Note: As part of a service revision, Fujitsu will not newly accept the Customer's Service Application for the PF Service after 00:00 (UTC) on 11th January, 2018 (Thursday). The timeframe for restarting the acceptance of new service applications will be notified separately.
*The contents of this document are subject to change after the major service revision.

FUJITSU
shaping tomorrow with you

FUJITSU Cloud Service K5
PF Service Functional Overview

January 2018
Fujitsu Limited

Contents

FUJITSU

What is PF?

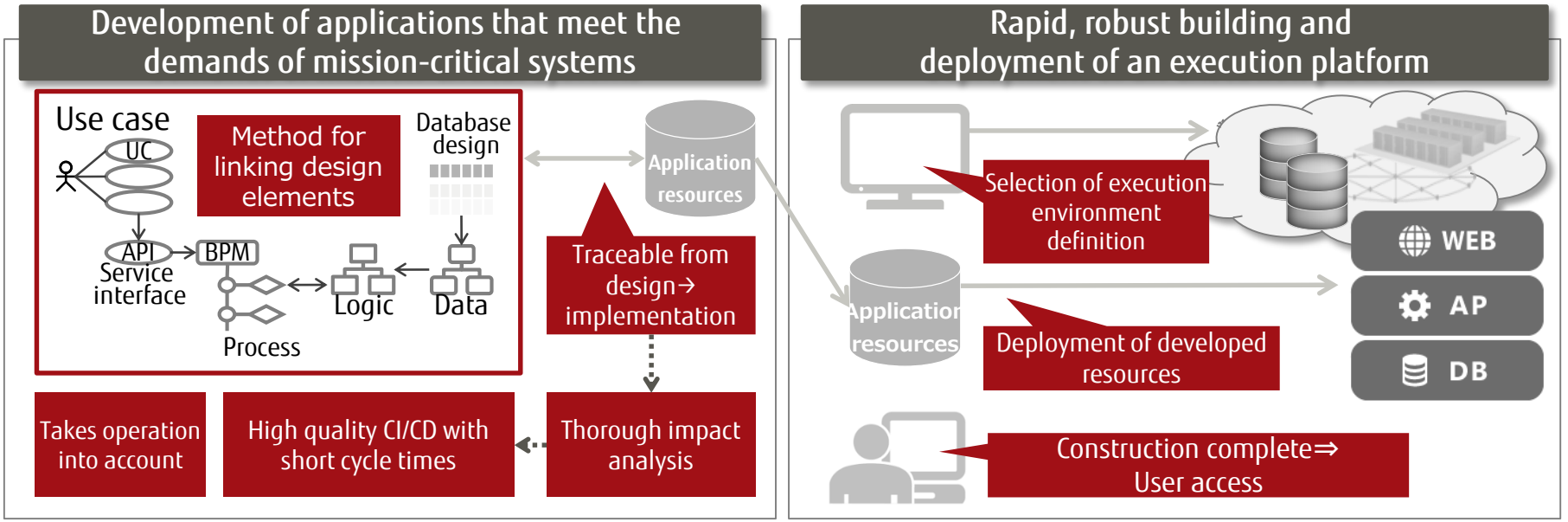
The PF Service allows legacy assets to be modernized, realizing a robust system responsive to changing business requirements.

- **Supports the development of applications that meet the demands of mission-critical business**

  Drastically reduces development timelines, improves maintainability and operability, and shrinks development scale.

- **Supports rapid, robust construction and the deployment of execution platforms**

  Systems can be easily built be simply selecting the execution environment definition and deploying the resources you have developed.



Development of Applications that Meet the Demands of Mission-critical Business

FUJITSU

The development methods and rules provided by PF enable development of applications with a high level of maintainability and operability.

Features of application development using PF

Standardization of Application Architecture

Standardizing the type of application architecture and managing the performance architecture at design time **realizes an application architecture without any gap between design and implementation.**

Modeling Individual Business Functions as Services

Manages design and implementation using the smallest unit of a business function, which is a service. Using models and rules to implement service functions, and defining and managing the linkage flow between services **improves service portability.**

Design Quality Assurance

Maintaining the relationship between changing design information **ensures design quality.**

Business Rule Definition

Using design tools to define the business rules that establish the principles that inform business decisions, and using a rules engine to evaluate these rules simplifies the **normalization of conditions and results.**

Implementation of Thorough Impact Analysis

Using design tools and a code checker for Java resources to strictly regulate standardization constraints enables **thorough impact analysis** at resource modification time.

Simple Implementation of Recovery, Fault Analysis, and External Linkage

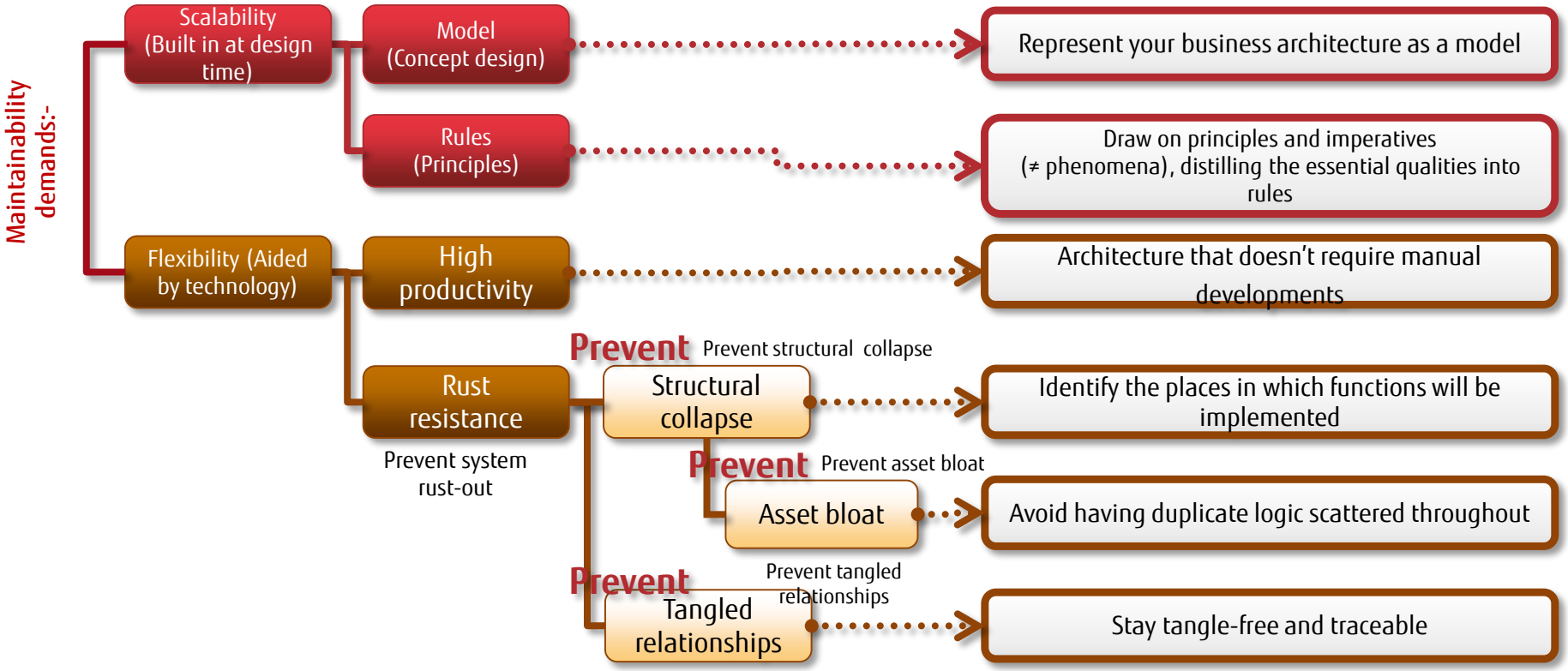
Facilitates implementation of data assurance, performance bottleneck analysis, data store linkage, and external service linkage.

Reference: Conceptual Structure for Ensuring Maintainability

FUJITSU

- A highly maintainable system is a change-tolerant system, and for this you need *scalability* and *flexibility*.

Scalability (= Built in at design time)	Predict the functions you want to add or change, and achieve this at minimal cost and in minimal time.
Flexibility (= Aided by technology)	Realize unforeseen function additions or changes at minimal cost and in minimal time.



Standardization of Application Architecture (1/2)

FUJITSU

Realizing high maintainability and high productivity

Having a standardized application architecture and robust business application architecture enables development of highly maintainable, high-quality applications. The design content can be generated without modification as definitions and execution resources, enabling application development that reduces the development load and boosts productivity.

- Application architecture can be standardized by defining models, use cases, and business rules.

Output is determined by how well the model, use case, and business rules are organized

Business list

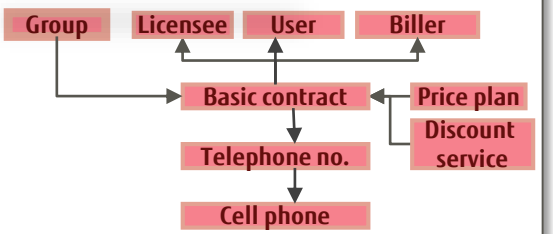
Business	Telecommunications carrier use case		Main rule
Standard contract (service start)	○	Perform new subscription	
Amendment of licensee information	○	Change licensee	
	○	Change licensee information	Name can/cannot be changed

Change of name, change of address, etc.

Extract elements

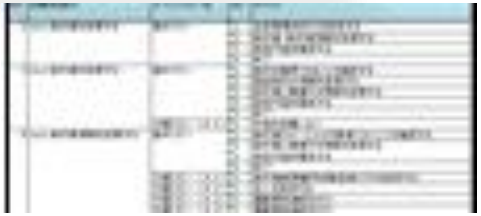
Example: Extracted design pattern

(Business) Model



Customer contract model

Use case (API)



Behavior → User / Other systems / Quasi-normal, or abnormal

(Business) Rules

1. Rule description	
Rule ID	xxx 10001
Rule name	Name can/cannot be changed
Rule type	Name
Description	Determines whether customer name can be changed
2. Input & output information	
Input information	Licensee ID, licensee surname after change, licensee name after change
Output information	Whether name can be changed: 1: Can be changed, 0: Cannot be changed

Conditions ↔ Behavior

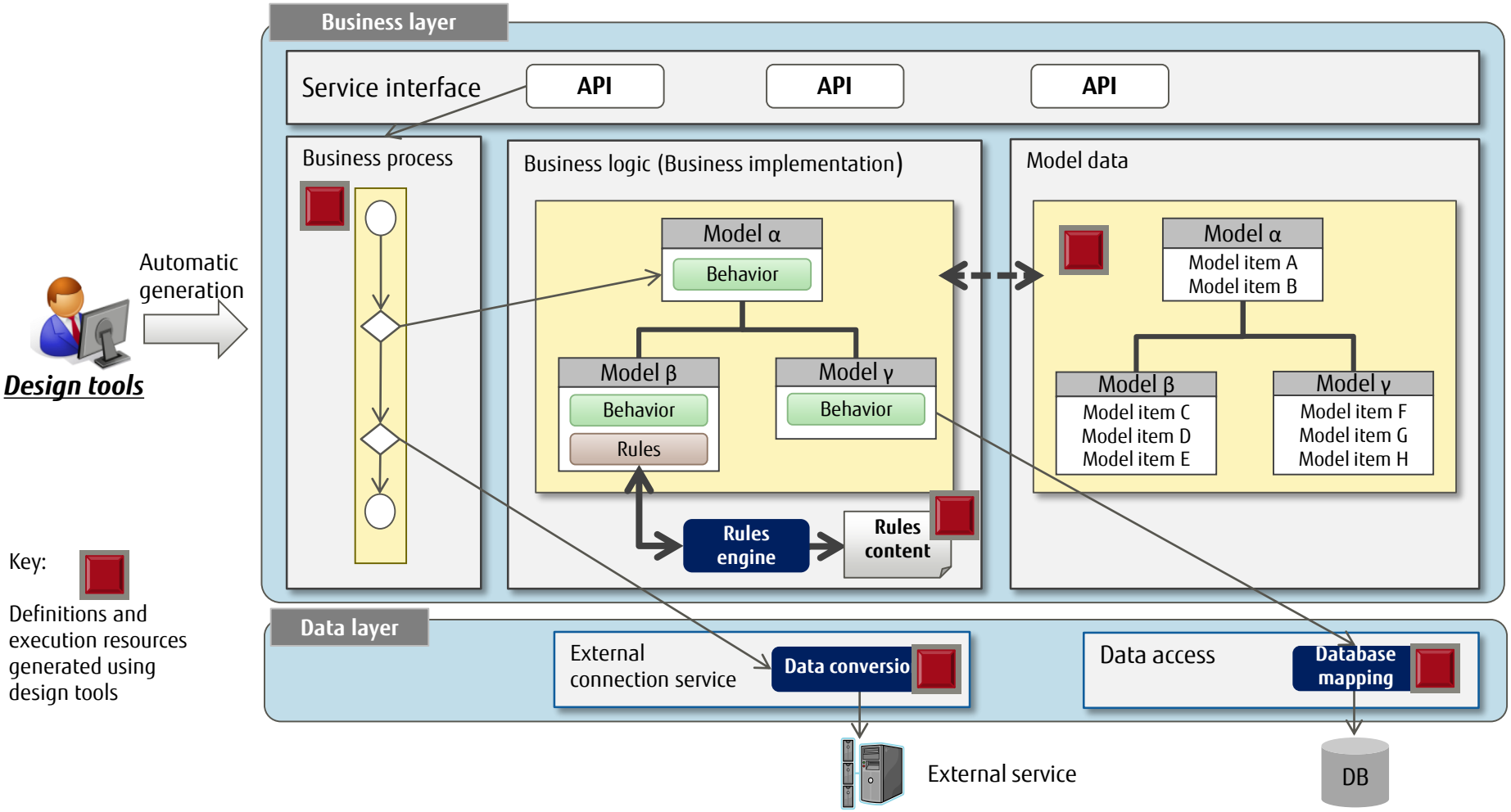
Standardization of Application Architecture (2/2)

FUJITSU

- Using design tools to automatically generate definitions and execution resources

① An application is **highly maintainable** when its architecture is standardized and there is no gap between design and construction, making it robust.

② Using design content as-is as an execution resource reduces development load and **enhances** application development **productivity**.



