



Overview

ver.1.0

Exastro developer

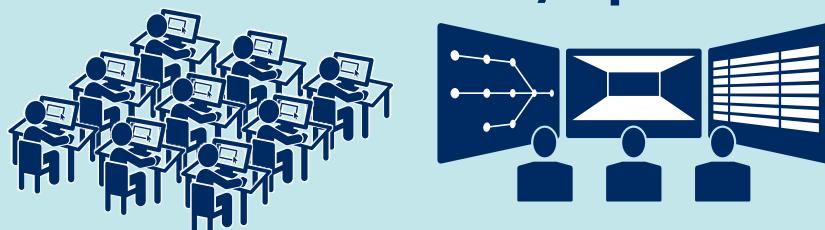
Introduction – What is “Offensive” and “Defensive” Automation?

Hard to incorporate new technology

Tightly grouped applications

Traditional static Environments

Manual Construction/Operation



Monolithic (Physical Machines)

When Virtualization technology was still undeveloped and young...

- ✓ In order to avoid services stopping
〔Highest traffic x buffers x 2x redundancy〕
were physically arranged.
- ✓ Additionally, there were few options for what kind of CPU, Memory and I/O to use,
Resulting in resources not being fully used.
- ✓ Even so, it was considered “correct” to **integrate multiple functions** in order to handle a large number of services with as few devices as possible
- ✓ **As a result, Apps were built on this premise.**

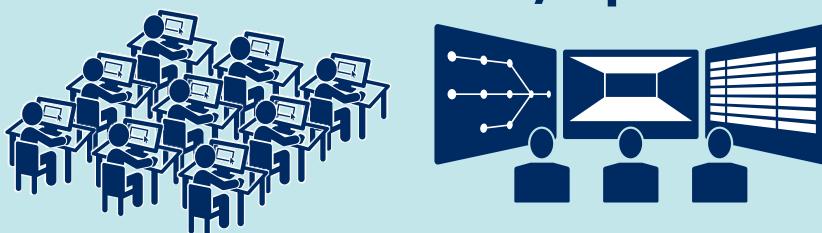
Introduction – What is “Offensive” and “Defensive” Automation?

Hard to incorporate new technology

Tightly grouped applications

Traditional static Environments

Manual Construction/Operation



Monolithic
(Cloudlift)

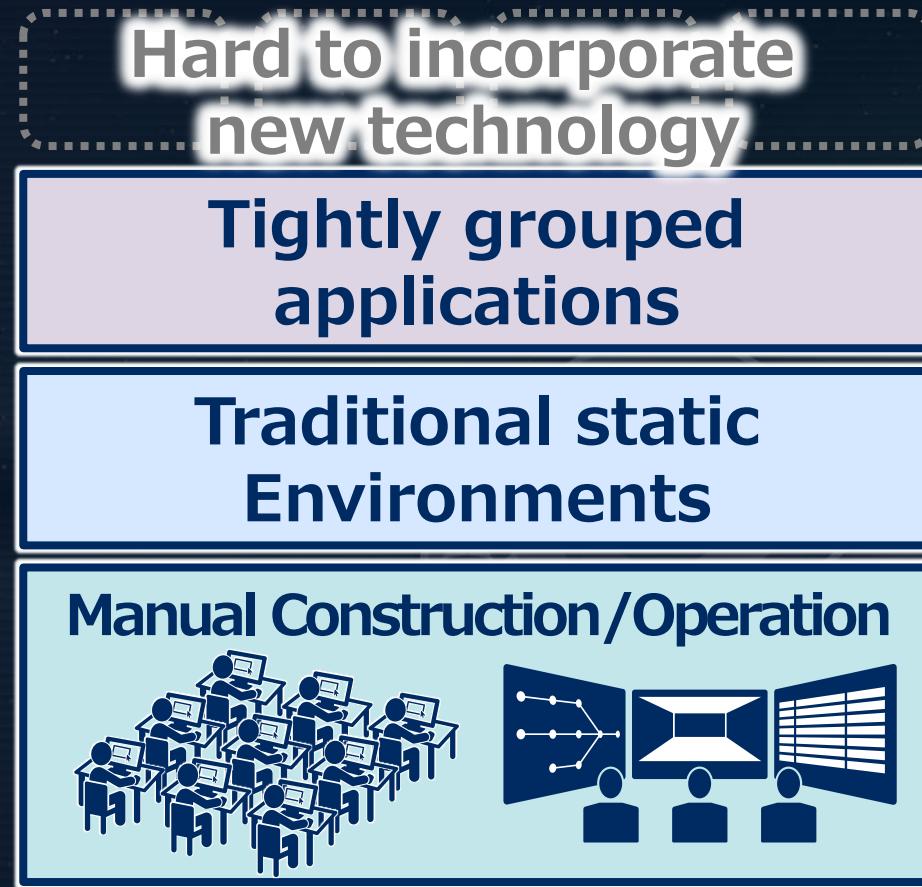
Thanks to virtualization technology...

- ✓ We can now freely select CPU, Memory and I/O
- ✓ It is now possible to create **Simple Dynamic systems** easily
- ✓ The size of the systems became more compact and we could get rid of end-of-life hardware.
- ✓ **Not mixing multiple functions** became the “correct” thing to do.

However, The following issues remains unsolved

- ✓ Many function's redundancy **continues being doubled** unless apps are redesigned.
→Expectations for **Cloudshift**... ①
- ✓ The shape of the system is not changing, **meaning that the amount of operation labor is not changing as much.**
→Expectations for **Automated operation** ... ②

Introduction – What is “Offensive” and “Defensive” Automation?



**Monolithic
(Cloudlift)**



Expectations for Cloud shift(①)

Improved OPEX efficiency.

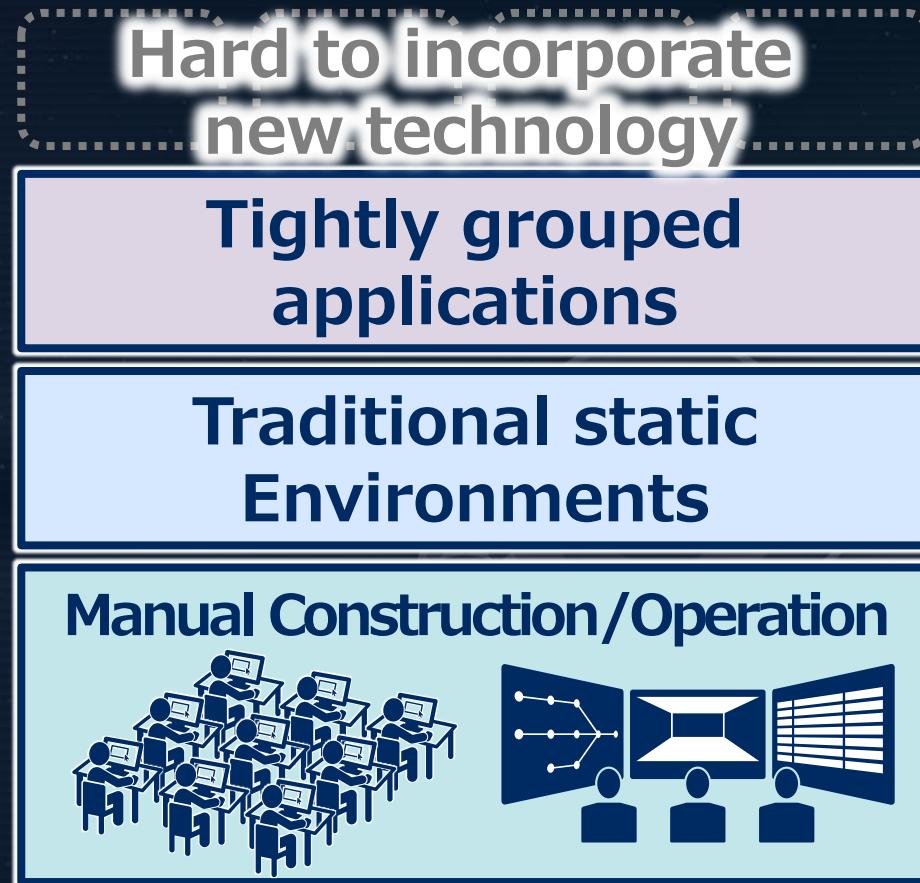
A 「**Full-fledged dynamic system**」 that uses necessary resources when they are needed by incorporating automation from the design stage.

(Gets rid of redundant resource allocation)

Increased ROI (Return On Investment)

By incorporating automation from the app design stage, we can add an 「**easily disposable system**」 that can add new functions even while running.(Realization of DevOps)

Introduction – What is “Offensive” and “Defensive” Automation?



**Monolithic
(Cloudlift)**



Problems with Cloud Shifting

- ✓ Have to **Rebuild apps.**
- ✓ Hard to insert **new mechanics during the initial setup.**

⇒ In other words, the threshold is too high !

Users have to frequently select the part they want to modify.

Introduction – What is “Offensive” and “Defensive” Automation?



↑ Tech
↑ AP
↑ PF
↑ OP

Introduction – What is “Offensive” and “Defensive” Automation?

Hard to incorporate new technology...

Tightly grouped applications

Traditional static Environments



Monolithic (Cloudlift)

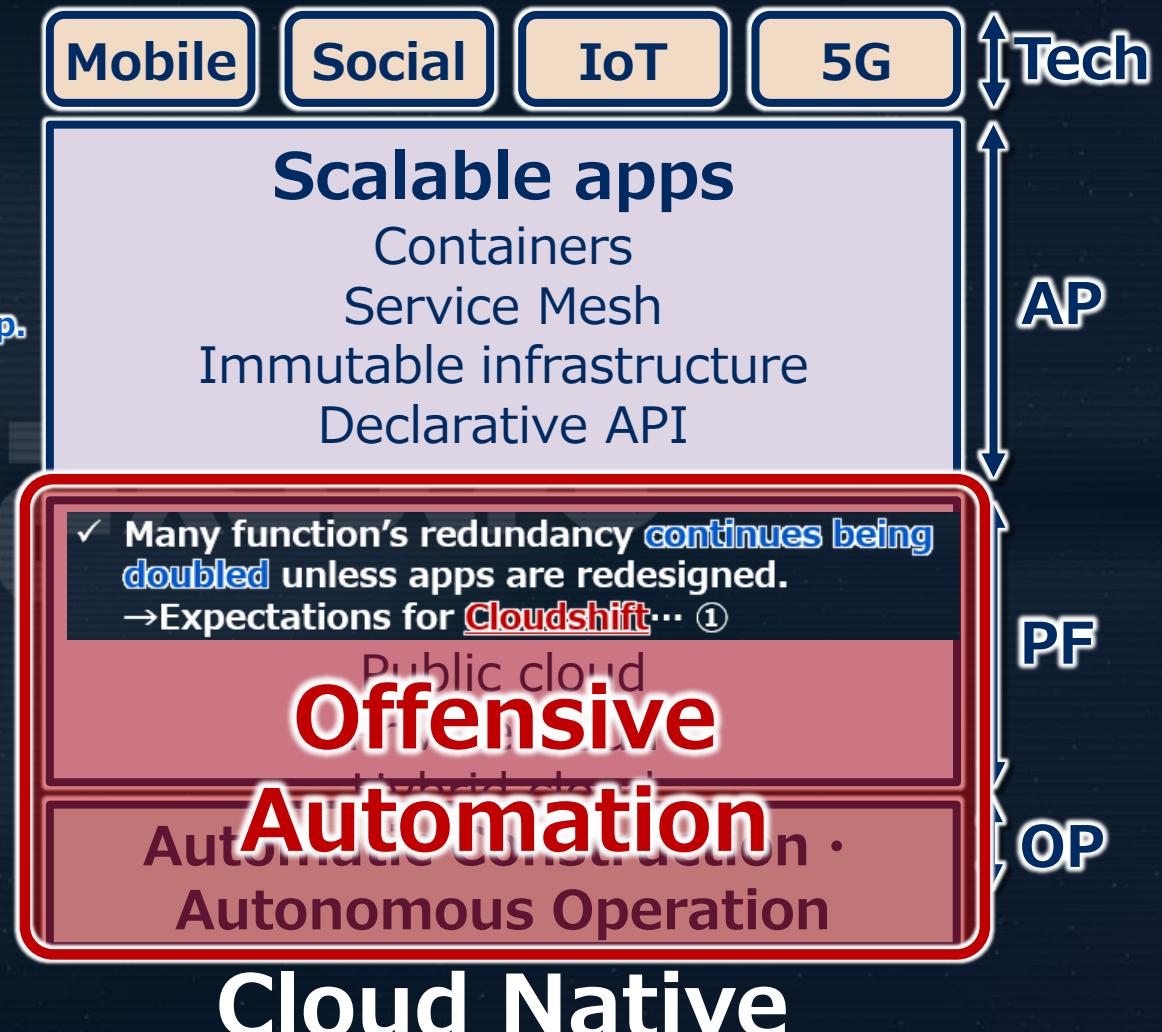
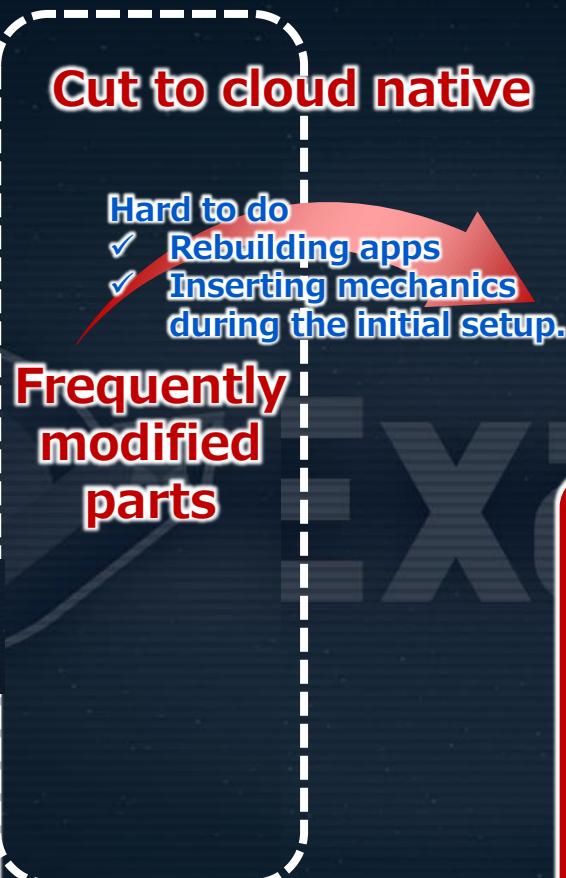
Thanks to virtualization technology...

