calculation is the total combined value for the organization overall.
- Billing period

Billing begins from the moment an application starts up. Billing stops the moment an application shuts down.



Note: The 'amount of memory allocated' is the amount of storage specified by the user when deploying an application. Amount of memory allocated is not equal to Amount of memory used.

- Pay-per-use billing, based on plan charge x service usage time



- One day of usage time is calculated in units of hours (rounded up) per generated service.

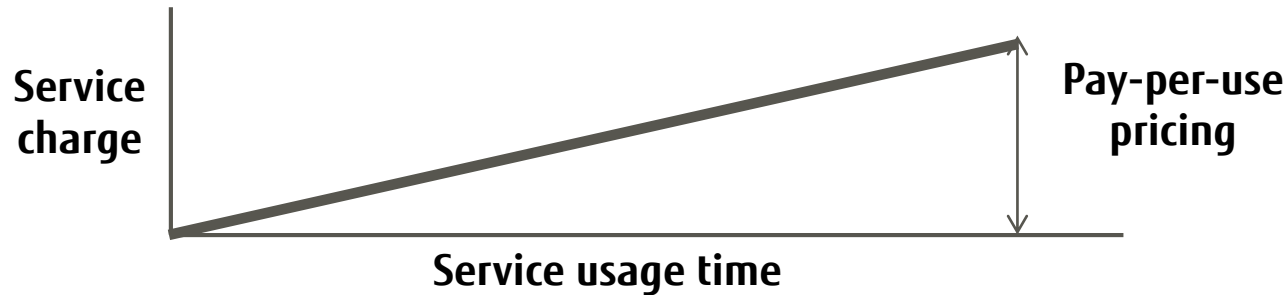
Example: 1 hour 45 minutes is rounded up to 2 hours

- Charging begins the moment a service is generated. Charging continues until the service is deleted.



Logging and Monitoring Service (1)

- Total of log usage charges and custom meters.
- Log usage charge
Pay-per-use billing, based on plan charge x service usage time.

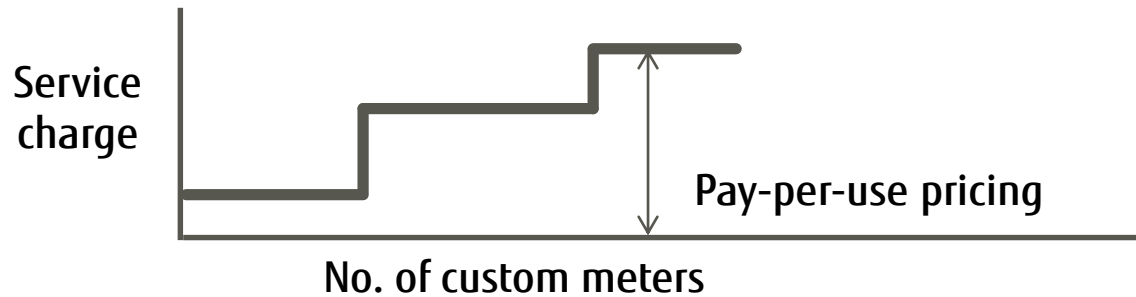


- Usage time is calculated using the daily usage time for each service generated (unit of time is shown to 4 decimal places)
- Billing starts from the moment a service is generated. After service generation, billing continues until the service is deleted.



■ Custom meter charge

Pay-per-use pricing, based on custom meter unit price x number of custom meters.

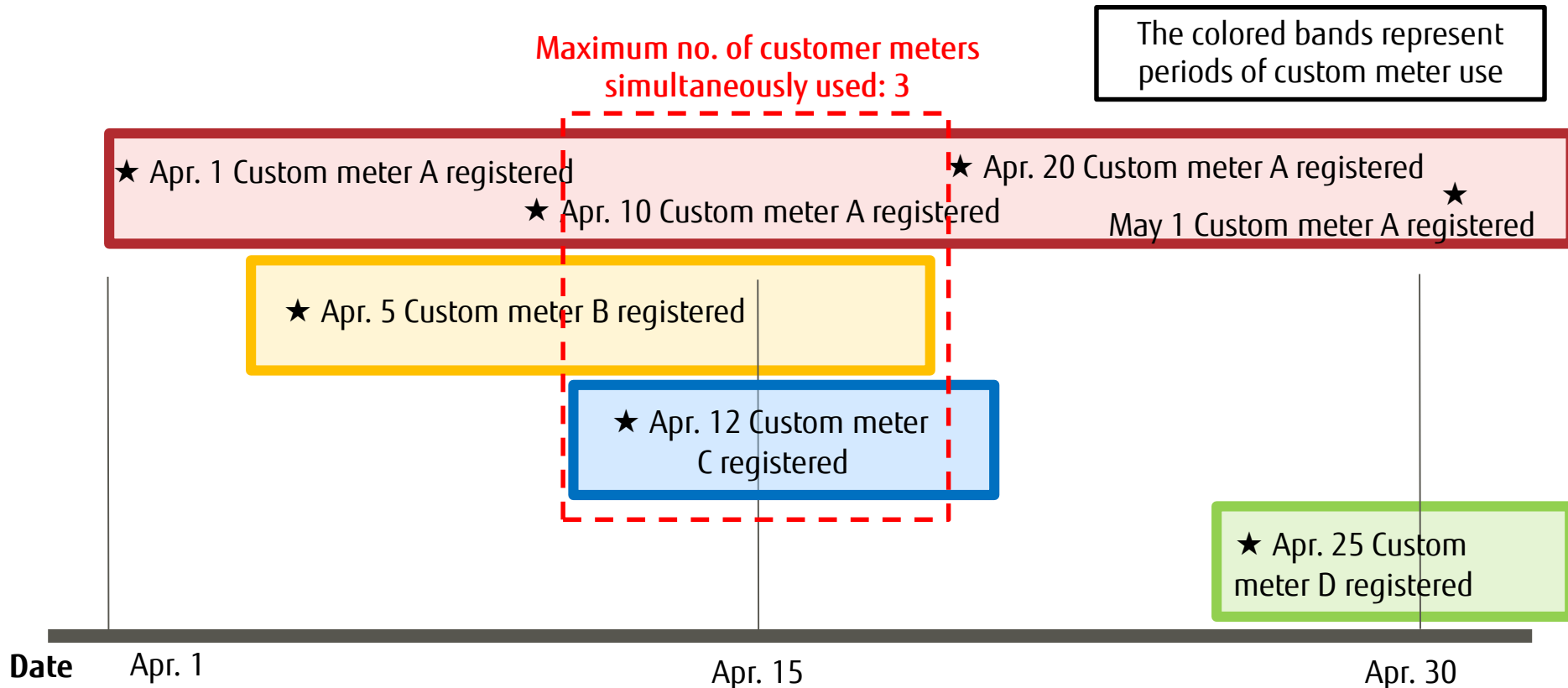


- The number of custom meters is calculated by grouping the meters by name and counting the maximum number of custom meters used simultaneously.
- When a new custom meter is saved, it is added to the number of custom meters and additional charges will be incurred.


Example of Billing for Custom Meters

- Each month, the user will be billed for the maximum number of custom meters (monitoring items) that are registered and used simultaneously on the following API.
POST /v1/monitor/meters/{meter_name}
- If custom meters of the same name are registered multiple times in the same month, those meters will be counted as 1 for billing purposes.

Example: In the following case, the number of custom meters billed (maximum number for this month) will be 3.



- Refer to the Service Description on FUJITSU Cloud Service K5 Website to confirm the regions in which this service is offered.



FUJITSU

shaping tomorrow with you