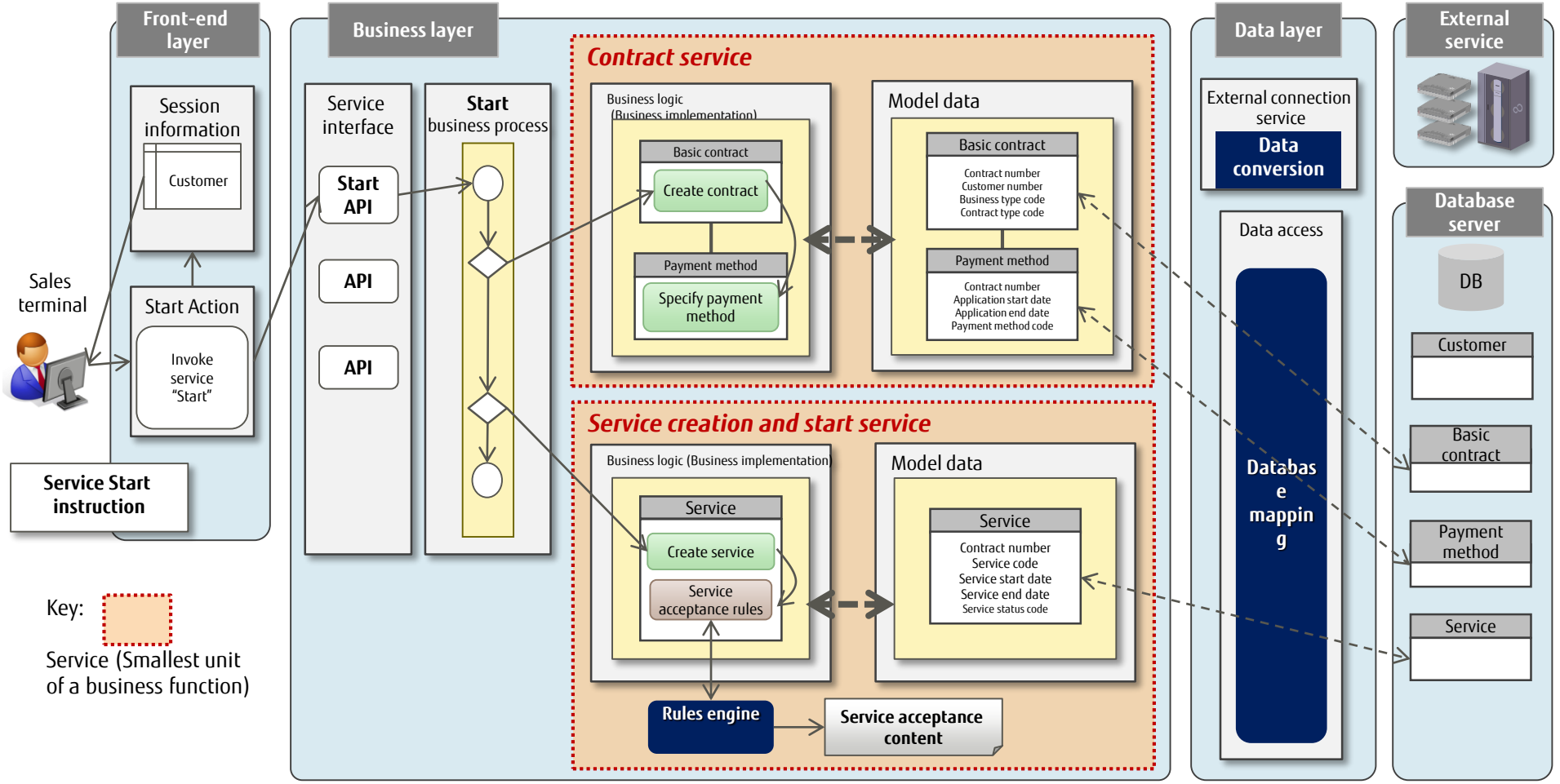
Modeling Individual Business Functions as Services

FUJITSU

Improving service portability

You can manage design and implementation using the smallest unit of a business function, which is a service. Using models and rules to implement service functions, and defining and managing the linkage flow between services improves service portability. Using business process flow control to create associations between multiple services also enables provision of services with a higher level of granularity.

Example: Using PF to model the service start task as a service



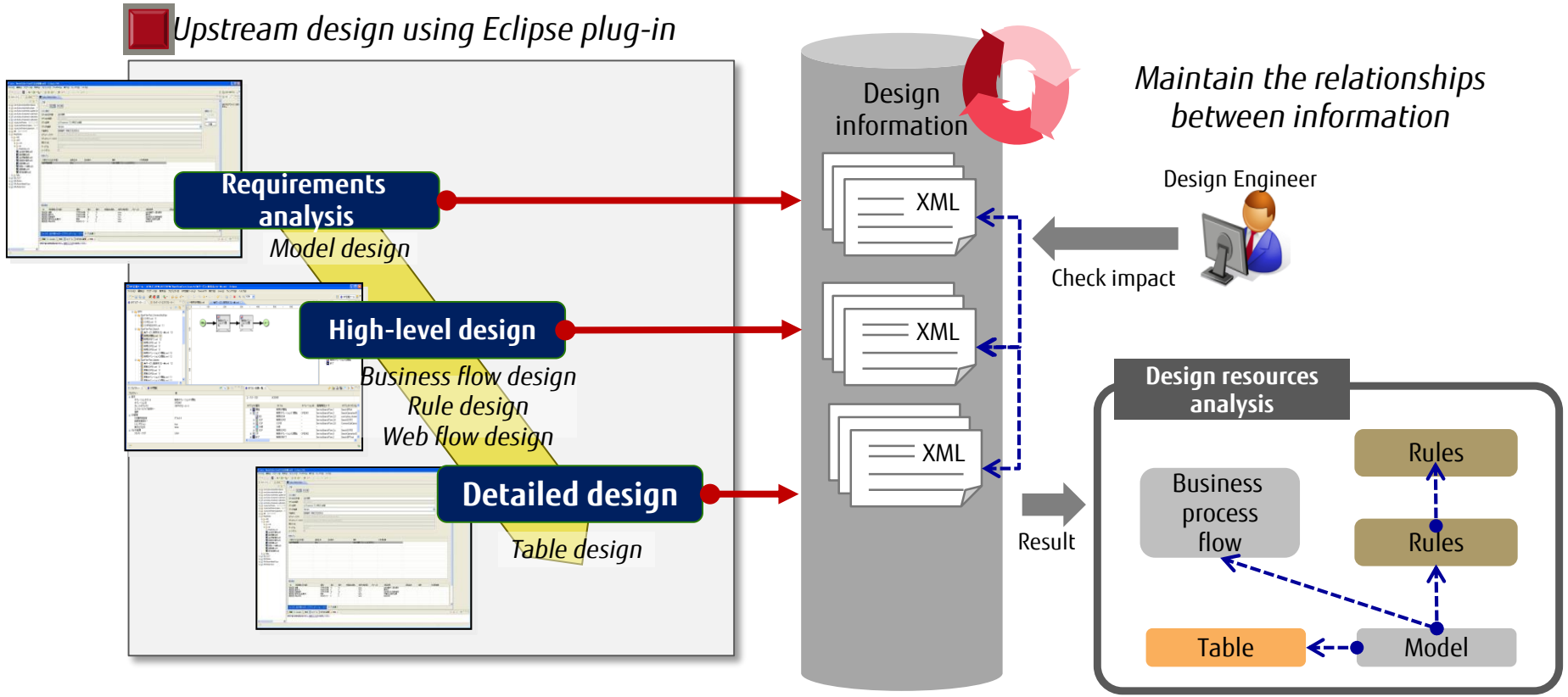
Design Quality Assurance

FUJITSU

Ensuring design quality

Ensure design quality by maintaining the relationship between changing design information.

- Provides a design plug-in to ensure the relationship with design resources.
  - ① Centralized management of design information in the design plug-in enables regular updates of the relationship between changing design information.
  - ② The relationship with affected design documents can be visualized hierarchically.

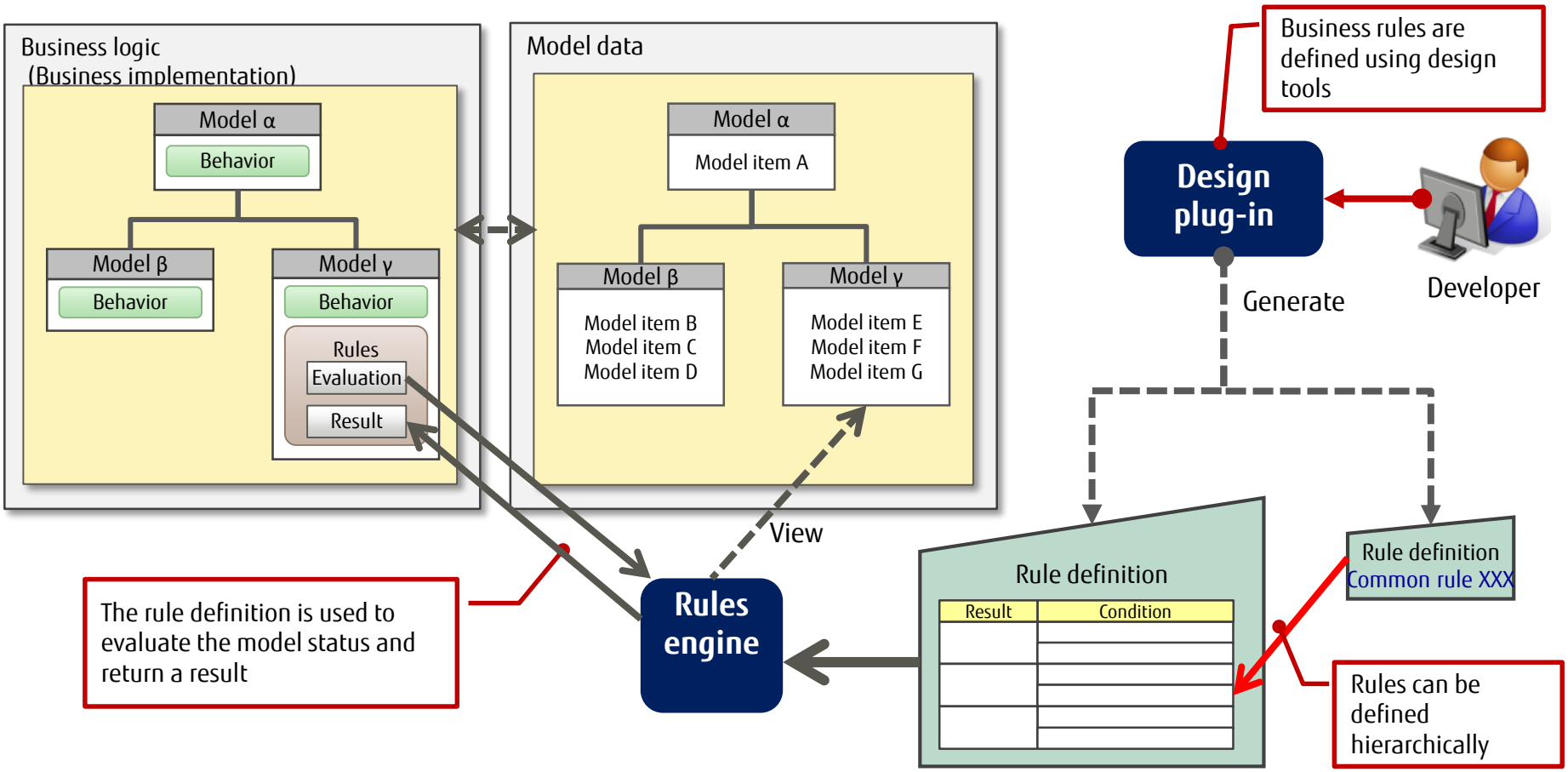


Business Rule Definition (1/2)

FUJITSU

- Facilitating normalization of conditions and results
  Using design tools to define business rules simplifies the normalization of conditions and results.

- The rules engine evaluates the data model content based on the rule definition, and returns a result.
  - ① You can describe business data item names as they are in the rules definition, making rule content clearer.
  - ② You can invoke a rules definition hierarchically from another rules definition, enabling standardization of like rules.



Business Rule Definition (2/2)

FUJITSU

■ Rule content supports an evaluation criteria table and a decision table.
It is easy to define rules content in accordance with business principles and imperatives.

Tips for creating rules content

1. Rules definition methods usually consist of an *evaluation criteria table* and a *decision table*, depending on rules content.
2. Place items with highly variable elements in a master.
3. Draw on principles and imperatives to simplify the normalization of conditions and results as much as possible.

Example: Parking lot service specs

Table of fees applied		Fee Table A	Fee Table B	Fee Table C
One parking period		Up to 1_{day}	From 2_{days} to 3_{days}	4 or more $_{days}$
Basic fee (per day)	Tax inclusive	1,200 $_{yen}$	1,000 $_{yen}$	800 $_{yen}$ (No upper limit)

② **Optional conditions**

① Use of limited express train to/from station center ⇒ Mandatory condition
② For a product spend of $30,000_{yen}$ or more ⇒ Shopping discount
③ When using a cinema within the center ⇒ Cinema discount
④ When using both ② and ③ ⇒ Combo discount

Key: Implementation policy
Rule content
Master

Rules content creation

①

Evaluation criteria table

Ranking criteria applied to parking fee	
Result	Condition
Rank A	Parking period 0-1 day
Rank B	Parking period 2-3 days
Rank C	Parking period >3 days

②

Decision table

Parking fee optional conditions (criteria)		Pattern specification			
Condition	Use of limited express train to/from station = "Yes"	N	Y	Y	Y
	Product spend >= $30,000_{yen}$="Yes"	-	Y	N	Y
	Cinema use = "Yes"	-	N	Y	Y
Result	Shopping discount		X		
	Cinema discount			X	
	Combo discount				X

Implementation of Thorough Impact Analysis (1/3)

FUJITSU

■ Thorough impact analysis

Using design tools and a code checker for Java resources to strictly regulate standardization constraints enables a thorough impact analysis during resource modification.

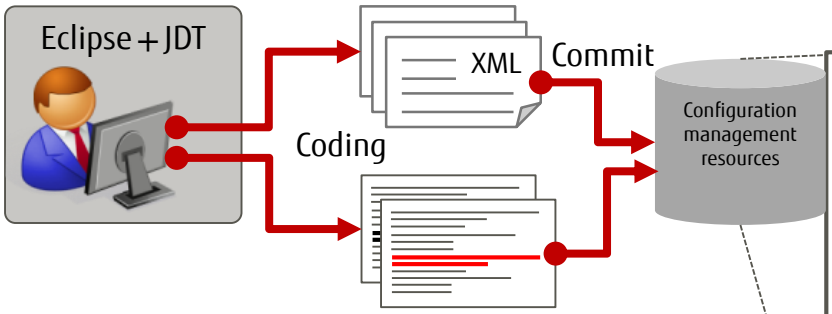
Problems with existing impact analysis

1. The Eclipse and Java hierarchy viewers both have limitations imposed by messaging and data retention (conversion to XML, etc.)
2. Given the support for highly flexible naming of internal variables, etc., 100 points cannot be collected under the standard naming conventions.= Input control imposed by JDT(*)
3. grep and other string searches require decisions on ineligible resources, leading to operational inefficiency.