- ✓ We can now freely select CPU, Memory and I/O
- ✓ It is now possible to create **Simple Dynamic systems** easily
- ✓ The size of the systems became more compact and we could get rid of end-of-life hardware.
- ✓ **Not mixing multiple functions** became the “correct” thing to do.

However, The following issues remains unsolved

- ✓ Many function's redundancy **continues being doubled** unless apps are redesigned.
→Expectations for Cloudshift... ①
- ✓ The shape of the system is not changing, **meaning that the amount of operation labor is not changing as much.**
→Expectations for Automated operation ... ②

Introduction – What is “Offensive” and “Defensive” Automation?





How to proceed with “Offensive Automation”

By the way...

How does Cloud-Native **work** ?
What kind of **mechanisms** should we put in
during the initial setup to make DevOps run
smoothly?

What we can expect from Cloud Shift

The 11 Mechanisms (10 techniques + Virtualization)

Raising ROI (Return on Investment)

Improving investment quality to drive sales

Number of functions provided per unit time = increase in value

Providing high speed and scalable functions (Release)

Improving OPEX efficiency

Lower OPEX (Operation expenses) to invest

Reduction of operational labor costs

Reduction of operational maintenance costs

End goal

Leveraging IT Resources and increase profits.

Steps to realizing Cloud Native

CI(Continuous Integration)

CD(Continuous Delivery)

Micro service Architecture

Service Mesh

Blue green deployment
(Immutable Infrastructure)

Declarative API

Auto scaling

Distributed tracing

Metrics monitoring

Centralized management and Automated maintenance tasks

Virtualization
(Free from Hardware life)

How to proceed with “Offensive Automation”

What we can expect from Cloud Shift

The 11 Mechanisms (10 techniques + Virtualization)

Steps to realizing Cloud Native

Raising ROI (Return on Investment)

Improving investment quality to drive sales

Number of functions provided per unit time = increase in value

Providing high speed and scalable functions (Release)

CI(Continuous Integration)

CD(Continuous Delivery)

Micro service Architecture

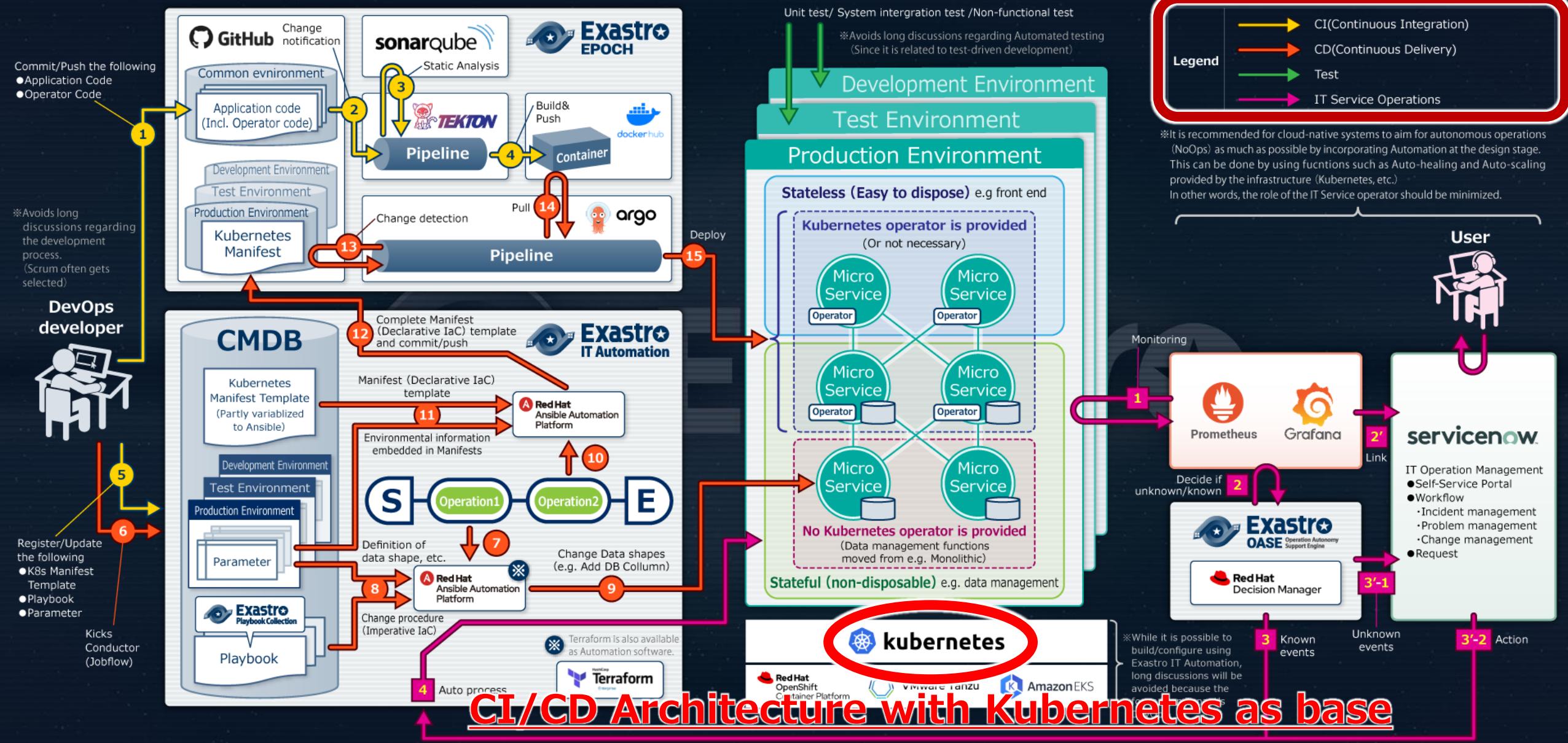
Service Mesh

Blue green deployment
(Immutable Infrastructure)

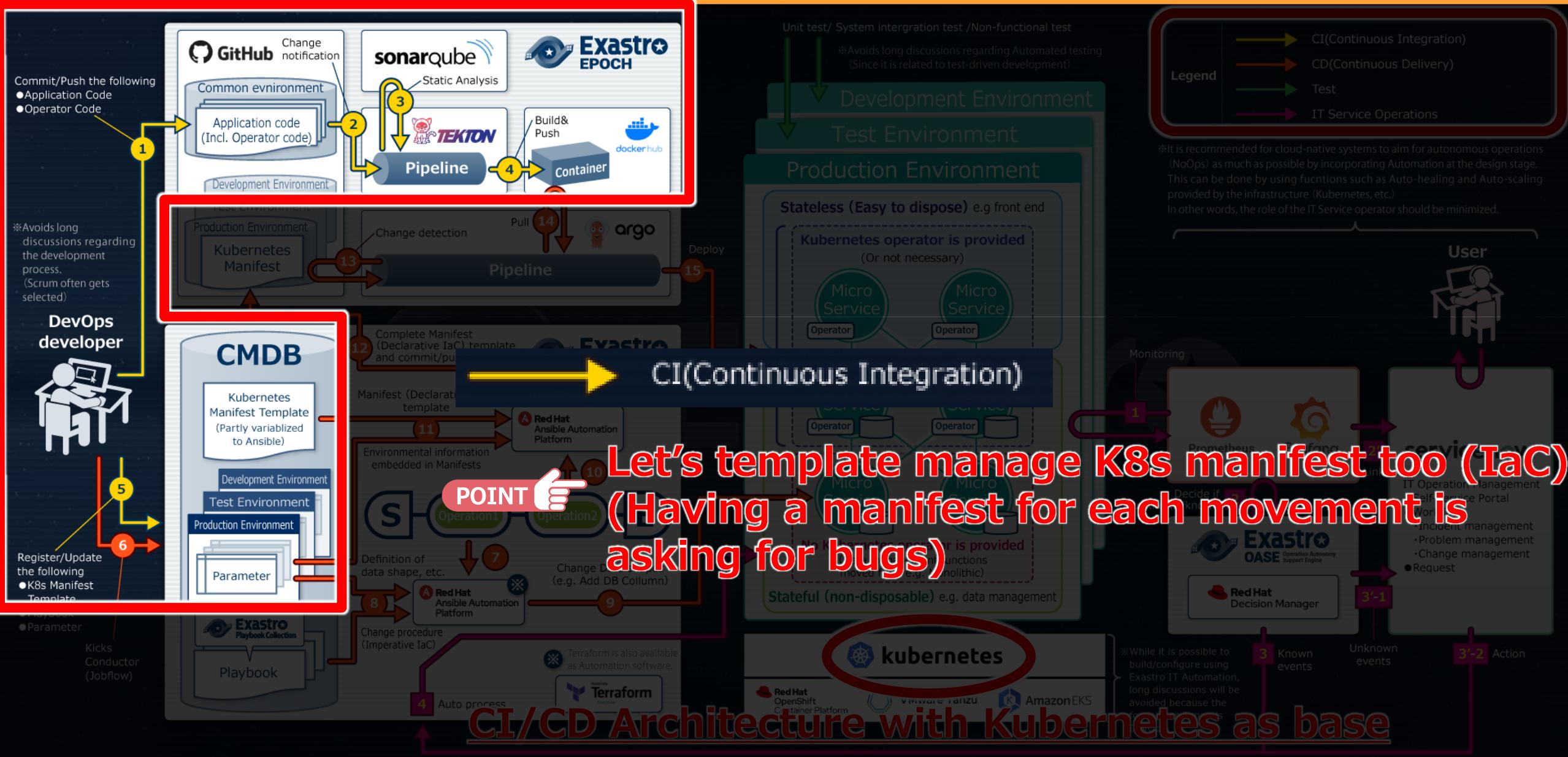
Declarative API

These 6 steps directly relates to the biggest theme in Cloud nativeness Increasing ROI

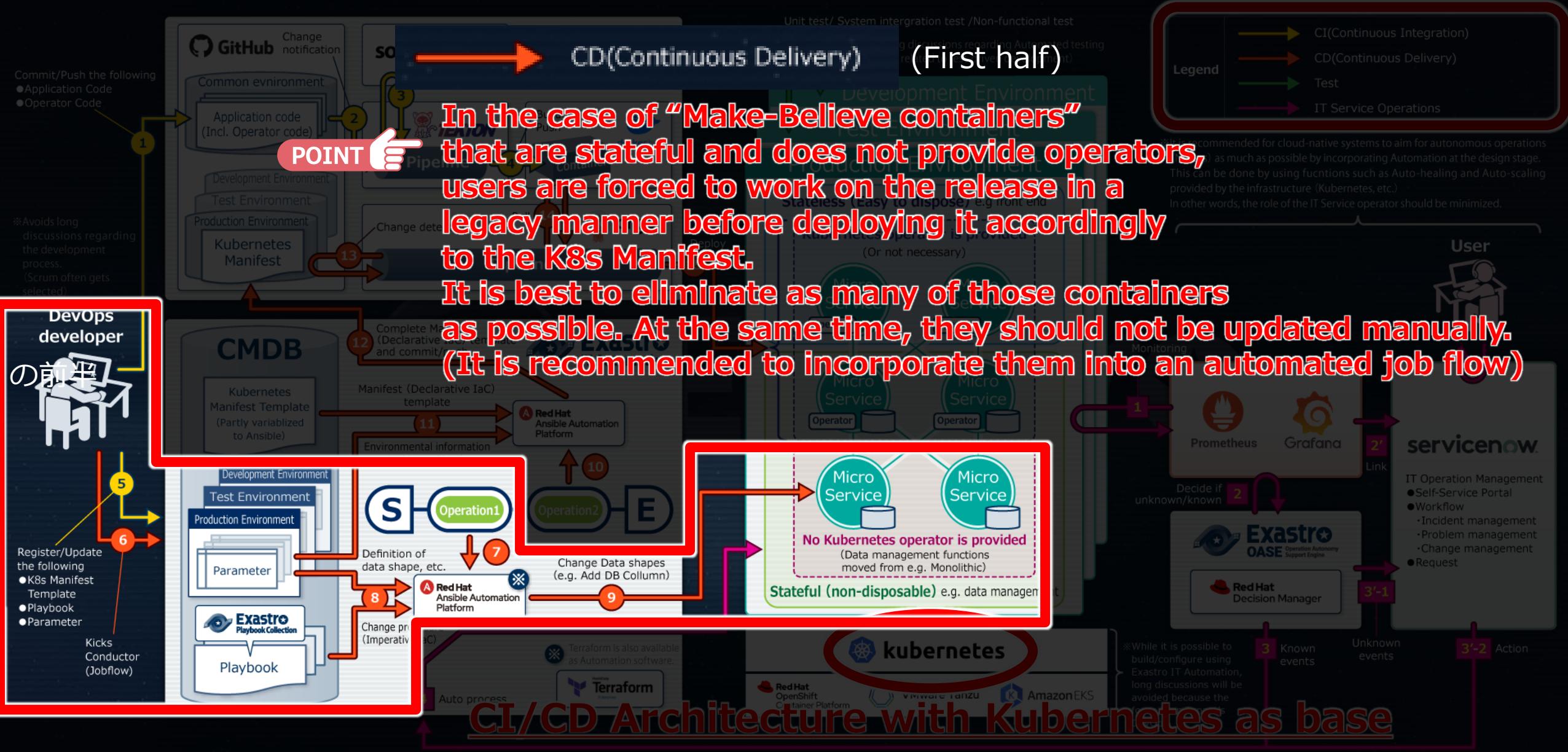
How to proceed with “Offensive Automation”