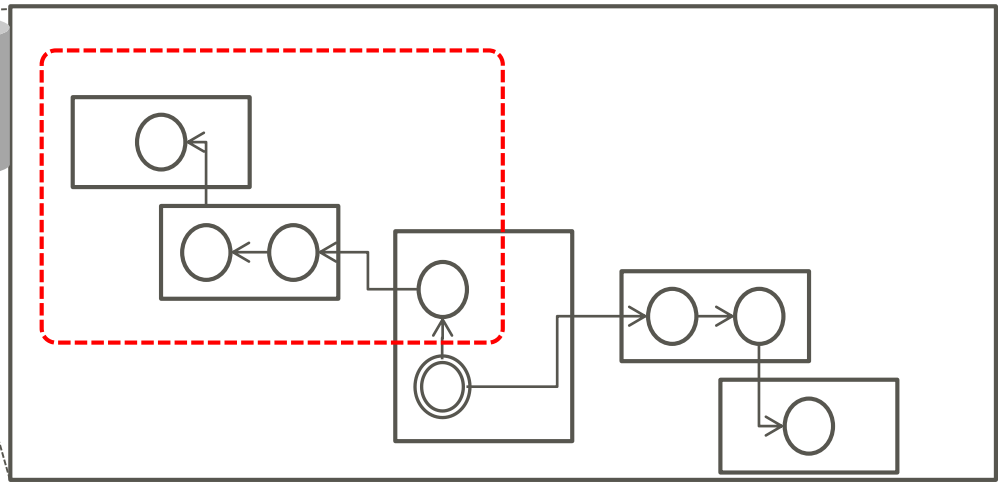
Note: JDT (Java Development Tools): Tools feature for Java source compilation, code completion, etc.

Solve problems

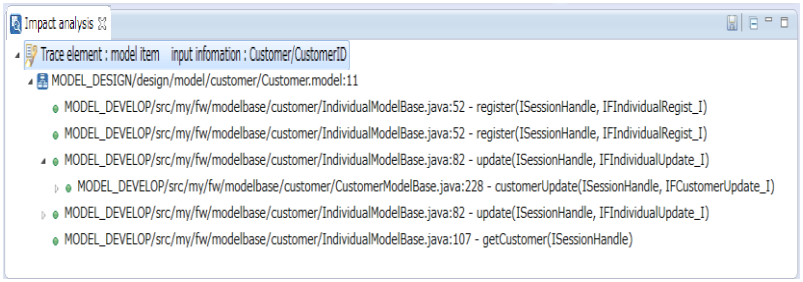
A thorough impact analysis can be implemented using design tools, resource constraints imposed by a code checker, and impact analysis viewer.



Impact analysis result (relationship chart)



Impact Analysis Viewer



: Change element

: Related resource

: Relationship

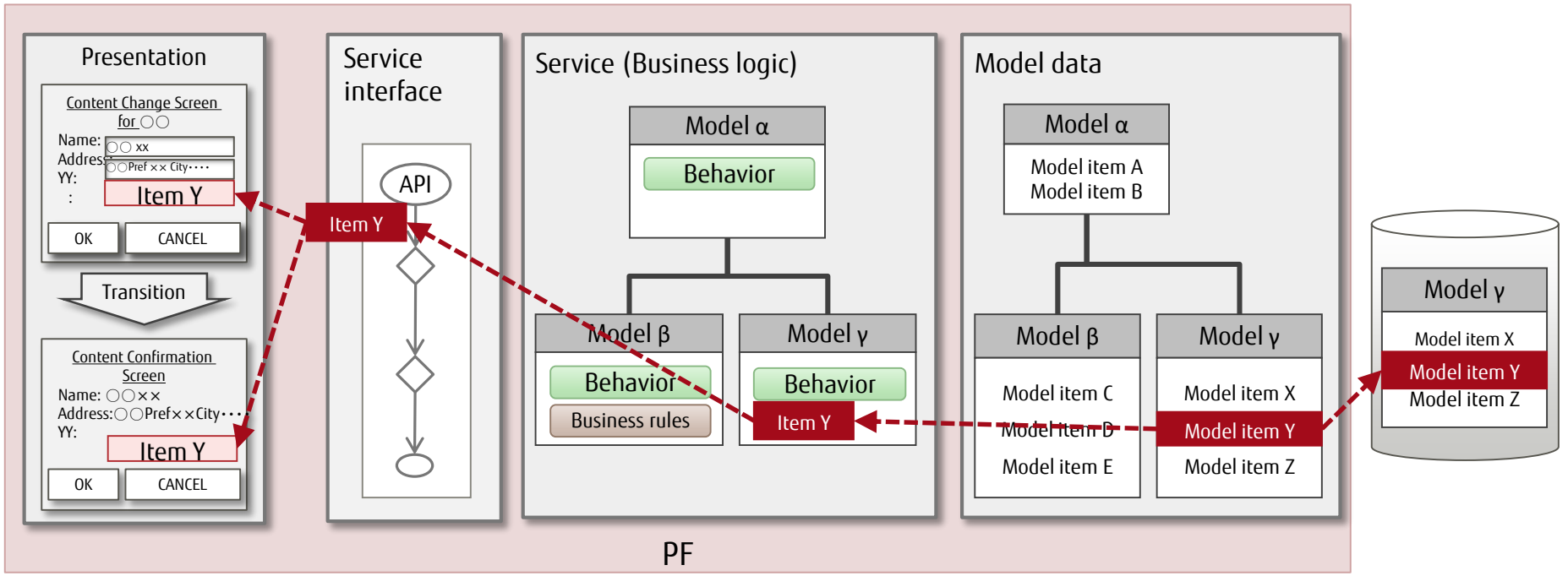
: Area identified by a user as requiring resource modification (Example)

Implementation of Thorough Impact Analysis (2/3)

FUJITSU

- Impact analysis viewer
  Thorough impact analysis is realized by using design tools to design the architecture to associate everything based on *model items*, making it self-explanatory as to *which function* is used *from where*, *by whom* and *for what purpose*. This enables the discovery of meaningful relationships, rather than merely the discovery of words, facilitating multi-level analysis.

- A framework that associates everything based on model data ensures a correct understanding of the influence of each element.

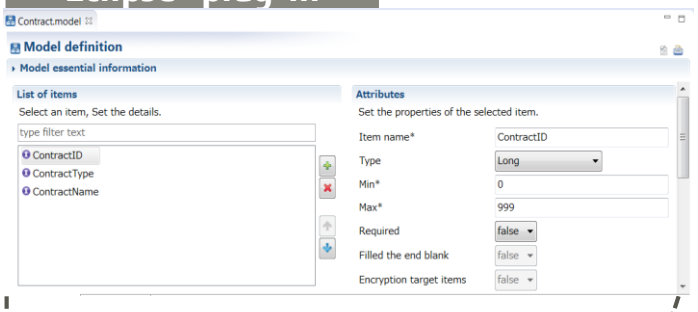


Implementation of Thorough Impact Analysis (3/3)

FUJITSU

- Practical example of thorough impact analysis
  - ① Manage model item attributes as class variables by generating the class from the design information.
  - ② Business java: Use the Key class to manipulate model items.
  - ③ The relationship between the checking of a model item and a variable is managed using a structure that sets an instance in an internal variable to ensure that the relationship is not severed.

Eclipse plug-in



Catalog class (generated by plug-in)

```
public class Keys  {

    /**
     * <DL>
     * <DT>constructor.
     * </DL>
     *
     */
    private Keys() {

    }

    /** Contract */
    public static final X31CKey<IModelBase> Contract = new X31CKey<>("Contract", "com.fujitsu.xframework.x31.X31CMo

    /** ContractID */
    public static final X31CKey<String> ContractID = new X31CKey<>("ContractID", "java.lang.String");

    /** ContractType */
    public static final X31CKey<String> ContractType = new X31CKey<>("ContractType", "java.lang.String");

    /** ContractName */
    public static final X31CKey<String> ContractName = new X31CKey<>("ContractName", "java.lang.String", "com.fujit

    /** ContractDetail */
    public static final X31CKey<IModelBase> ContractDetail = new X31CKey<>("ContractDetail", "com.fujitsu.xframewor
```

② (Contract ID)

contract.model

Item list

- Contract ID
- Contract type
- Licensee name

Item attributes ①

Item attribute	Contract type
Type	Long
Min	0
Max	9999
Mandatory	false
Space-fill to end	false
Encrypt	false

Java class (MBC coding)

```
30      // Fetch the model to be manipulated (contract model)
31      Contract contract = (contract) this.getModelBase(Keys.contract);
32      // Set interface data value in model
33      contract.setValue(Keys.contractID, contractInput.getLong(Keys.contractID));

34      // Contract type in interface data
35      String contract type_input = contractInput.getString(Keys. contract type);

36      String str = contractInput.getString(Keys.contract type);
```

③

Implement using design info as is

[E140-08]: Use either ＊model item name (IF data item name) or ＊model item name (IF item name)_＊ as the instance variable name.

Compile fails due to error

Simple Implementation of Recovery, Fault Analysis, and External Linkage

FUJITSU

- Facilitates implementation of data assurance, performance bottleneck analysis, data store linkage, and external service linkage. The following functions are provided:
  - ① Data assurance (recovery) function
  - ② Data store function
  - ③ Performance bottleneck analysis support
  - ④ External service linkage function

Data Assurance (Recovery) Function (1/3)

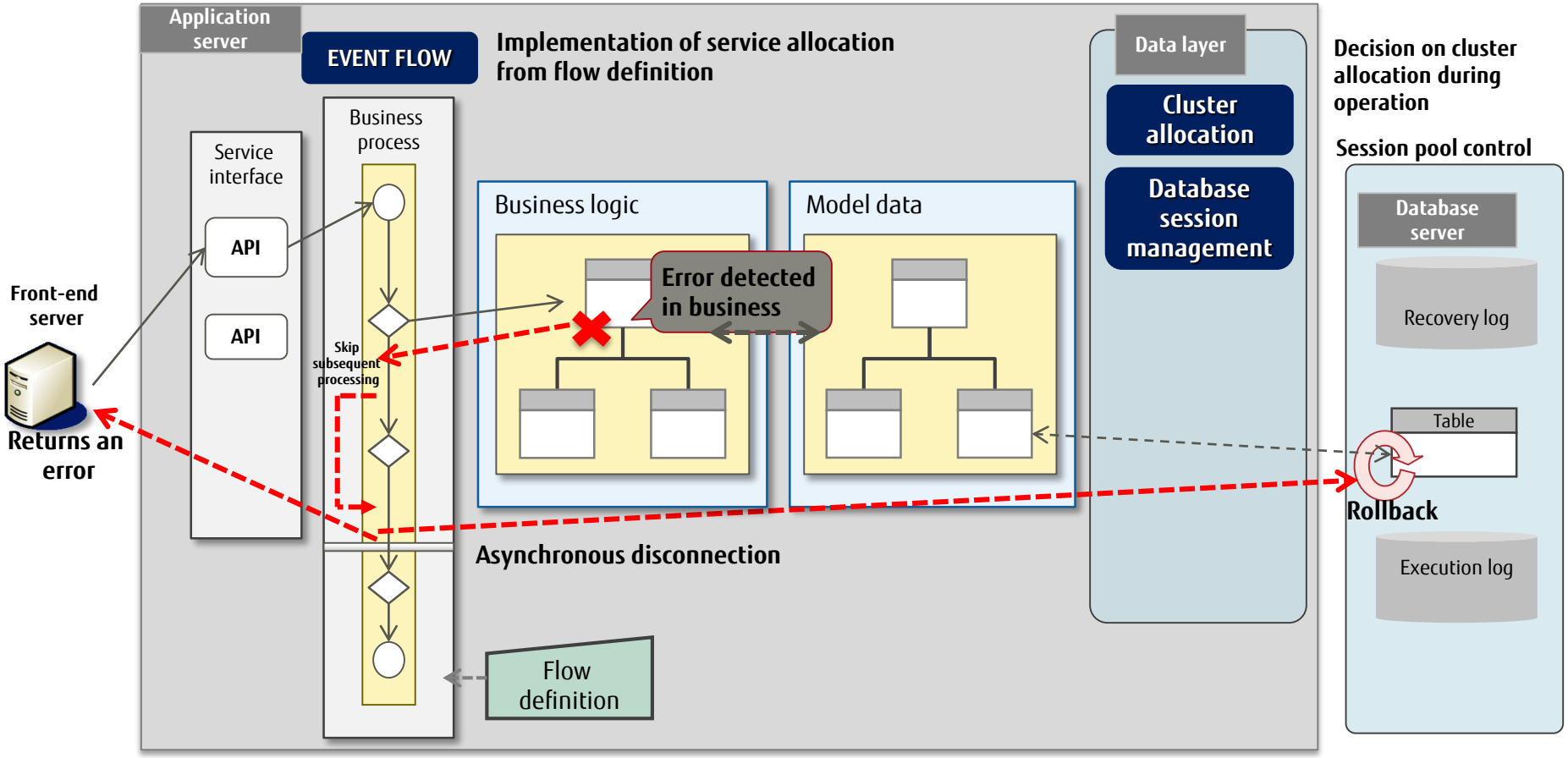
FUJITSU

■ Data assurance (recovery) function

- Ensures consistency in transactions that include database update and message update at recovery time.
- You can select Cancel recovery and Forward recovery.

Image: Cancel recovery control

If an error occurs during synchronous processing, you can skip subsequent processing and roll back the database to the data update.



Data Assurance (Recovery) Function (2/3)

FUJITSU

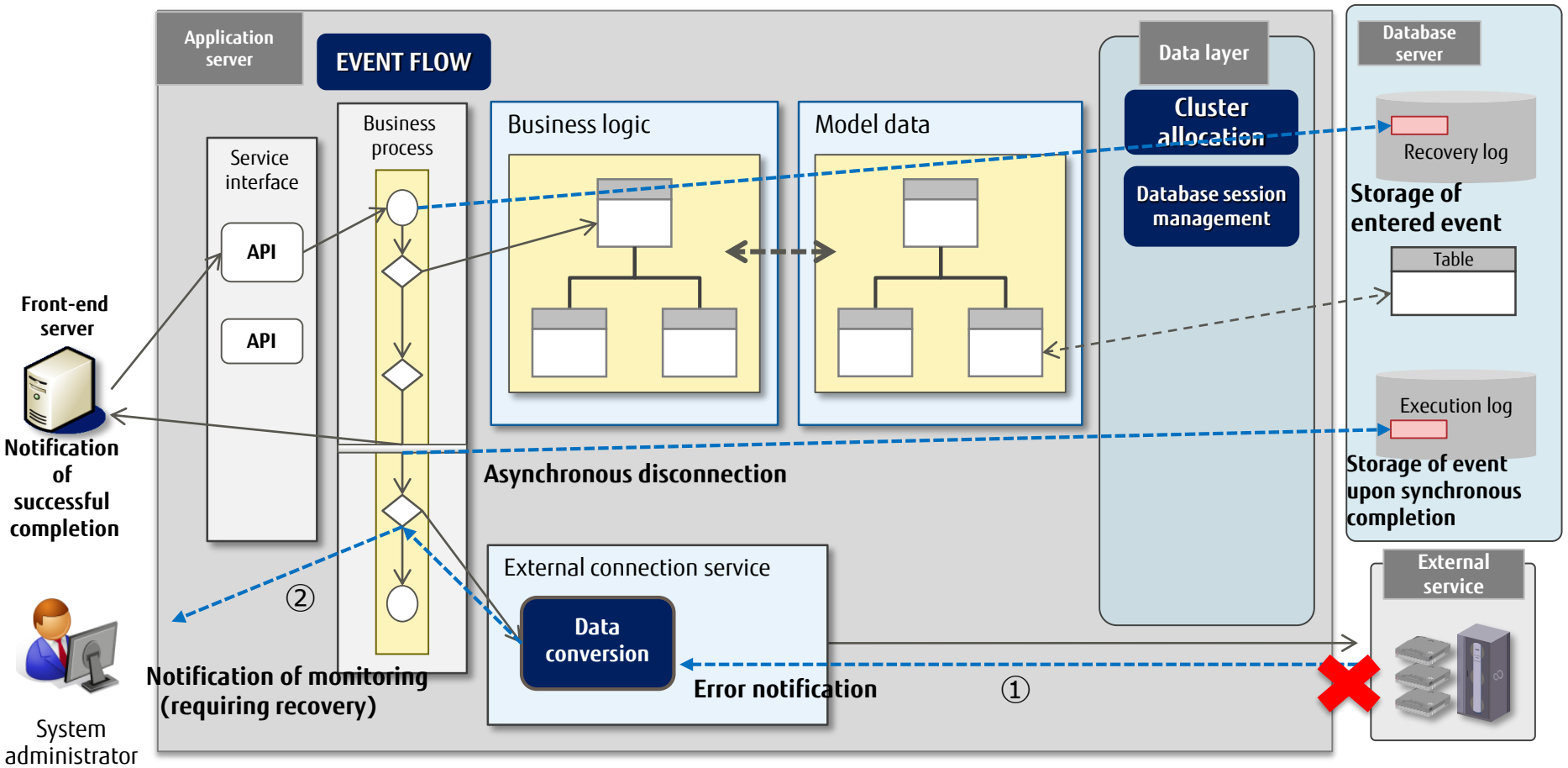
Image: Forward recovery control

If an error occurs during asynchronous processing, the asynchronous part can be re-executed simply by re-entering the event.

- Forward recovery operation (continued on next slide)

① An error occurs during asynchronous processing.

② The system administrator is notified.



Data Assurance (Recovery) Function (3/3)

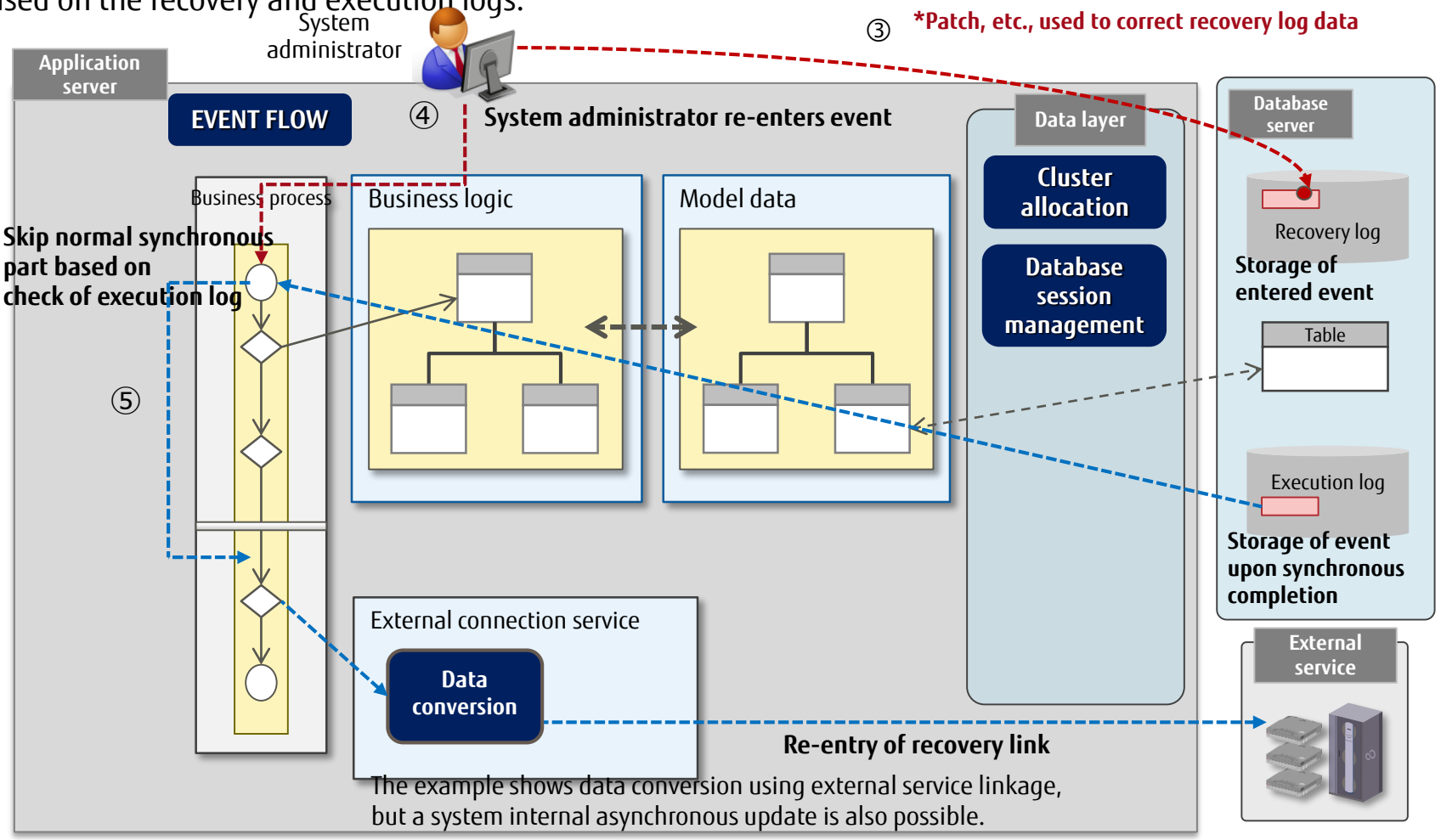
FUJITSU

- Forward recovery operation (continued from previous slide)

③ The system administrator uses a patch, etc., to correct the recovery log data.

④ The system administrator re-enters the event in which the error occurred.

⑤ The data assurance function is used to skip the part that completed normally and re-execute the synchronous part based on the recovery and execution logs.

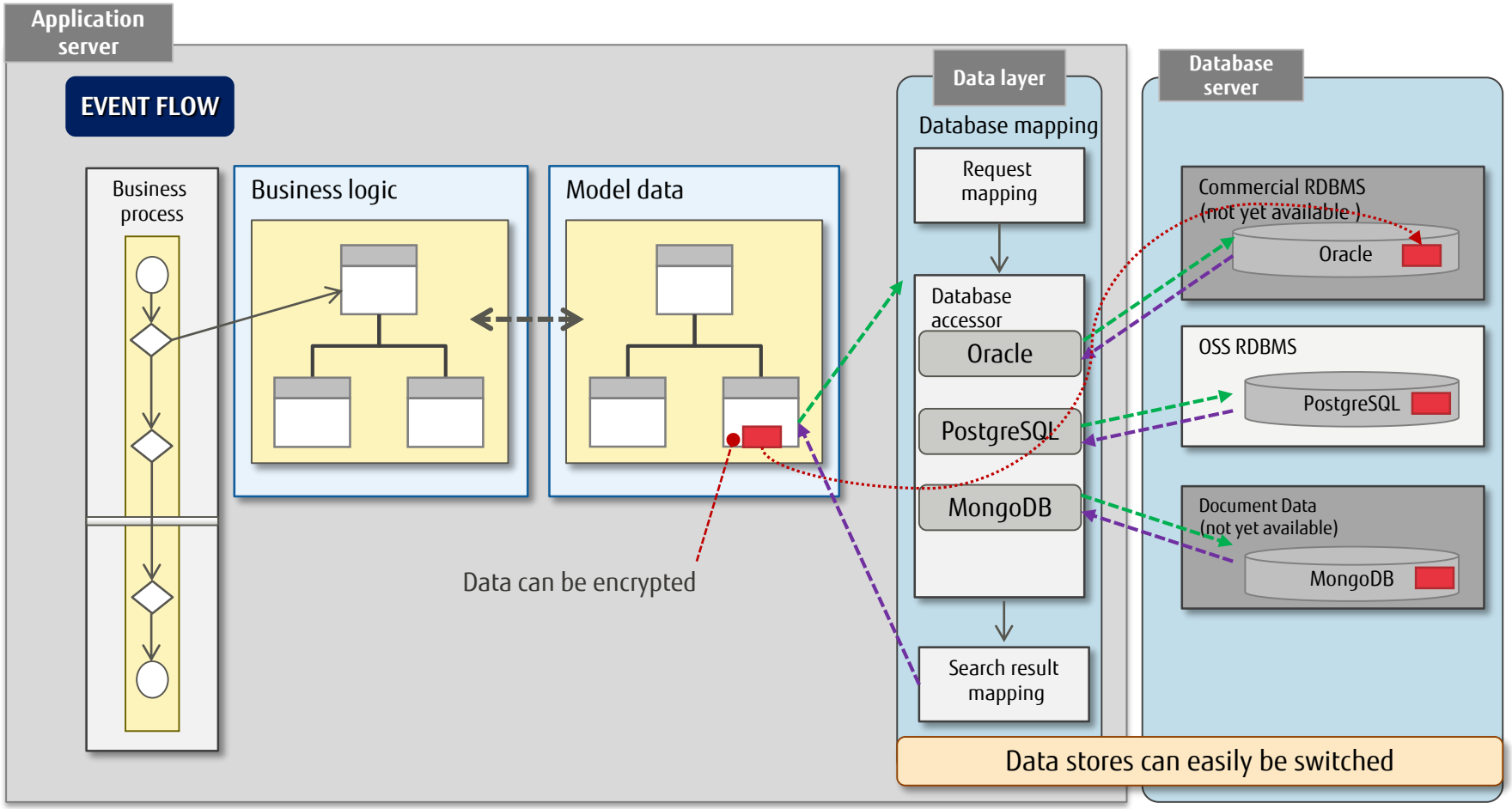


Data Store Function

FUJITSU

Data Store Function

- Data stores can easily be switched without changing applications.
- Data can be encrypted and stored by specifying in the model definition that the model item is to be encrypted.

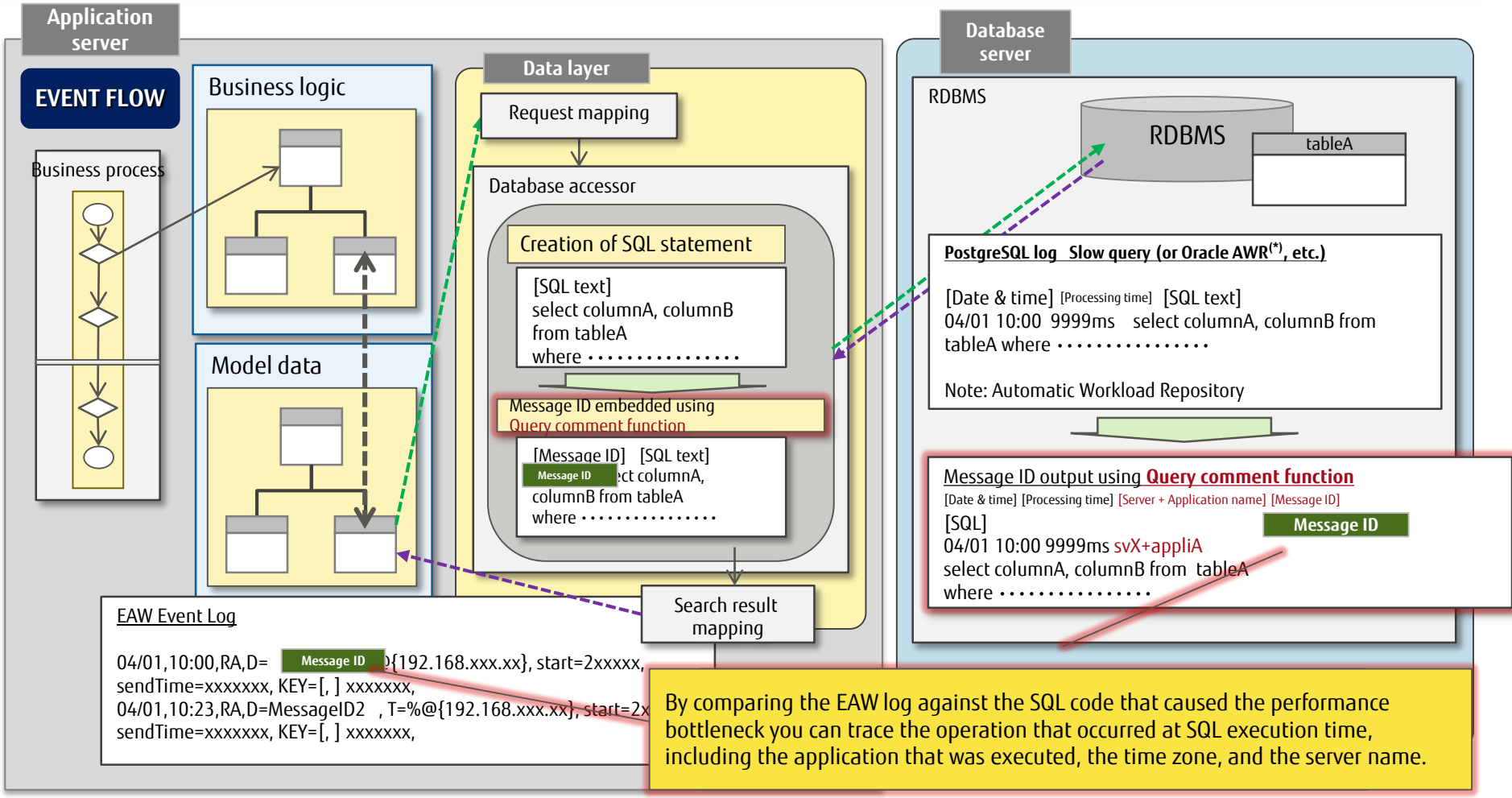


Performance Bottleneck Analysis Support (1/2)

FUJITSU

- Query comment insertion during SQL query execution

  Embedding a message ID in the SQL query makes it easier to identify choke points, thus reducing the hours of labor involved in investigating performance issues.