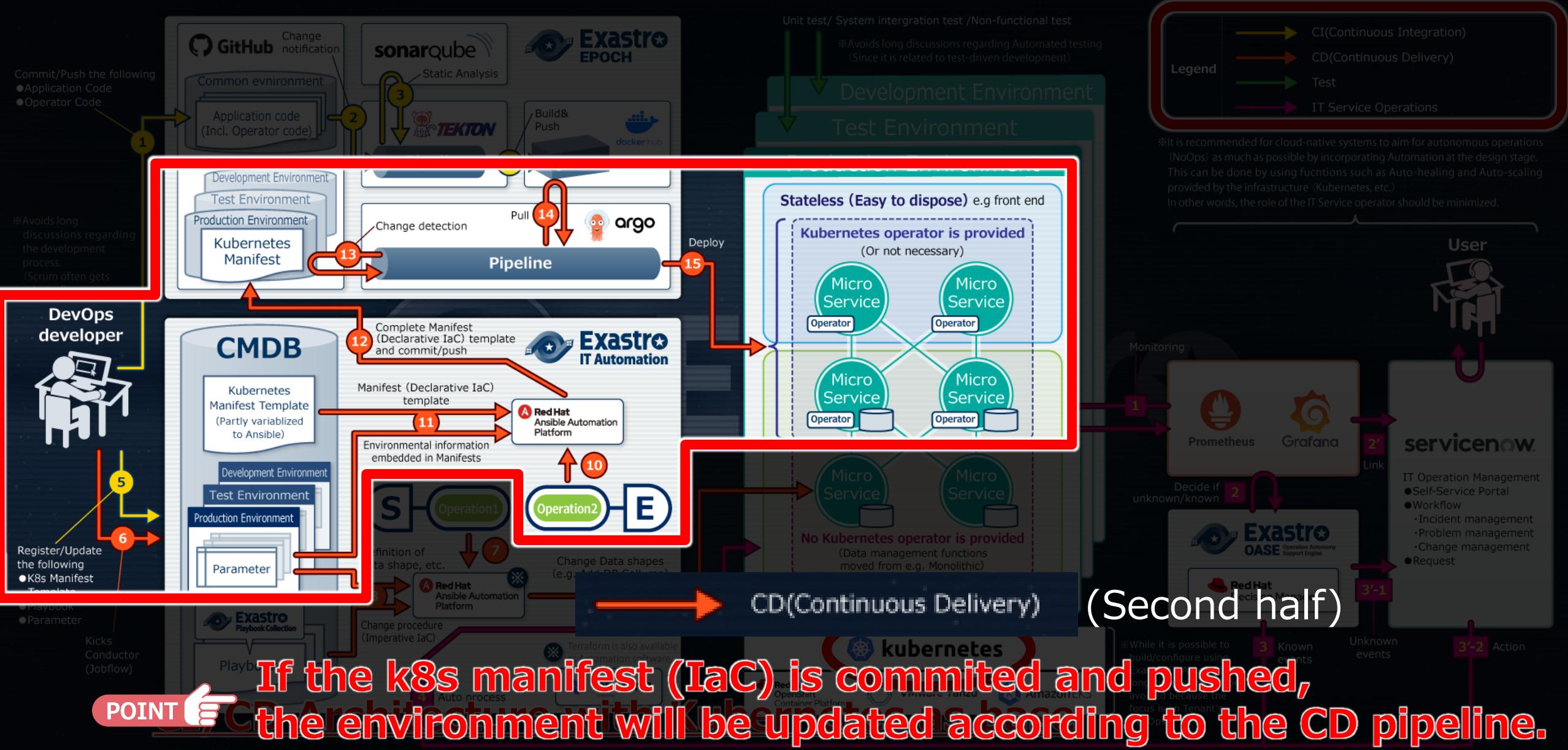
How to proceed with “Offensive Automation”



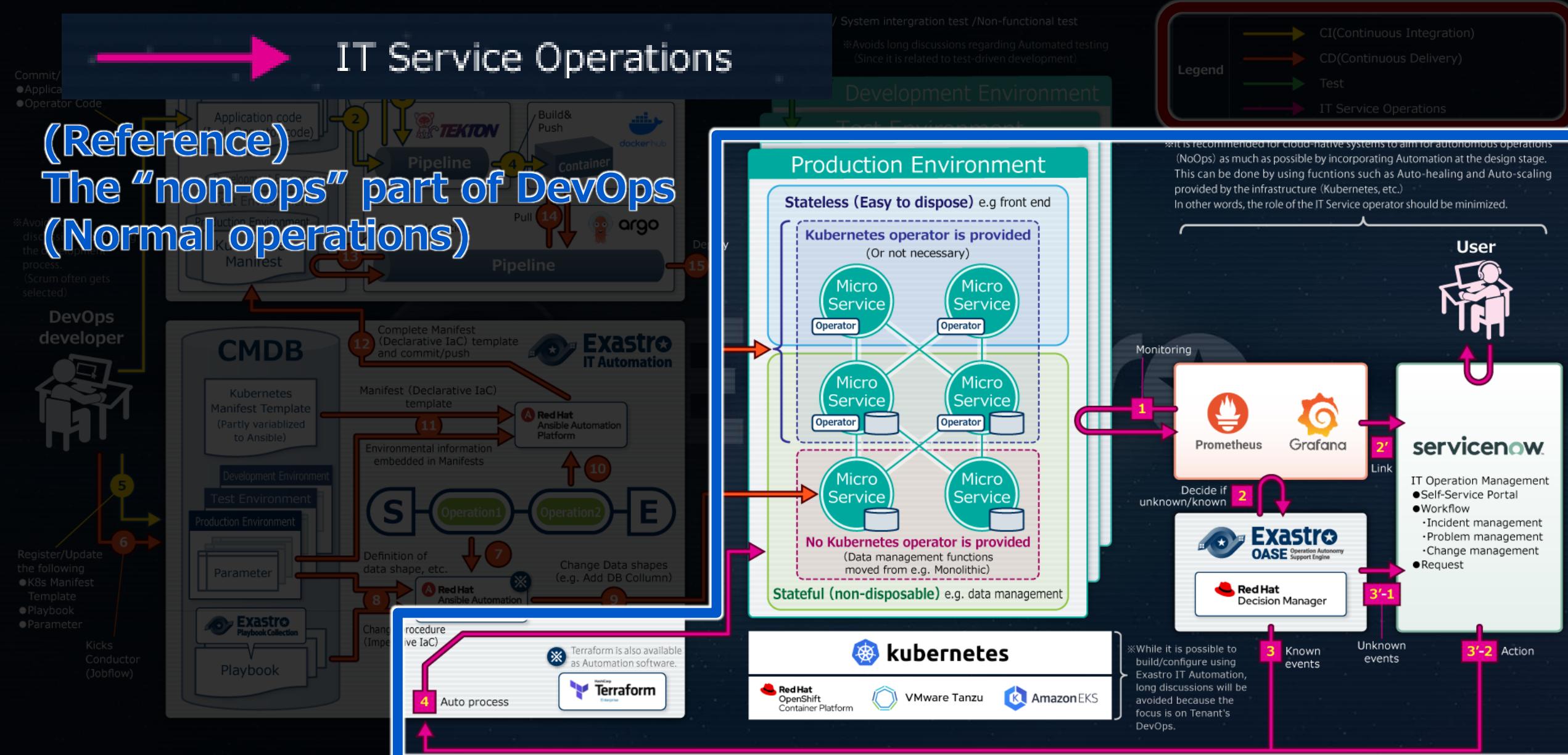
How to proceed with “Offensive Automation”



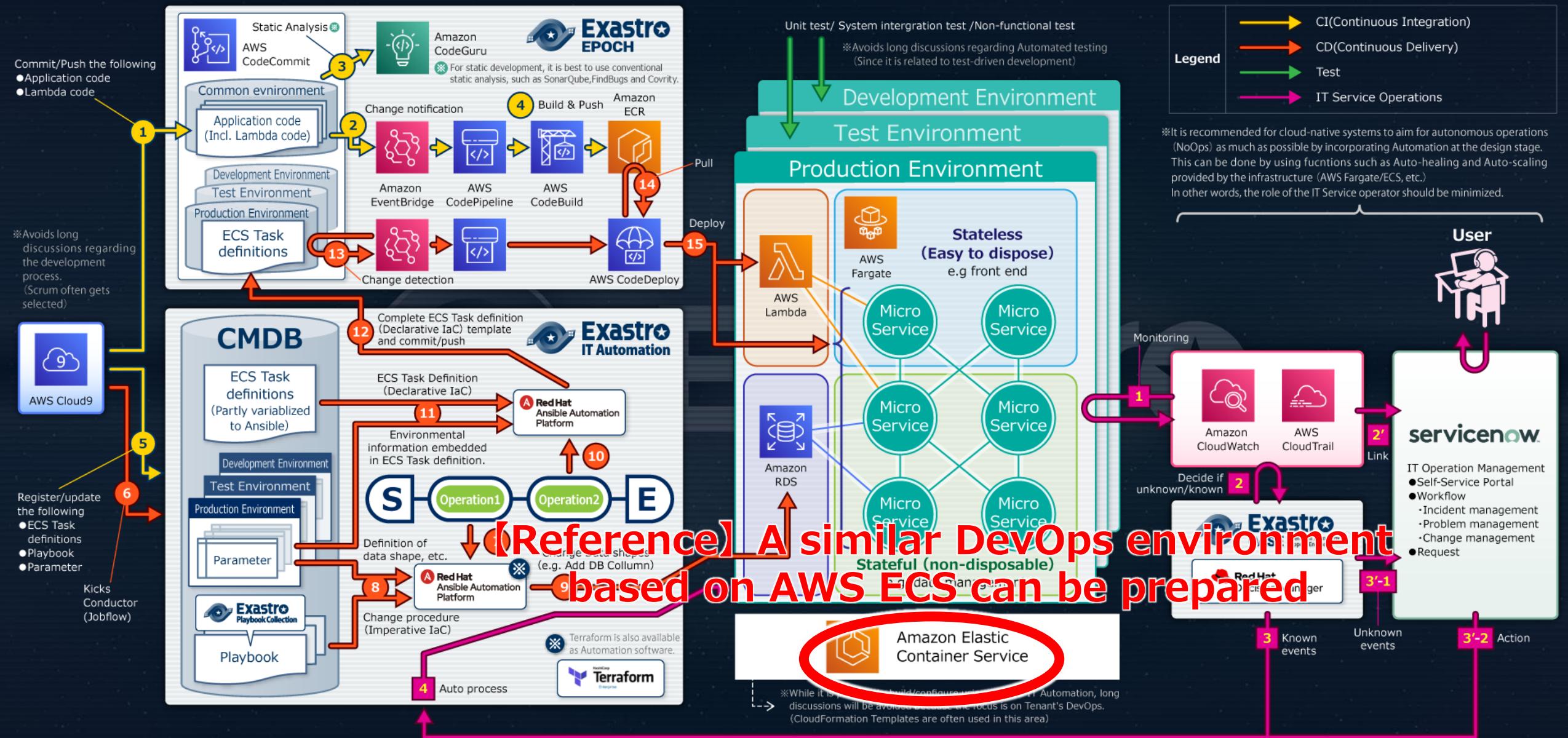
How to proceed with “Offensive Automation”



How to proceed with “Offensive Automation”



How to proceed with “Offensive Automation”





How to proceed with “Defensive Automation”

We summarized the voices of IT engineers that works with SI for Monolithic systems.