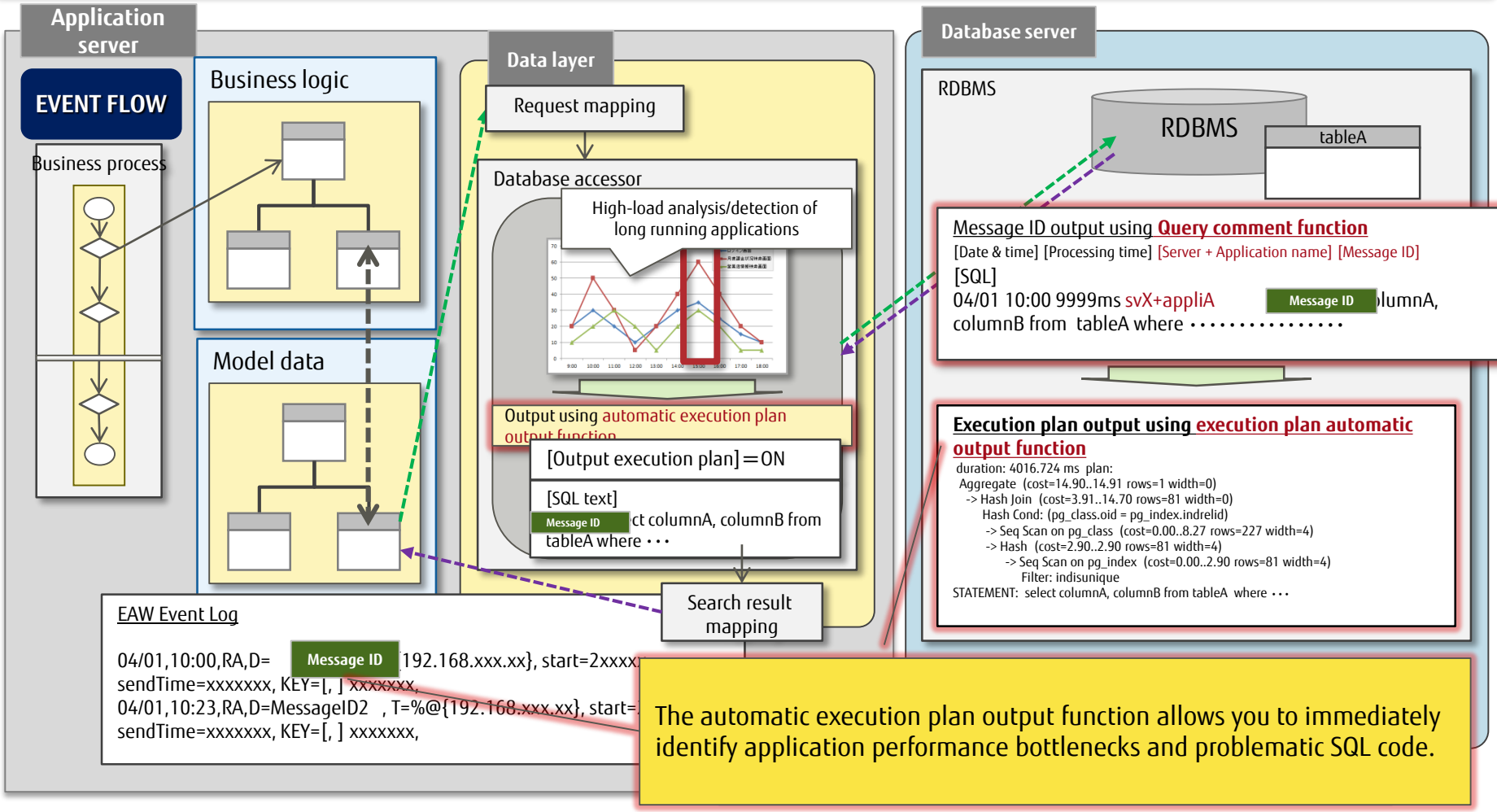


Performance Bottleneck Analysis Support (2/2)

FUJITSU

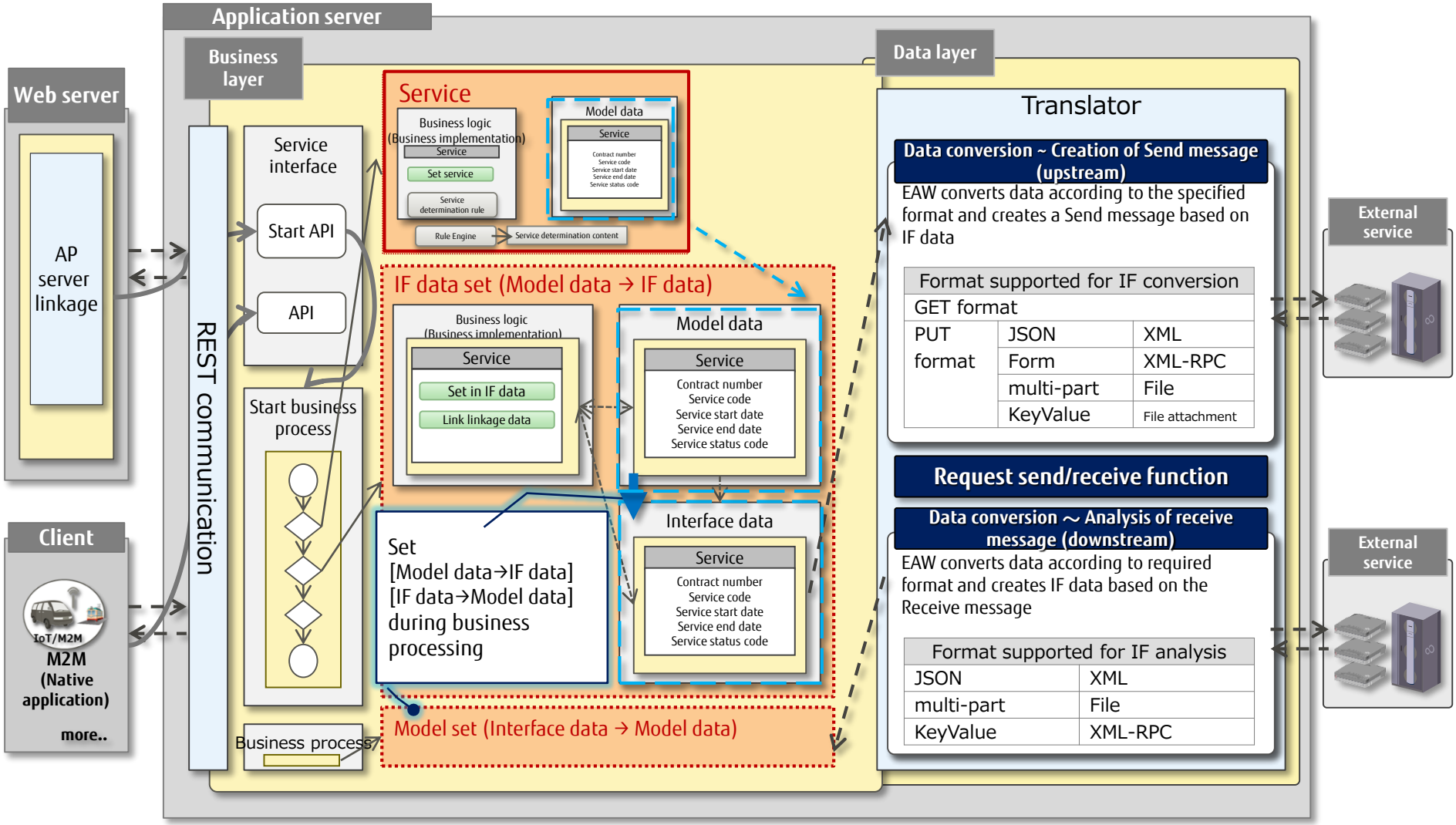
Automatic execution plan output function

Outputting an execution plan during SQL execution makes it easier to identify problematic applications and SQL code, thus reducing the hours of labor involved in investigating performance issues.



External Service Linkage Function

- Provides a simple mechanism for data linkage with external services
  Provides a data conversion function and a request send/receive function



Supplied Plug-ins

FUJITSU

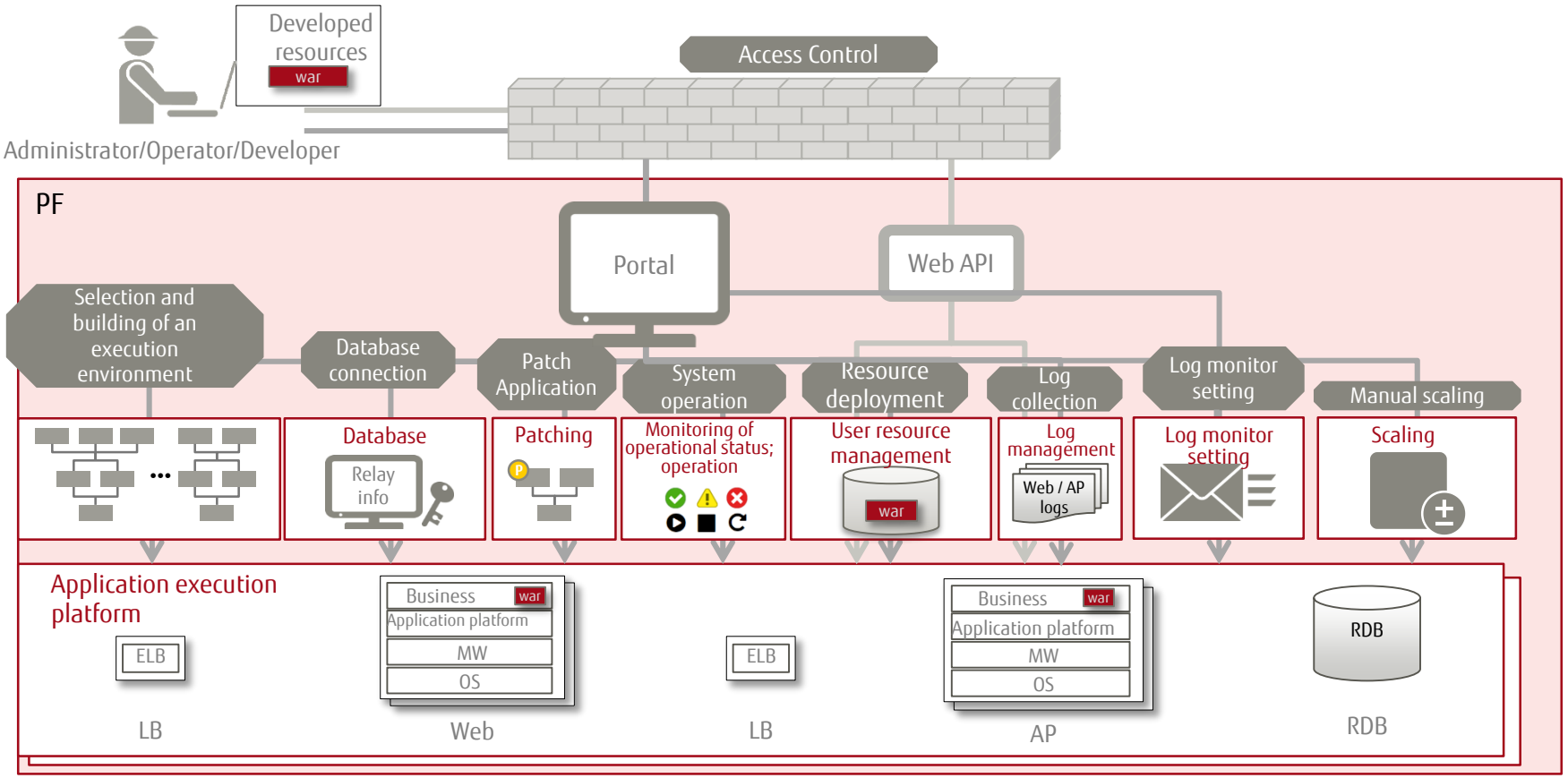
List of the plug-ins provided

Plug-in name	Description
Model and Rule Definitions	Defines the models, rules, the database tables connected to models, and the manipulations (search criteria, etc.) of those tables
SimpleEventFlow (BPM) Definition	Defines the execution sequence (flow control) of models, on a per-use-case basis
URI Mapping Definitions	Defines the connection between the external access REST interface and the application developed by the licensee
Code Checker	Checks whether Java resources conform to Java coding standards
Impact Analysis	Conducts analysis of model definitions, rules definitions, Java resources, etc., analyzing the potential consequences of resource modification

Rapid, Robust Construction and Deployment of an Execution Platform

FUJITSU

Provides an application execution platform that can also activate mission-critical systems. Simply select the execution environment definition (system configuration package), deploy the resources you have developed, and the system build is complete. Functions are also provided to monitor the operational status and for browsing operation screens and logs.



Rapid, Robust Construction and Deployment of an Execution Platform

FUJITSU

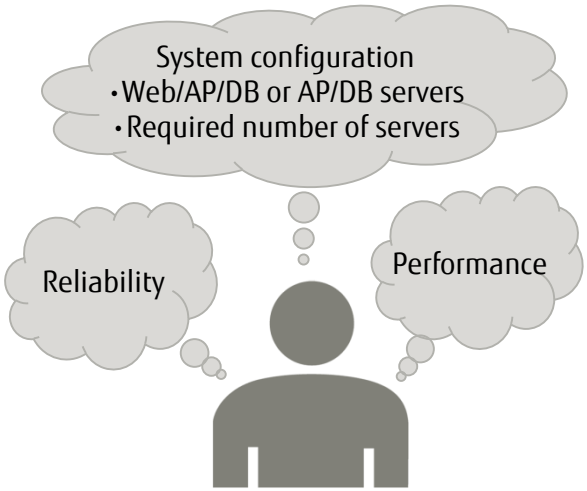
List of functions provided

No	Function	Description
1	Application execution platform	Provision of system configuration packages Building of an execution platform
2	User resource management	Registration and deployment of developed resources
3	Monitoring of operational status; operation	Displays the system's operational status (Start/stop/restart)
4	Log management	Collection of system logs and business logs
5	Database	Provision of database connection information
6	Patch Application	Displays the patch list and applies patches
7	Access control	Settings to allow or prohibit access to the execution platform
8	Manual scaling	Scaling (in/out) of the Web and AP servers
9	Log monitor setting and email recipient setting	Monitors logs and sends notifications by email

Application Execution Platform (1/2)

FUJITSU

- You can select the application execution environment definition (system configuration package) from the K5 Portal catalog according to conditions such as size and reliability.
- A system configuration package allows you to **easily build** and **instantly use** an application execution platform.
- Each system configuration package supports auto-scaling.



Establishment of system configuration, performance, and reliability requirements

Selection and deployment of a system configuration package that meets requirements

Instant access and use

Application Execution Platform (2/2)

FUJITSU

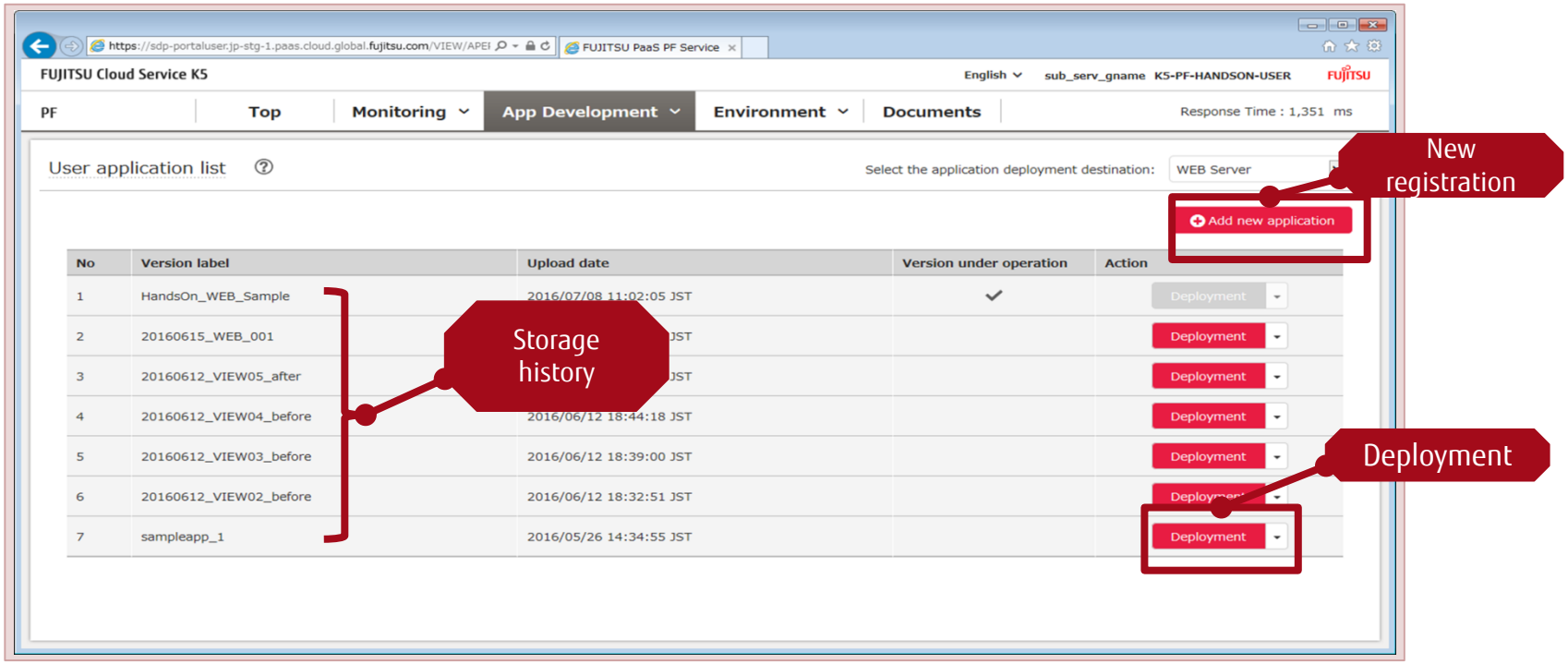
- System configuration packages are integrated packages for building execution platforms for applications that include environment definitions for web/application/database servers and load balancers. The packages **have been verified for robustness and operability,** eliminating the need for customers to design the environment themselves.

System configuration package features

Category	System configuration packages address the following:
Performance & scalability	・Preparation of system configuration tailored to system size ・Auto-scaling support (in future)
Reliability	・The ability to distribute a system across multiple availability zones (in future) ・Database redundancy (in future) ・Disaster readiness through multi-regional configuration (in future)
Security	・Fujitsu has completed Security audit ・Architecture logically separated from other customers ・Access control has been set via a security group ・Ability to select an IDS/IPS service (in future)
External connection	・Mechanism for communicating with multiple availability zones/regions or other sites (internet connection, IPsec VPN connection, leased line connection) (in future)
Operation & maintenance	・Application monitoring and operation ・Collection of system logs and application logs ・Listing/Applying patches ・Ability to implement blue-green deployments (in future)

User Resource Management

- New registration, deployment, and storage history for application resources (.WAR files) **is simple, reducing the load** on administrators and operators.
- Registered resources can be deployed at the touch of a button. Generation management is based on storage history, allowing instant resource rollback.
- By considering downtime at resource deployment time, the deployment group setting allows web/application server restarts to be scheduled.

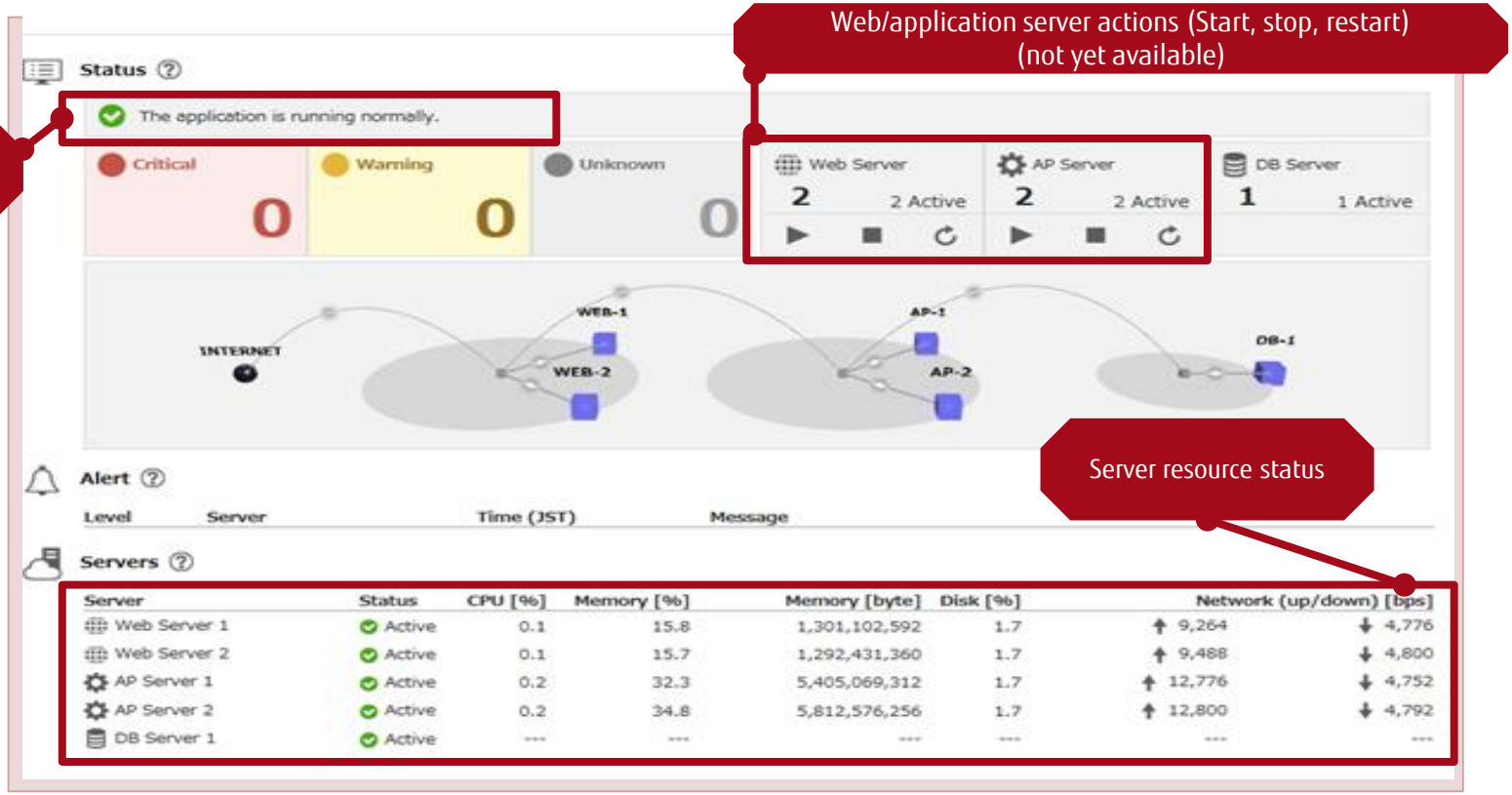


Monitoring of Operational Status; Operation

- A dashboard is used to display the operational status of the application execution platform in real time. Icons allow the user to **visually and easily determine** the overall operational status of the application execution platform.
- Web and application servers can be started, stopped, or restarted.

Monitoring of operational status

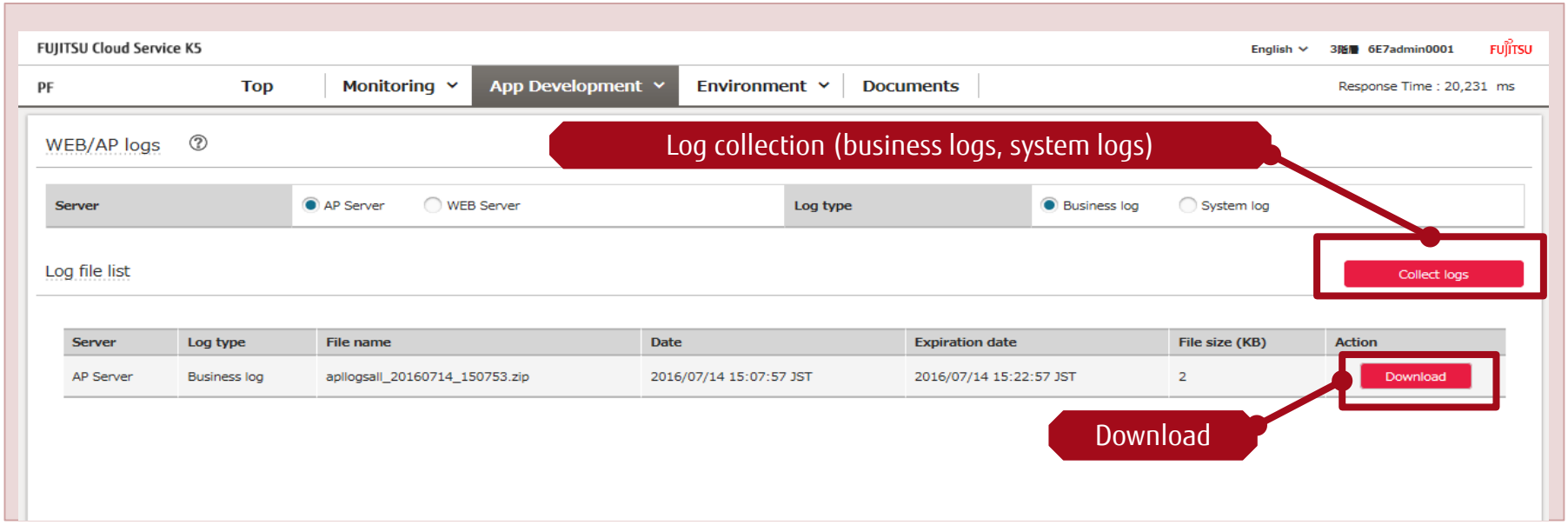
- Normal state
- Requires action
- Error state



Log Management

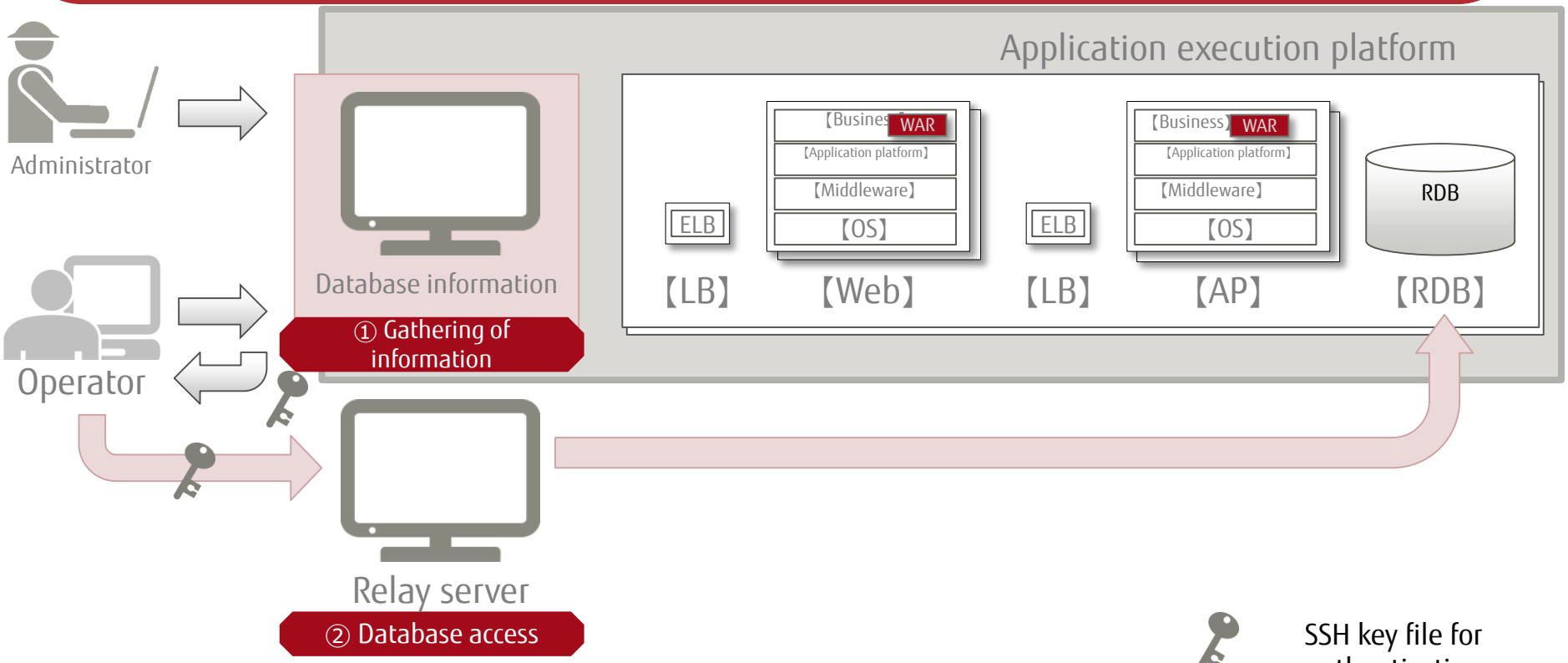
- Administrators and users can collect system logs(*), application logs (business logs), and database logs without regard to the type or number of servers.
- System logs and business logs **are automatically backed up on a daily basis**.
- The current day's system logs and business logs, as well as previously backed up system logs and business logs, can each be **downloaded in ZIP format** for browsing.
- Database logs can be **viewed via a browser**.