Design



Preparation



Execution

- Delays and errors occurs when communicating between teams.
- Double managing data and proprietary wording leads to errors in the design
- Multiple development leads to complications with managing design documents (forms)
- As a result, we are unable to check before and after the settings.

- Work orders between teams are complex. Each time a time chart is created, it gets discarded.
- Every operation's Manual is discarded after its created/reviewed.
- Configurations are embedded in each procedure, and the number of patterns increases each time a new model/os is added (barrier to multi-vendor support)

- Since the operations are done manually, the production time is inconsistent.
⇒ People often have to wait before they can continue.
- Since most of the operations are done manually, human error is inevitable.

We summarized the voices of IT engineers that works with SI for Monolithic systems.



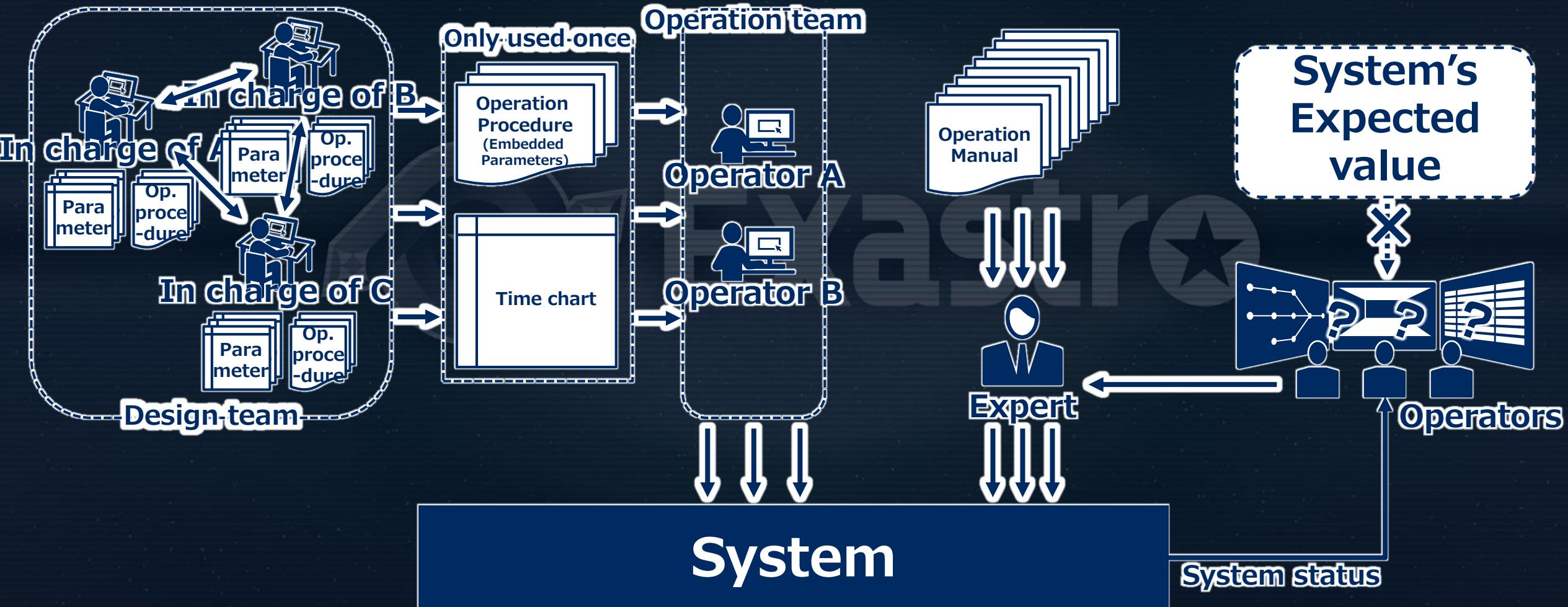
- There is lack of understanding of which parameters can be changed and which parameters should not be changed during operation.
- Current values of system parameter's change history is not managed.
- As a result, even though the parameters have been created, they are not being used in operations.

- Systems are becoming more complex and the workload is only increasing.
- Whenever something happens, we have to carefully execute one task at a time in a multi-person team while reading through a large number of manuals written in excel.
- As a result, system downtime is longer and service gets affected.

- Work isn't progressing due to the absence of experts.
- When the experts disappears, the know-how disappears with them.
- It is difficult to distinguish between known and unknown events, making us rely on the experience of experts.
- As a result, experts cant be transferred.

How to proceed with “Defensive Automation”

People involved spend a lot of time trying to make sense out of the scattered information

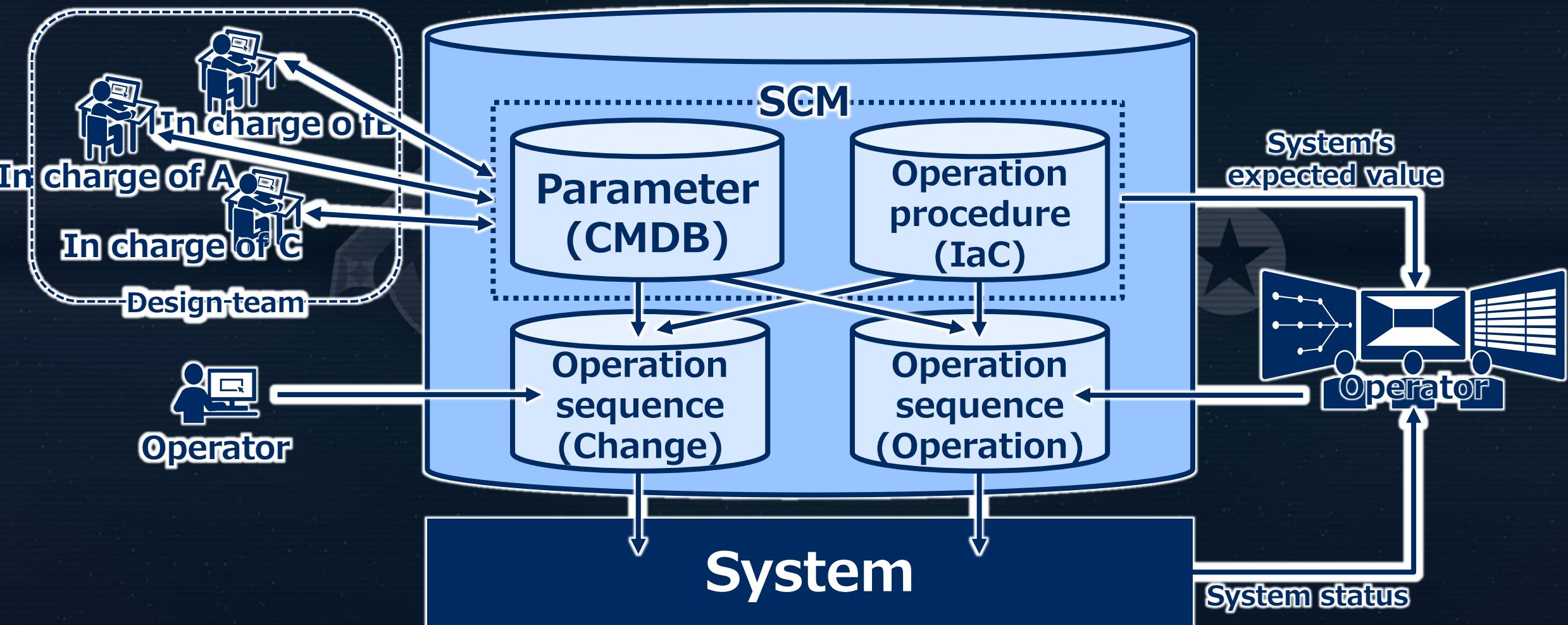


People involved **spend a lot of time trying to make sense out of the scattered information**

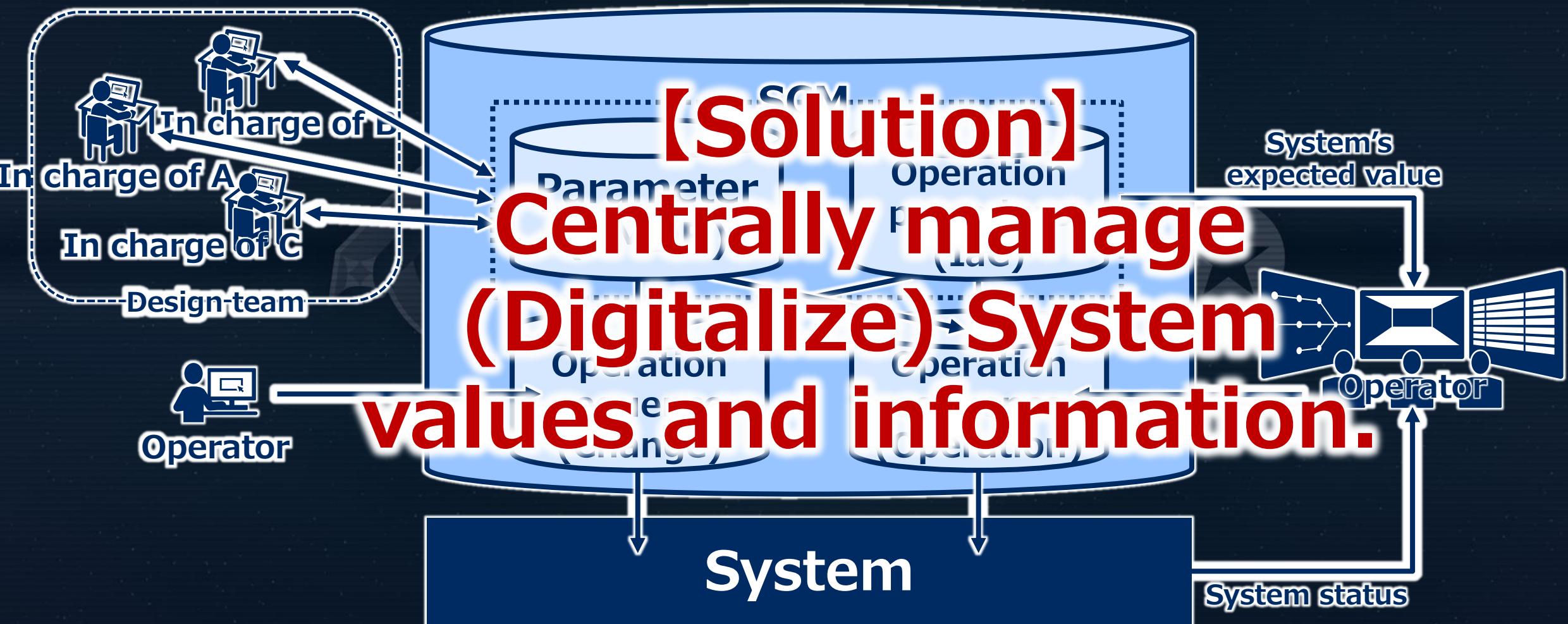


How to proceed with “Defensive Automation”

If we only could digitize and centralize the system information

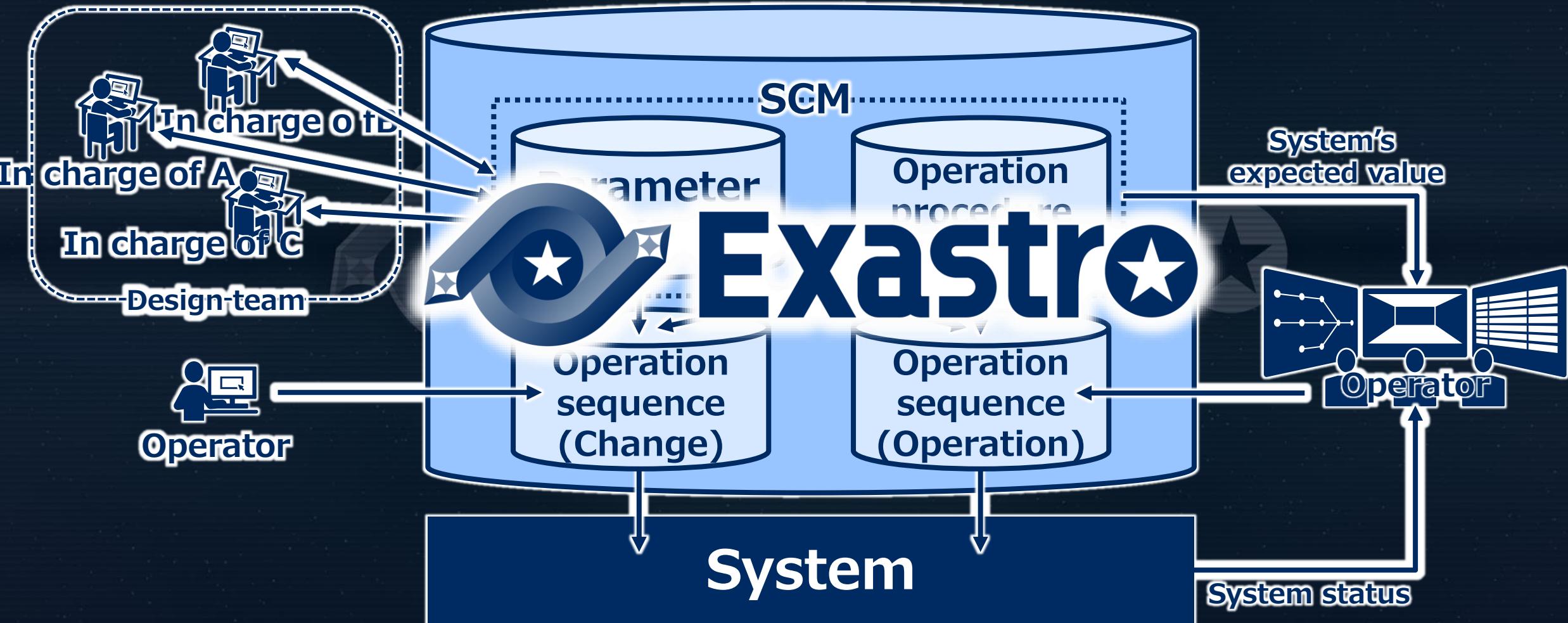


If we only could digitize and centralize the system information



How to proceed with “Defensive Automation”

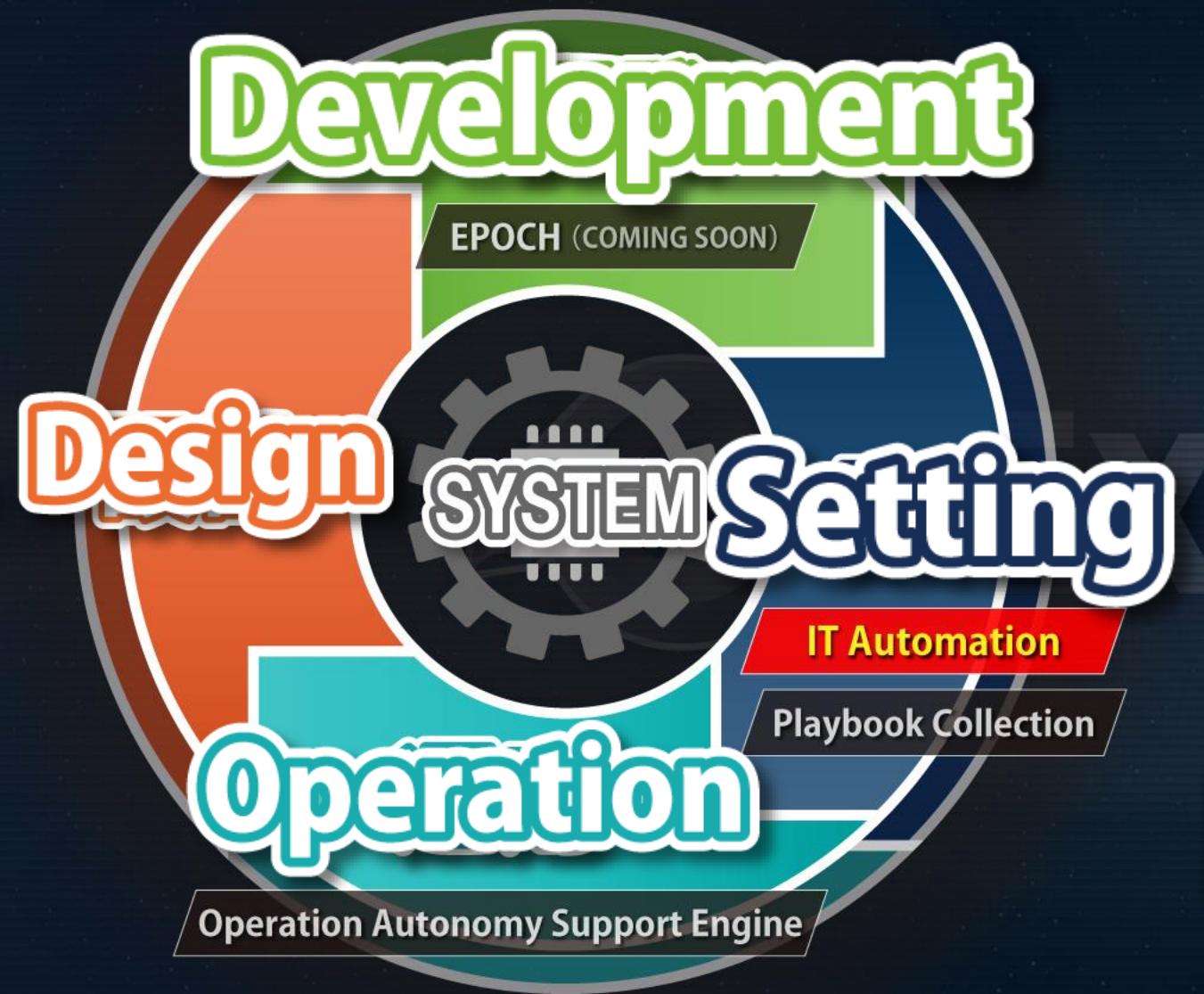
Exastro helps to digitize and centralize system information



Exastro Suite



What is Exastro Suite?



Exastro is an open source software suite

設計
design

開発
development

設定
setting

運用
operation

を

for digitizing, automating and labor saving
the system life cycle
(design, development, setting and operation)

Roadmap of the Automation and Autonomy that Exastro Suite is aiming for

Roadmap of the Automation and Autonomy that Exastro Suite is aiming for



More information can be found on the Community site!



Exastro

 Search

Exastro

<https://exastro-suite.github.io/docs/index.html>

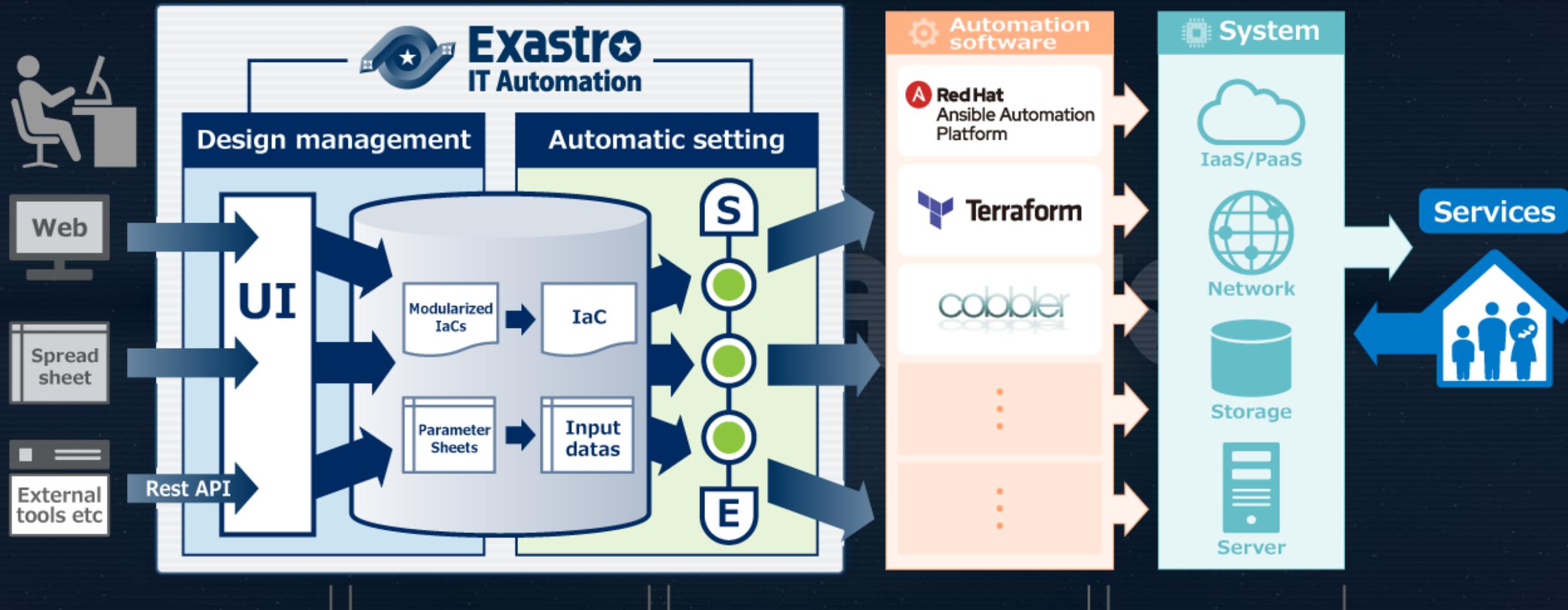


Exastro IT Automation



Design phase

Setting phase



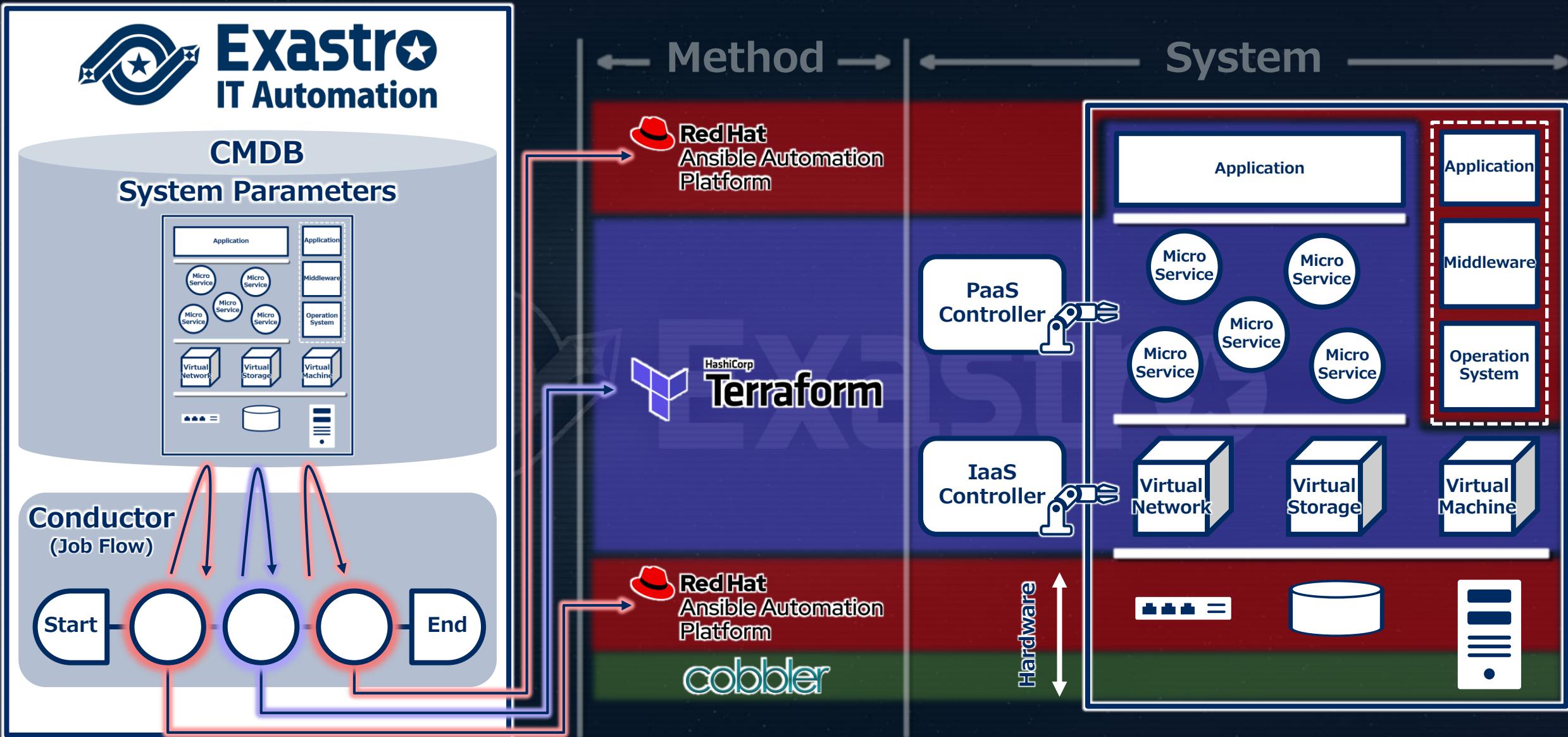
Multiple user interfaces
(Web, Spreadsheet, Rest API)