Example: Log browsing　WEB/AP log screen



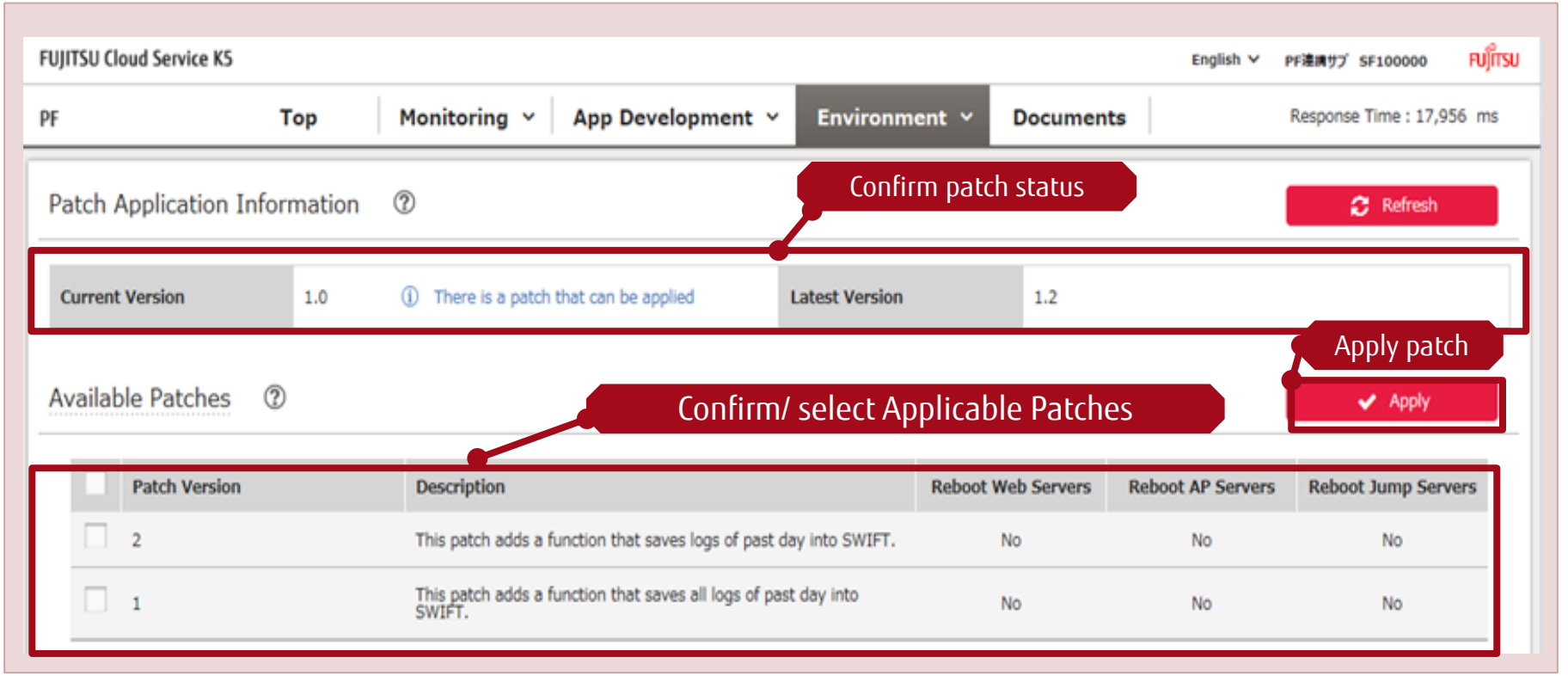
Database

- Provides a database service that uses PostgreSQL.
  Databases can be manipulated from a console.
- There is no need for managers and operators to build databases; they need only use the schemas provided to create tables, allowing **databases to be used immediately.**
- A relay server is deployed, taking into account the security implications of database access.



Patch Application

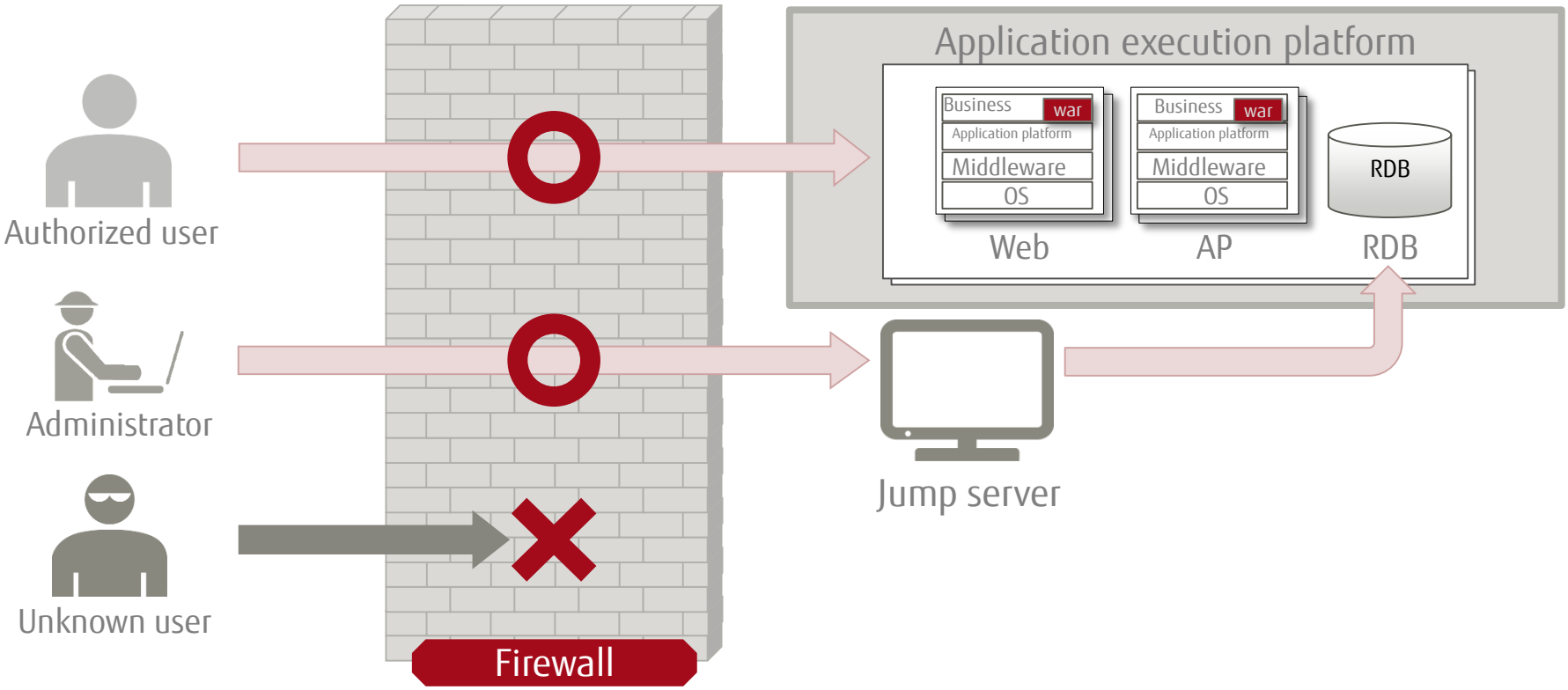
- Software and other patches from the Applicable Patches list can be used.
- Administrators and users can apply patches from the **predefined patch set** on-screen, without having to consider each individual patch.
- Patches can be applied from the screen at a time that suits the customer.



Access Control

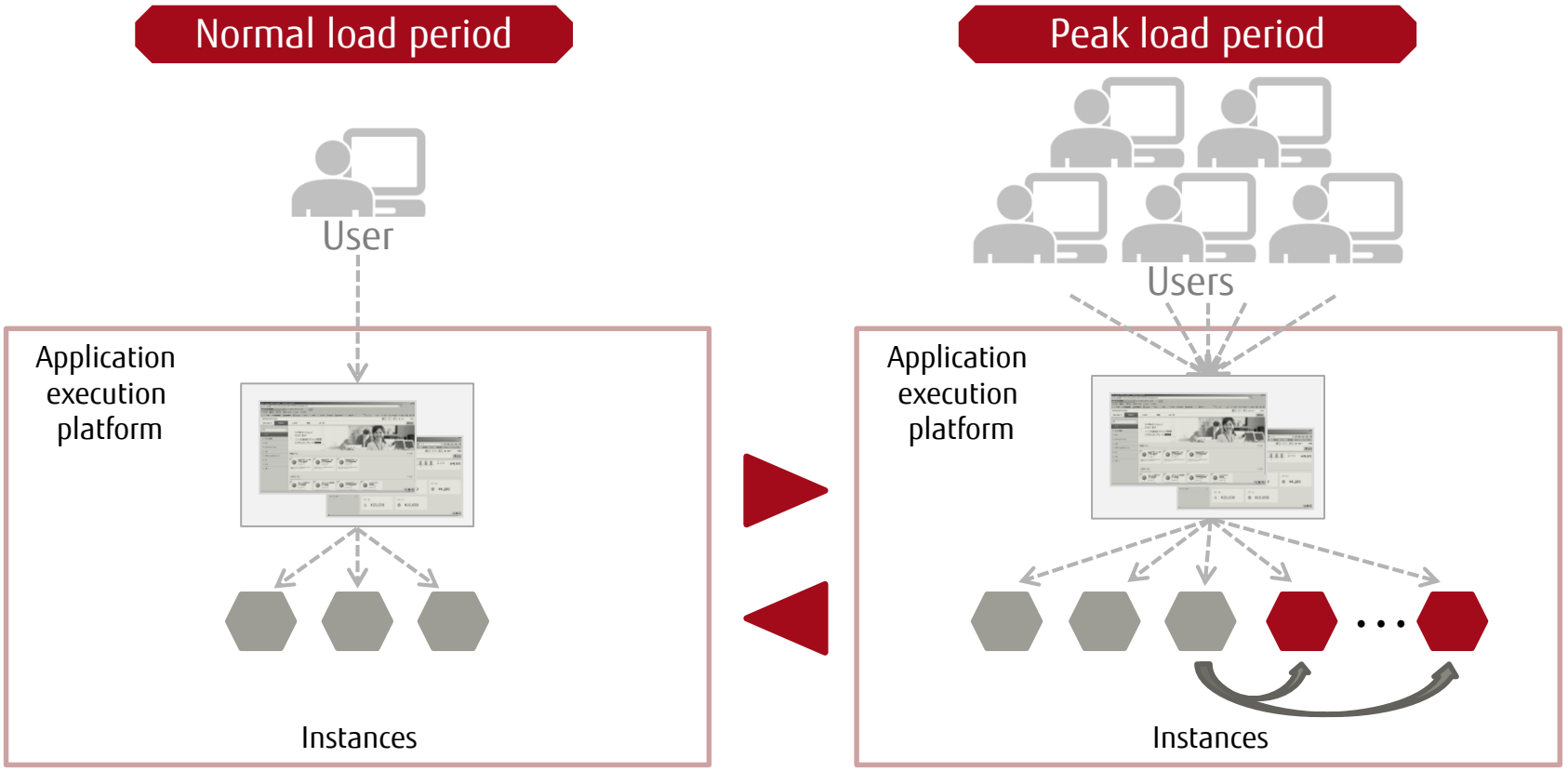
- Provides a firewall-based access control function.
- The administrator is able to **increase the security** of the environment by controlling user access to web servers, application servers and jump servers.
- Access control settings can be easily registered on-screen.

Note: Access to web servers, application servers and jump servers is controlled based on originating IP addresses.