Central management

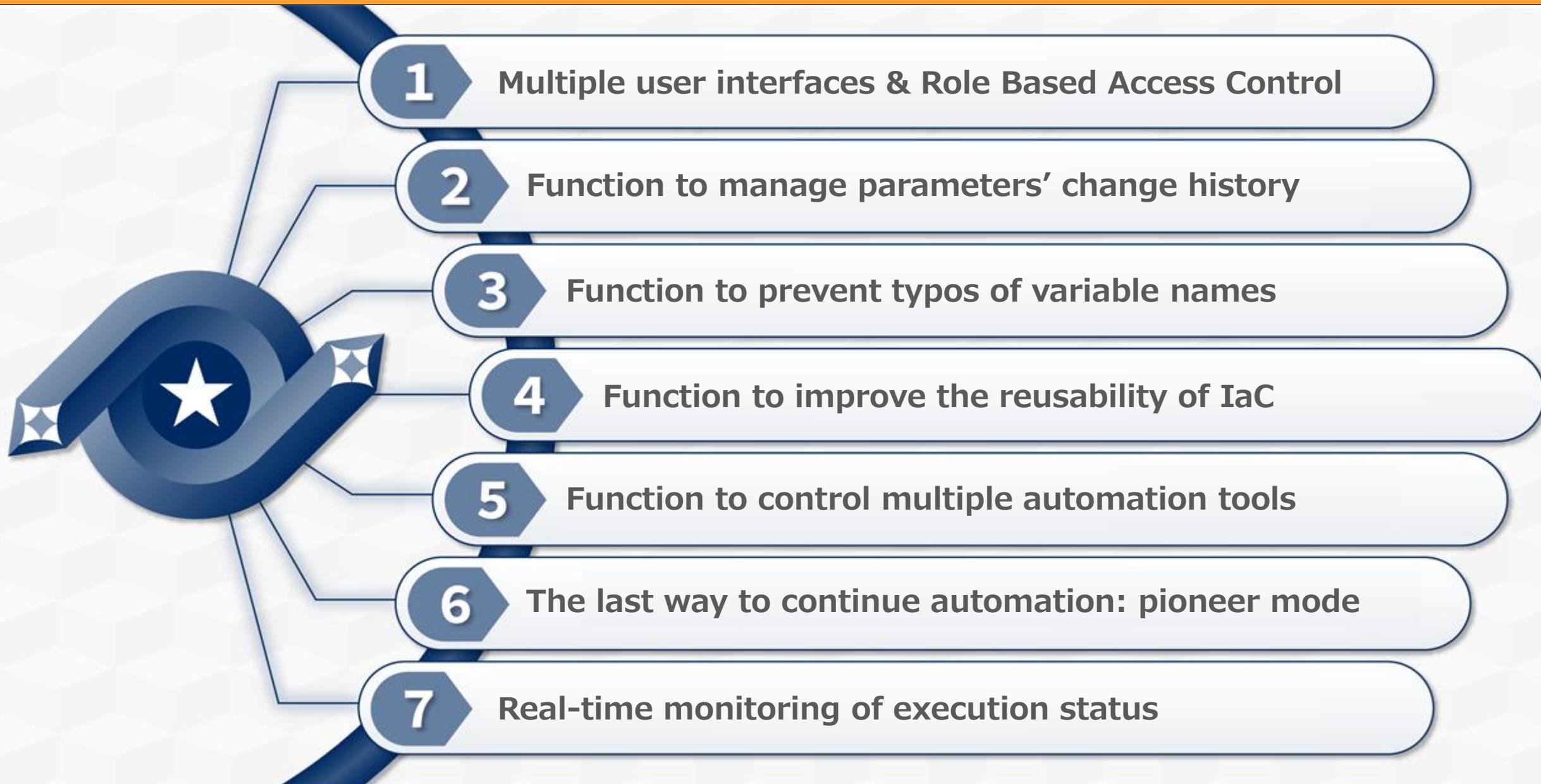
Multiple automation softwares

Many target device types
(Server/Storage/Network/IaaS/PaaS)

Exastro IT Automation : Merits of using system stacks



7 Features of Exastro IT Automation

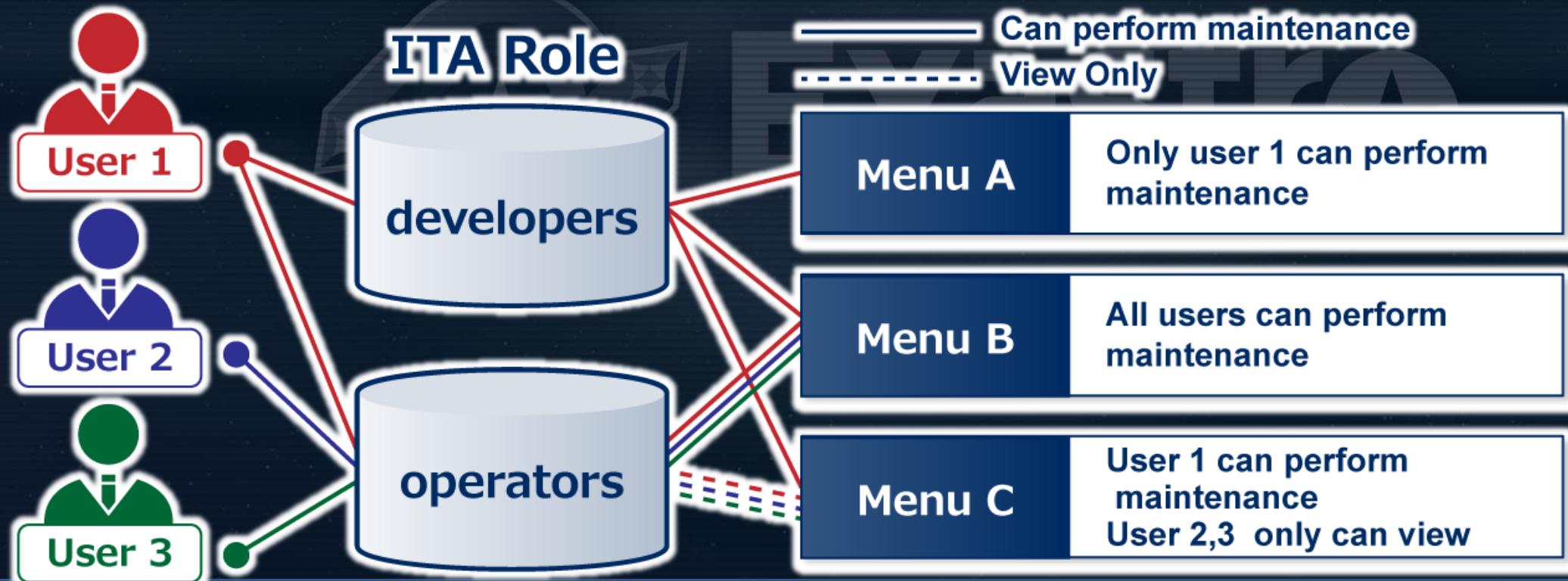


1st feature- Multiple user interfaces and RBAC

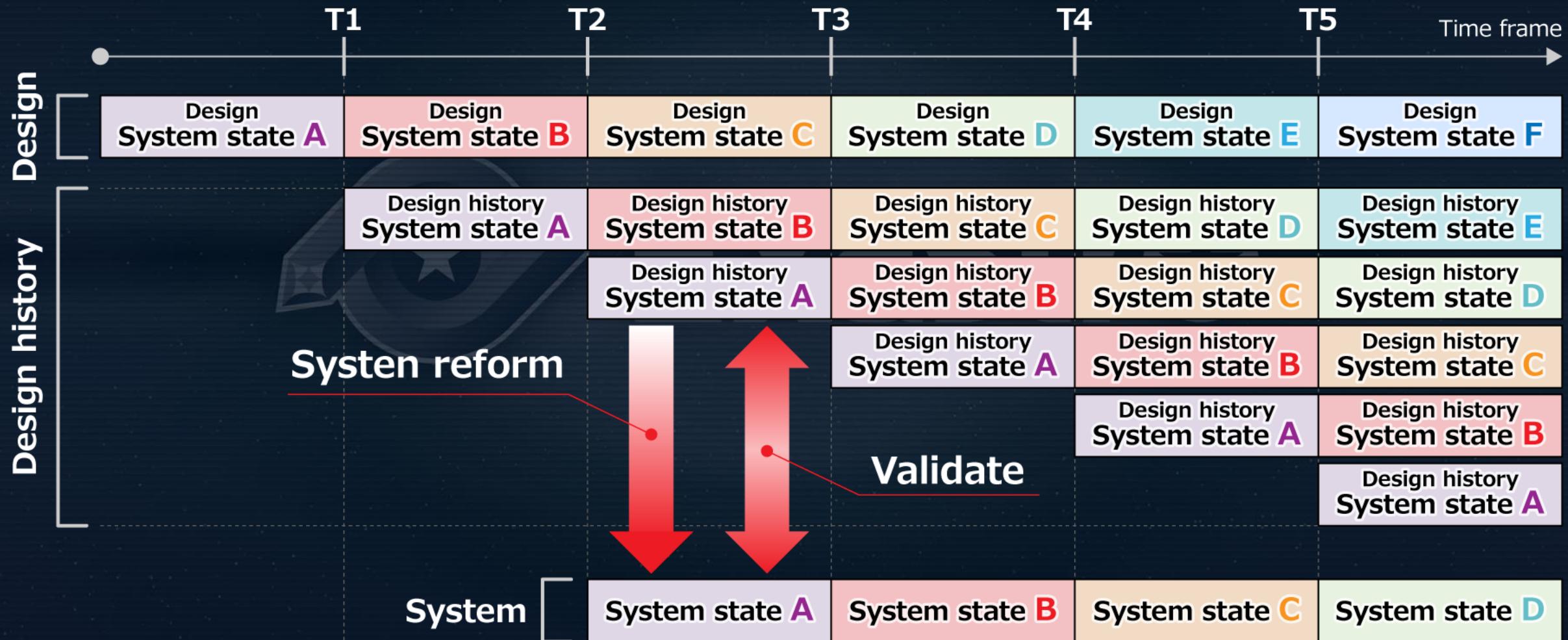
User operations can be executed from **3 types of interfaces**(Web, Excel, RestAPI).

ITA also records operation information from all interfaces, such as **[Who did it • when • what]**.

With Role based access control(RBAC), users can define roles, such as Developer, Operator and Users. They can also **control what each role can do**, such as **"View Only, Update, and Execute."**



Group and manage System parameter information.



2nd Feature – Group and manage Parameters' change history (2/2)

Parameter sheets come standard with **history management function**.
The system uses the extracted info from the design history to update the system

Parameter sheet with history management function provided by Exastro IT Automation

| host | Operation | | Parameters | | | | Design Date |
|-------|-----------|-----------------------|------------|-----|------|-----|-------------|
| | date | Name | P1 | P2 | P3 | ... | |
| hostA | 12/20 | Prepare for Christmas | 1024 | 512 | 2048 | ... | 10/1 |
| hostA | 11/20 | Add hostB | 512 | 256 | 1024 | ... | 8/3 |
| hostA | 9/3 | First release | 256 | 128 | 512 | ... | 7/7 |
| hostB | 12/20 | Prepare for Christmas | 16 | 32 | 64 | ... | 10/1 |
| hostB | 11/20 | Add hostB | 32 | 64 | 128 | ... | 9/2 |

Designers can concentrate on design.

Filter parameters by setting operation date to "11/20".

Expected parameters at "11/20"

| host | Parameters | | | | Design Date |
|-------|------------|-----|------|-----|-------------|
| | P1 | P2 | P3 | ... | |
| hostA | 512 | 256 | 1024 | ... | 8/3 |
| hostB | 32 | 64 | 128 | ... | 9/2 |

Operators can concentrate on operation.

Change

Validate



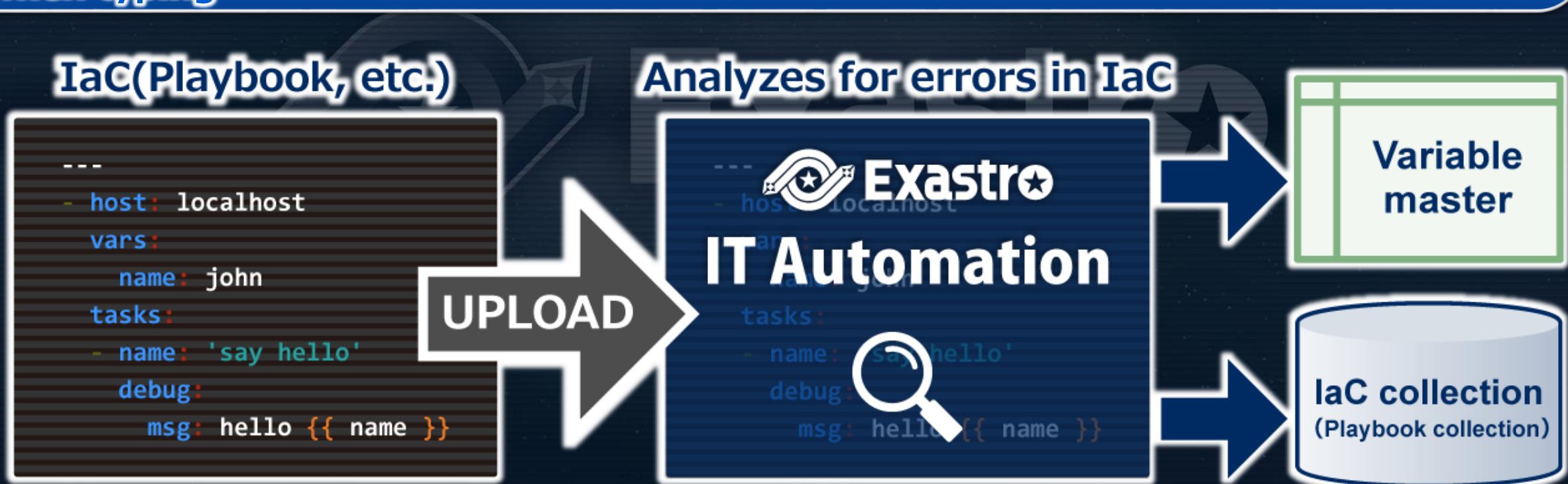
System

3rd feature- Preventing typos in variable names

When an IaC is uploaded in ITA, it will first be analyzed to check whether there are any errors in it.

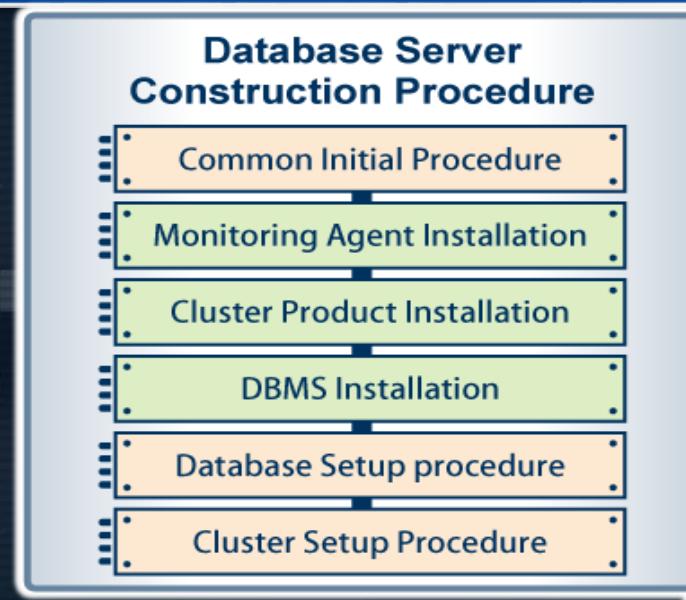
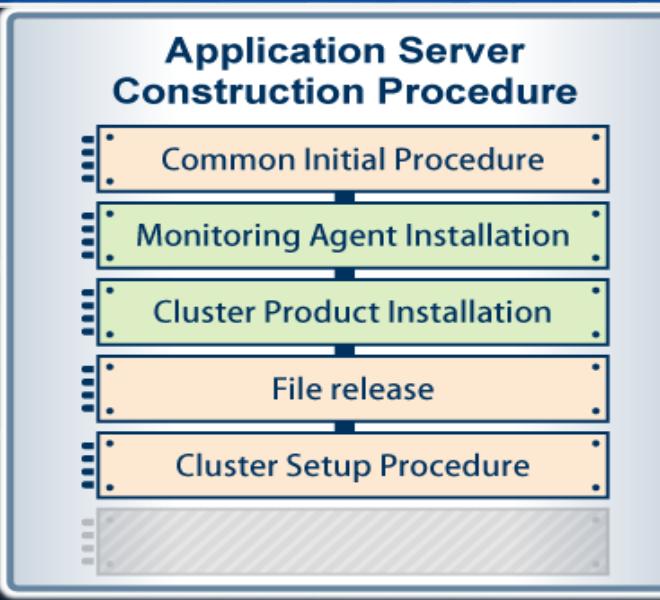
If there are no errors, the variable names from the IaC description are obtained and then managed.

Since variable names are used in selection formulas, this prevents human error when typing

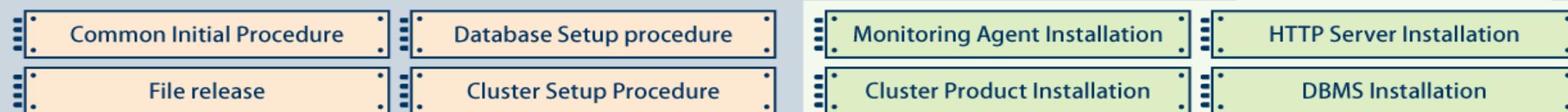


4th feature- Improve IaC reusability.

ITA can reuse IaC (Playbook, etc.) meaning that it **can be used multiple time without being obsolete**. They can then be modularized and assembled when you want to use them



Manage **common** procedures so they can be modularized and reused

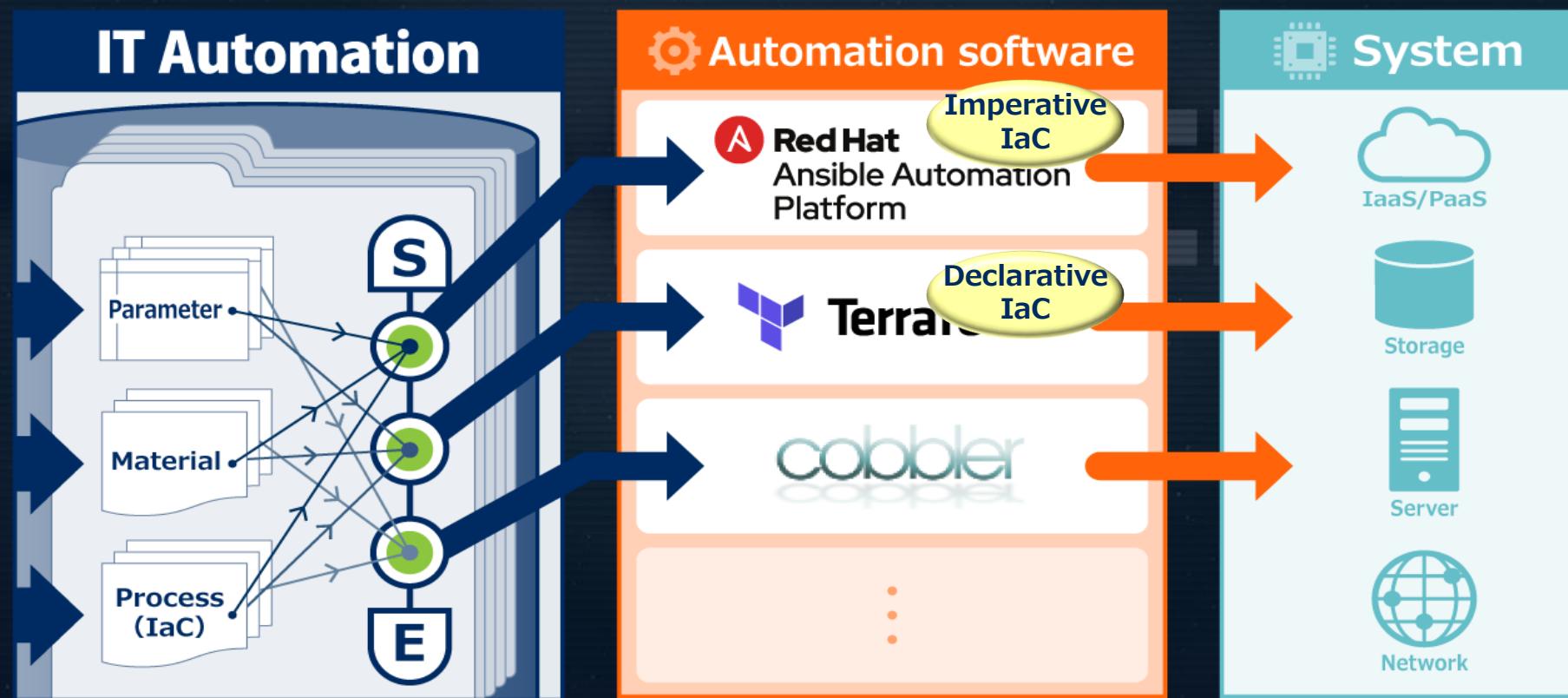


5th feature – Control multiple automation tools

In ITA, users can connect multiple automation software to work in a **single work flow**.

Additionally, input data necessary for operating different automation software will automatically be generated.

E.g) For Ansible, the required playbooks will be gathered and connected. Then it will create host_vars from the parameters for each node.



6th feature – The trump card for automating – Pioneer mode.

If users are unable to automate using any of the Ansible modules, they will only experience half of the benefits that automation can deliver.. Therefore, ITA offers **Pioneer mode**, which can act as a trump card when automation does not work.

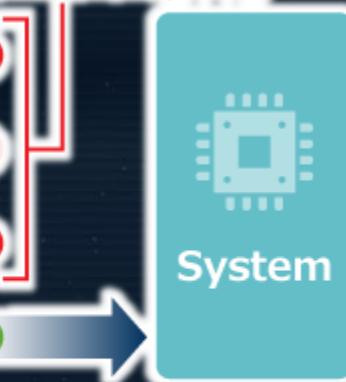
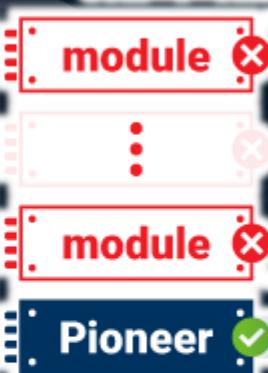
▼ Interactive file designated for Pioneer mode

expect command
+ variable entry
+ Unique IaC that allows conditional branching /repetition



A Red Hat Ansible Automation Platform

If you are not able to do automation, doesn't matter which one you use



ssh / telnet

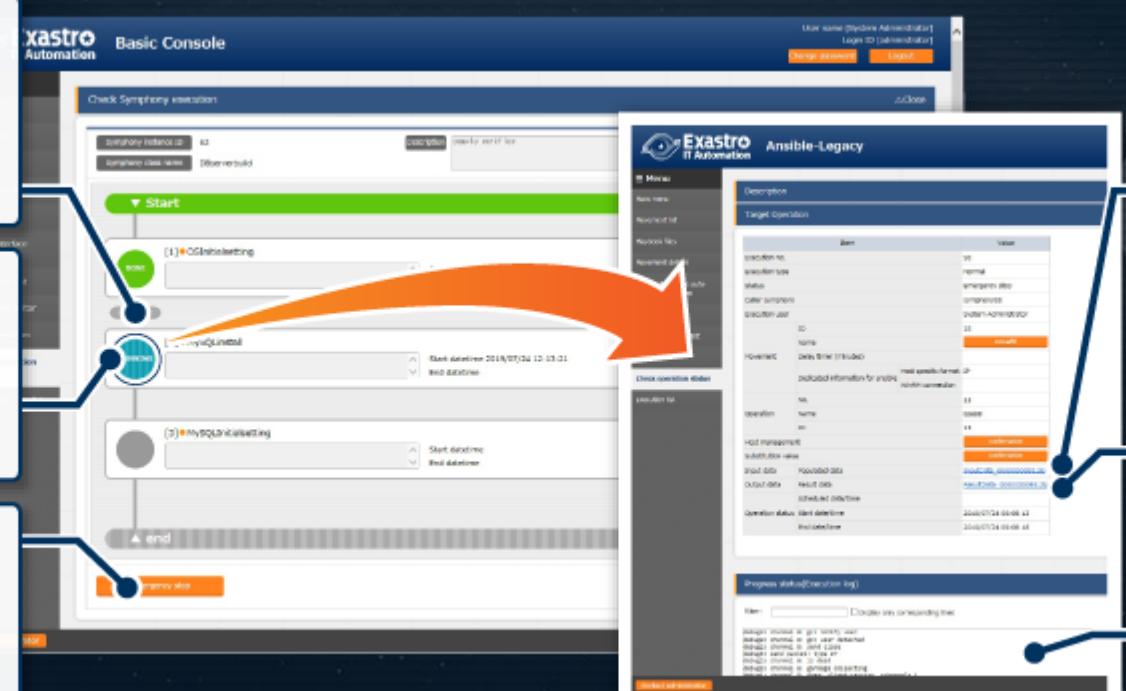
Ansible module designated for ITA
Interact when having ssh or telnet working.

In comparison to manual operations, ITA puts great emphasis on real-time monitoring of execution status
It also manages execution records/results (operation evidence) and allows users to download them.