Manual Scaling

- Supports a scaling function to manually increase (decrease) web or AP servers
- Instances can be increased or decreased according to the load status
- Instances can be increased or decreased from the screen

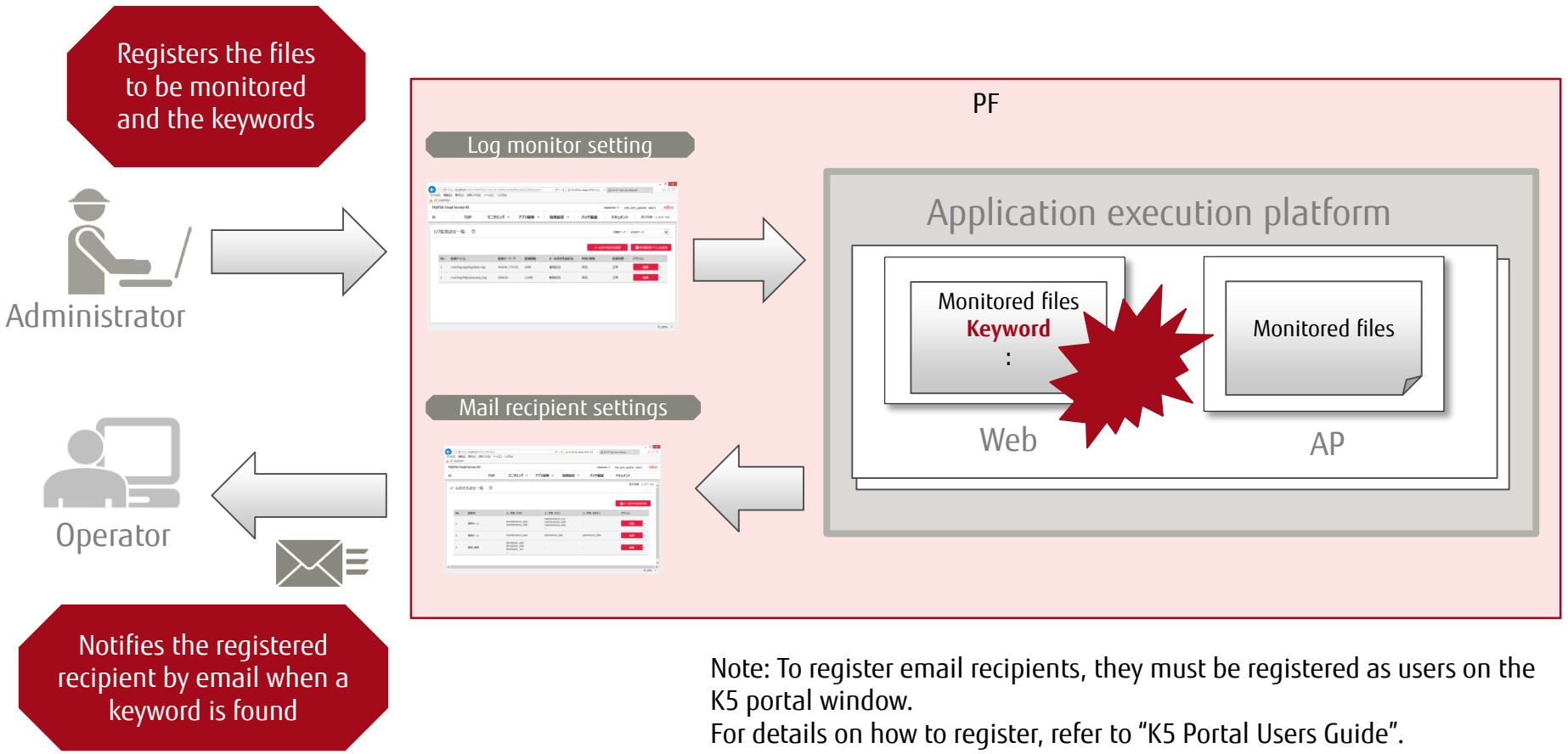


Operates according to the number of instances for the selected system configuration package

Manually increases the number of instances according to load status, reducing the number of instances under light load conditions

Log Monitor Setting and Email Recipient Setting

- Registers and monitors keywords to be searched in log files being output to the Web server/AP server
- Notifies the registered recipient by email when a keyword is found

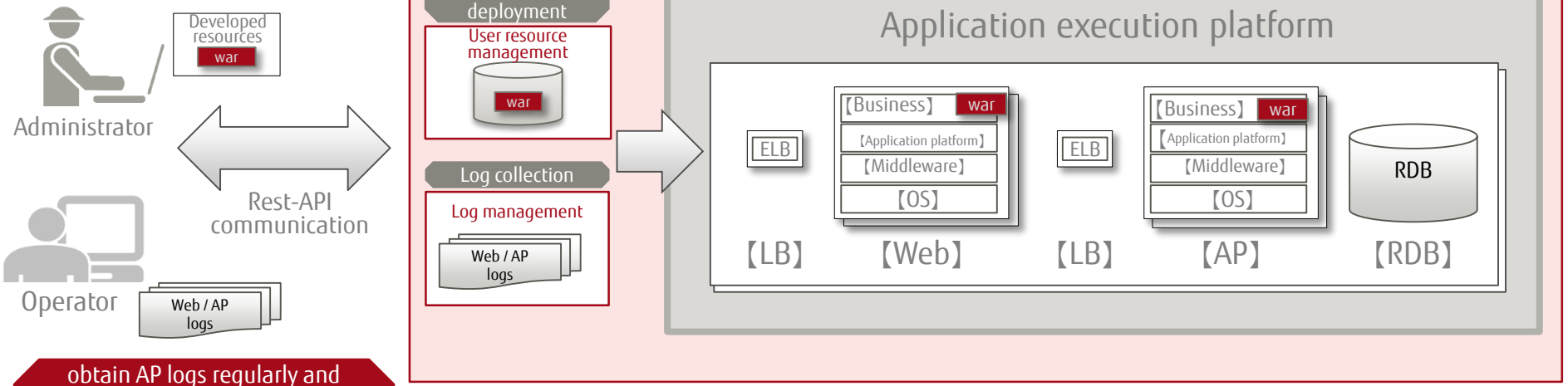


Note: To register email recipients, they must be registered as users on the K5 portal window.
For details on how to register, refer to “K5 Portal Users Guide”.

Web API

- A Web API function*1 is provided to manage the application execution platform.
- Cooperating with CI tools*2 used by customers allows the **automation of user resource registration and deployment.**
- Furthermore, it is possible to **obtain AP logs regularly** and automatically by developing application using the API.

Automate resource registration and deployment by cooperating with CI tools



obtain AP logs regularly and automatically by developing application using the API

*1 For a list of provided Web APIs, please refer to K5 Portal>Documentation>Manuals>PF>Web-API.
*2 Tools that support Continuous Integration.

Explanation of Billing Model

Billing using *Fixed monthly fee + Pay-per-use*

Fixed monthly fee

This is a monthly fee fixed according to the individual contents of the system configuration package selected at the time the application implementation platform is built.

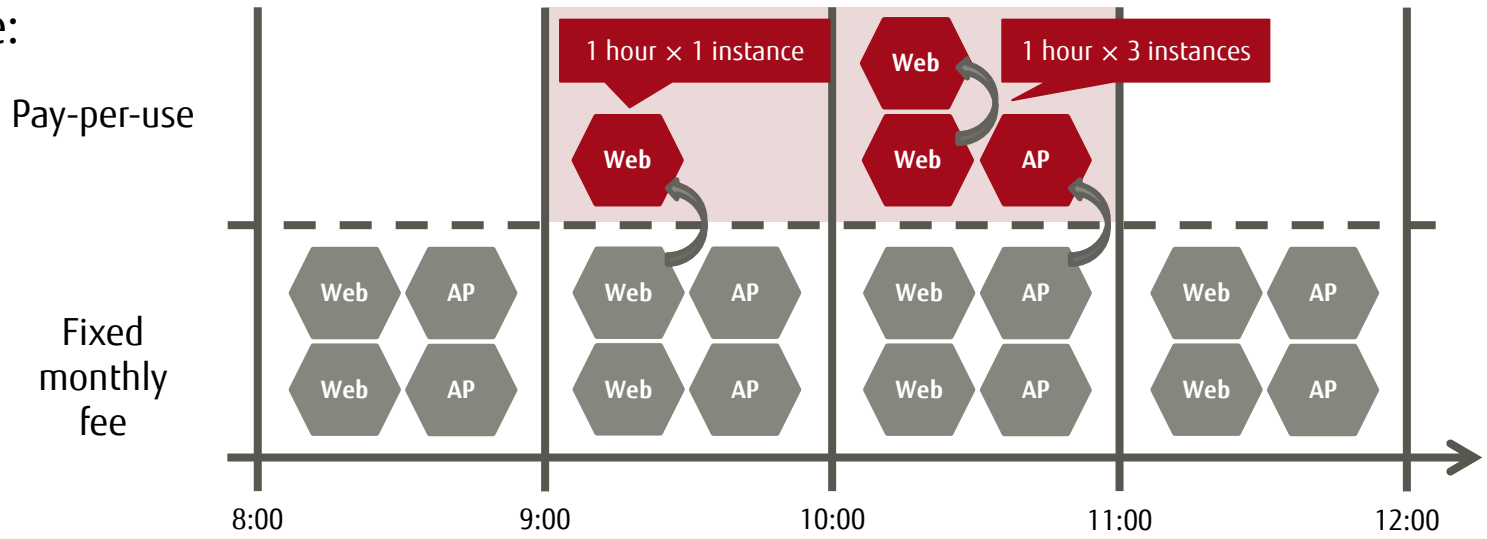
Pay-per-use

If the number of instances in the chosen system configuration package is exceeded as a result of manual scaling, this is billed as *excess instance runtime x instance count*.

Runtime is calculated in hourly units, rounded up to the next whole number.

For example, a run time of 1 hour 45 minutes is rounded up to 2 hours.

Example:



Restrictions and Notes

- The following functions will be provided in due course:
  - Deployment group settings in User Resource Management
- The medium and large System Configuration Package models and the database redundancy model will be provided in due course.
  Note: See “Attachment: System Configuration Package List”.
- Refer to the Service Description on FUJITSU Cloud Service K5 Website to confirm the regions in which this service is offered.
- A Client ID is required to use the Web APIs and can be obtained via the PF Service application screen. When using Web APIs in an environment in which the application process has been completed, a Client ID can be obtained via the user service screen on the K5 Portal.
- The time required from application to start of service is as follows:
  - Within two business days from completing the application via the service settings application screen on the K5 Portal.

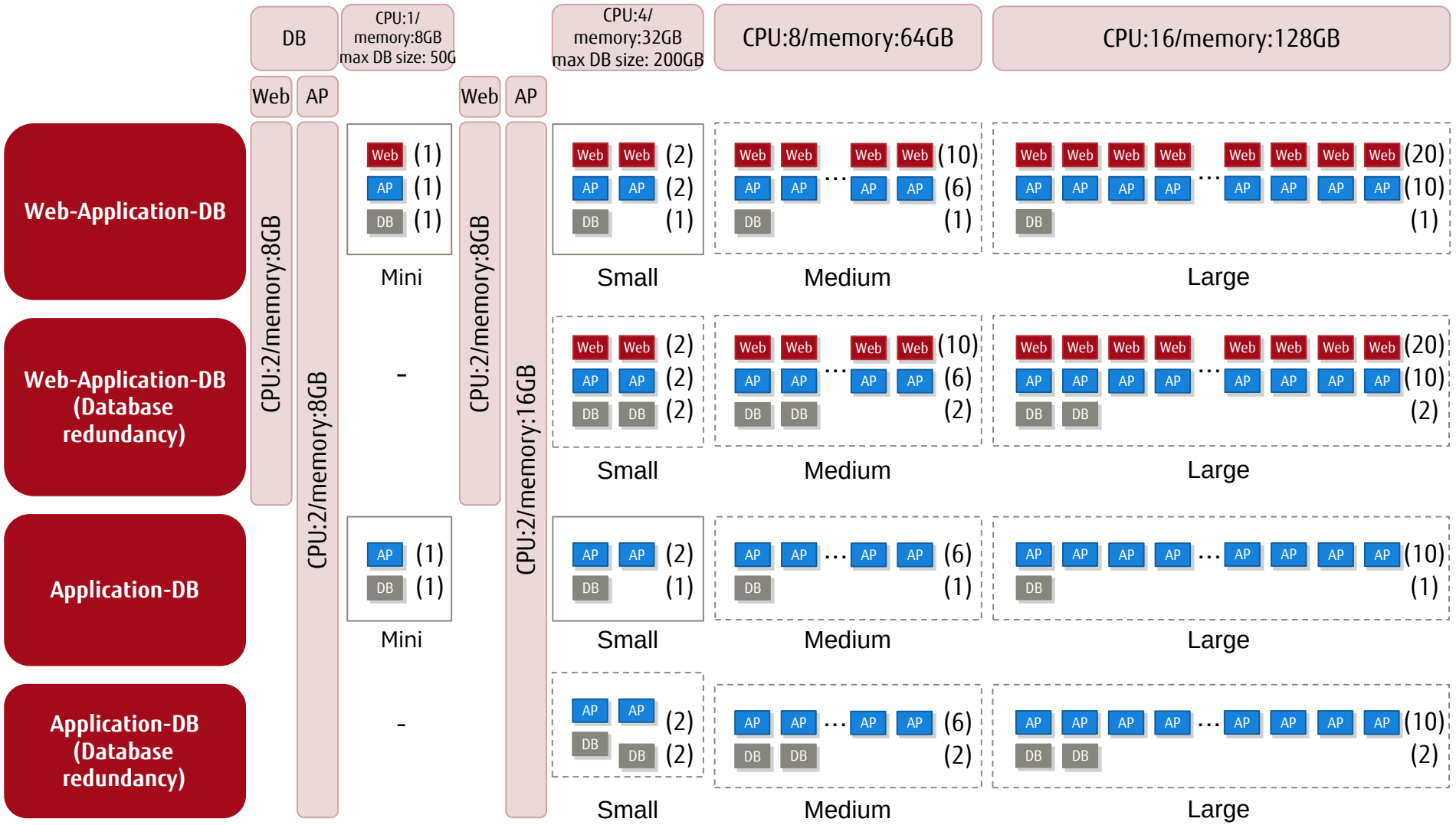
Attachment: System Configuration Package List

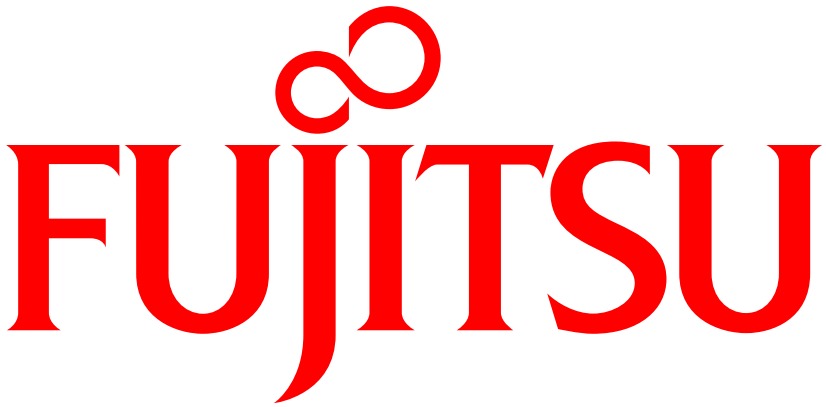
FUJITSU

- The system configuration packages provided by Fujitsu are:

Legend: Current Model / Future Model

Max.no. of instances at scale-out (including initial instances) ・Minimum scale: 5 Webs and 5 APs ・Small: 10 Webs and 10 APs





shaping tomorrow with you