A "hold point" can be set in the middle of the job flow

Click the execution status to drill down

In case of emergency, it is possible to stop execution by "Emergency stop"

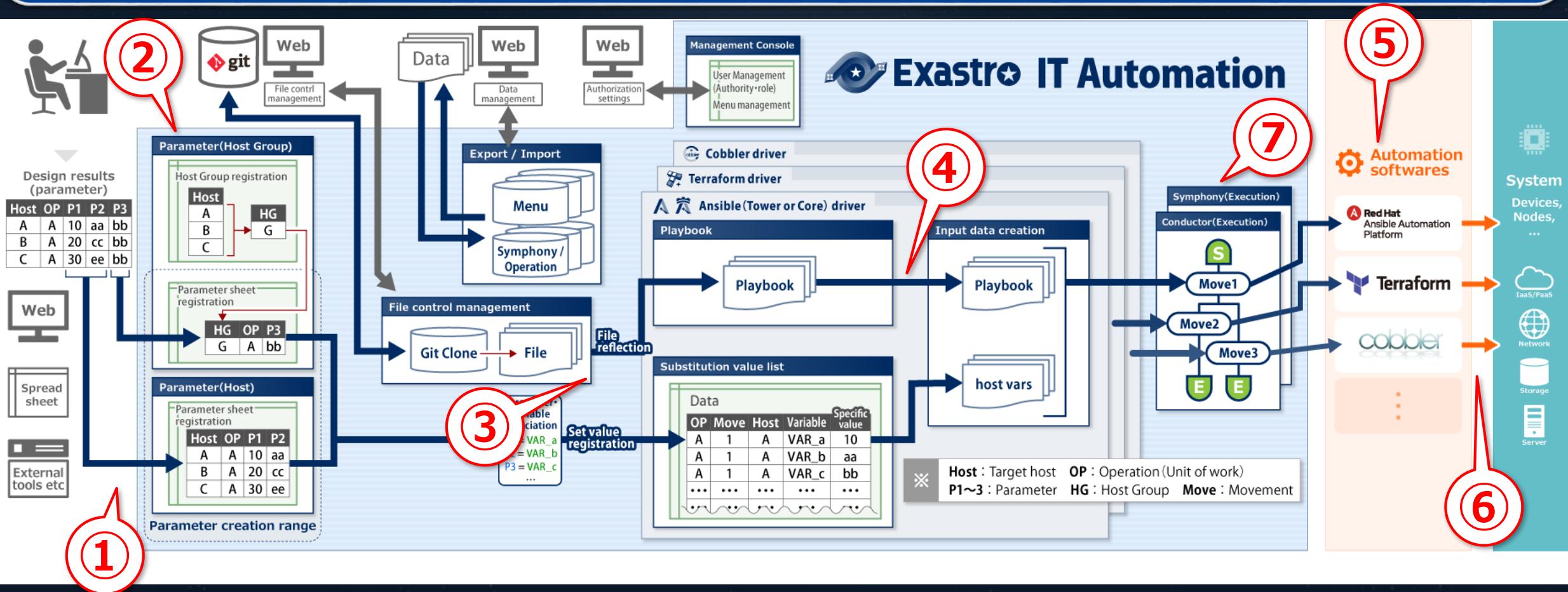


Input data (automatic generated) can be downloaded (zip)

Execution results (work evidence) can be downloaded (zip)

View the execution status of automation software in real time

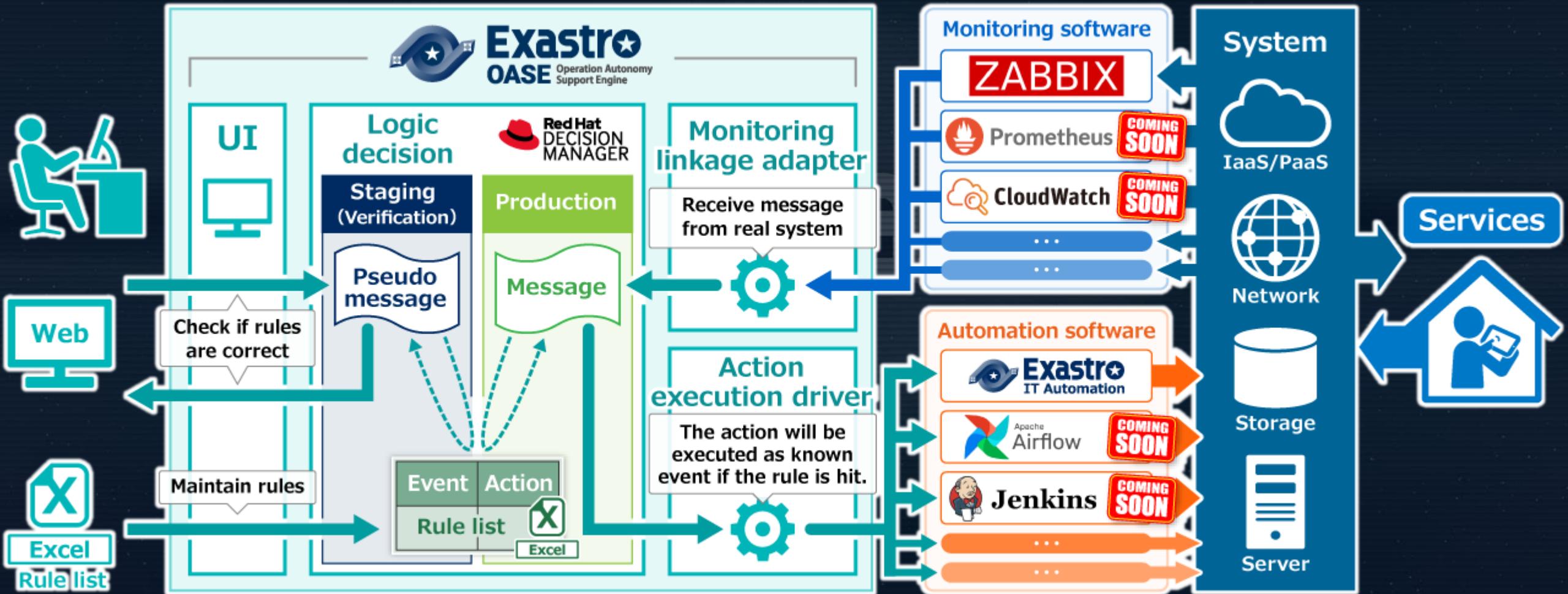
In addition to the 7 features mentioned above, we have devised various other ways to digitally manage system information.

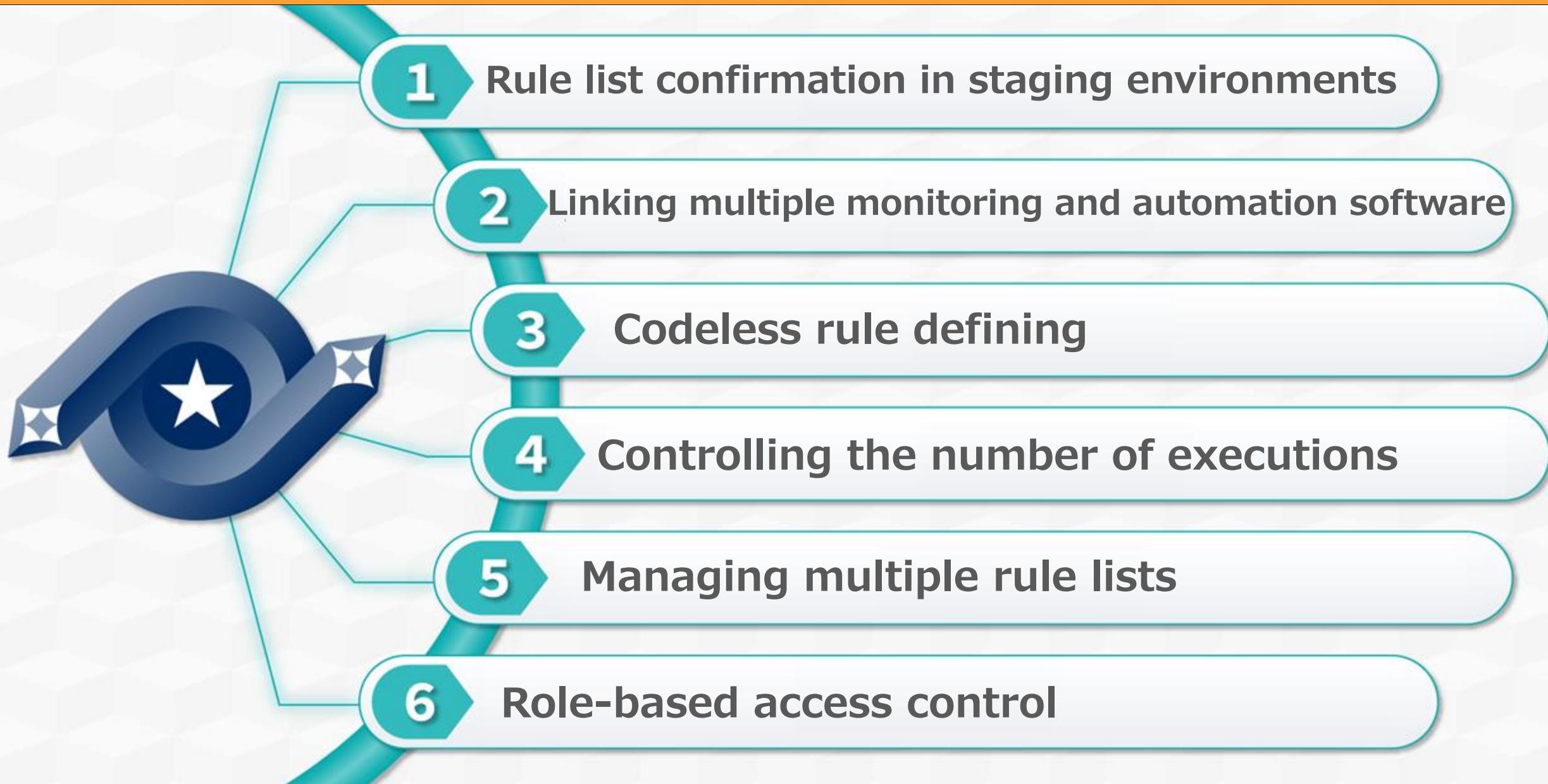


Exastro OASE



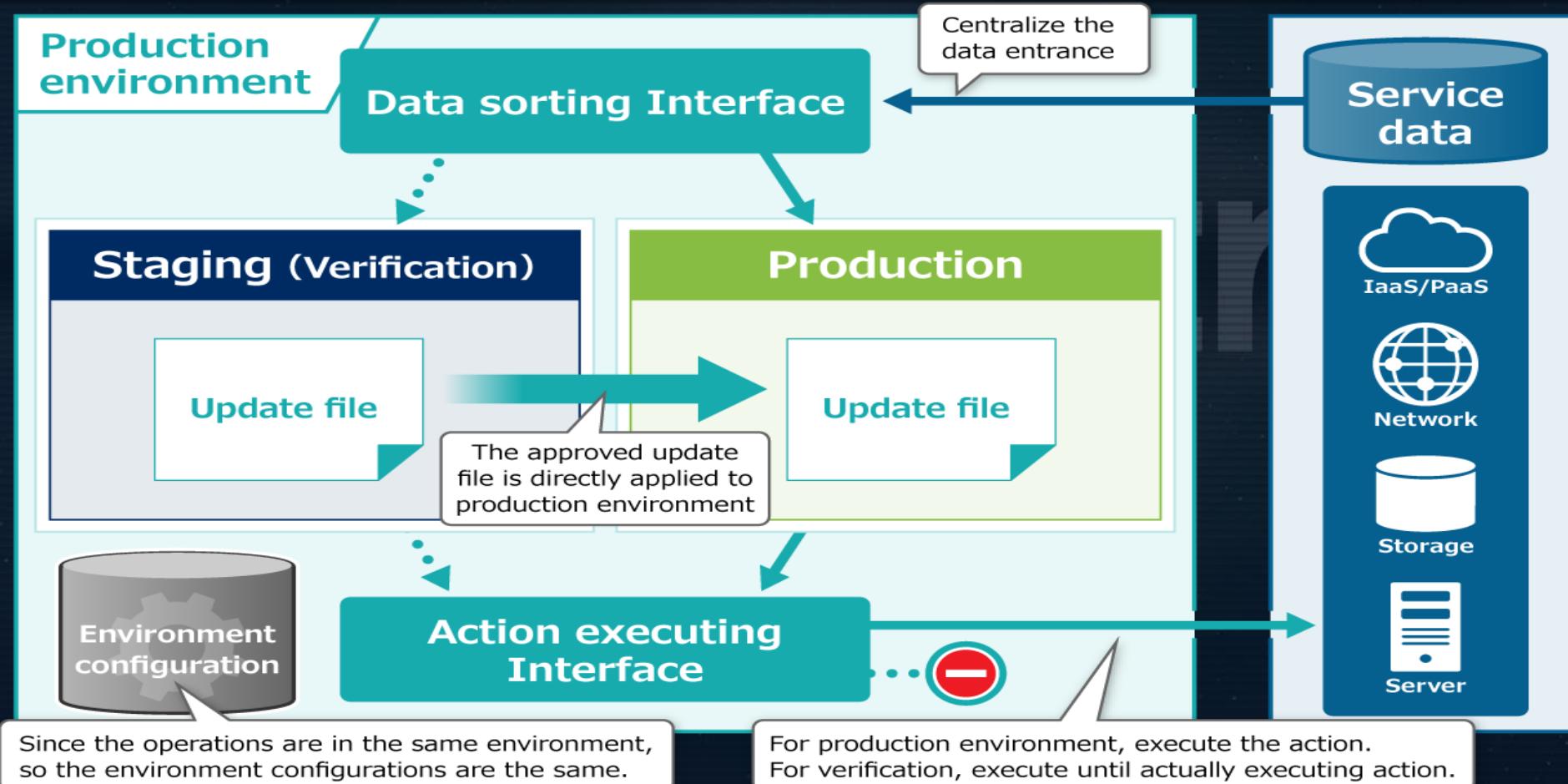
Exastro OASE (Operation Autonomy Support Engine) is a software that supports Autonomization of system operations.





1st feature -Rule list confirmation in staging environments

Staging environments can be used to **check the validity of the execution before applying to rule lists.**



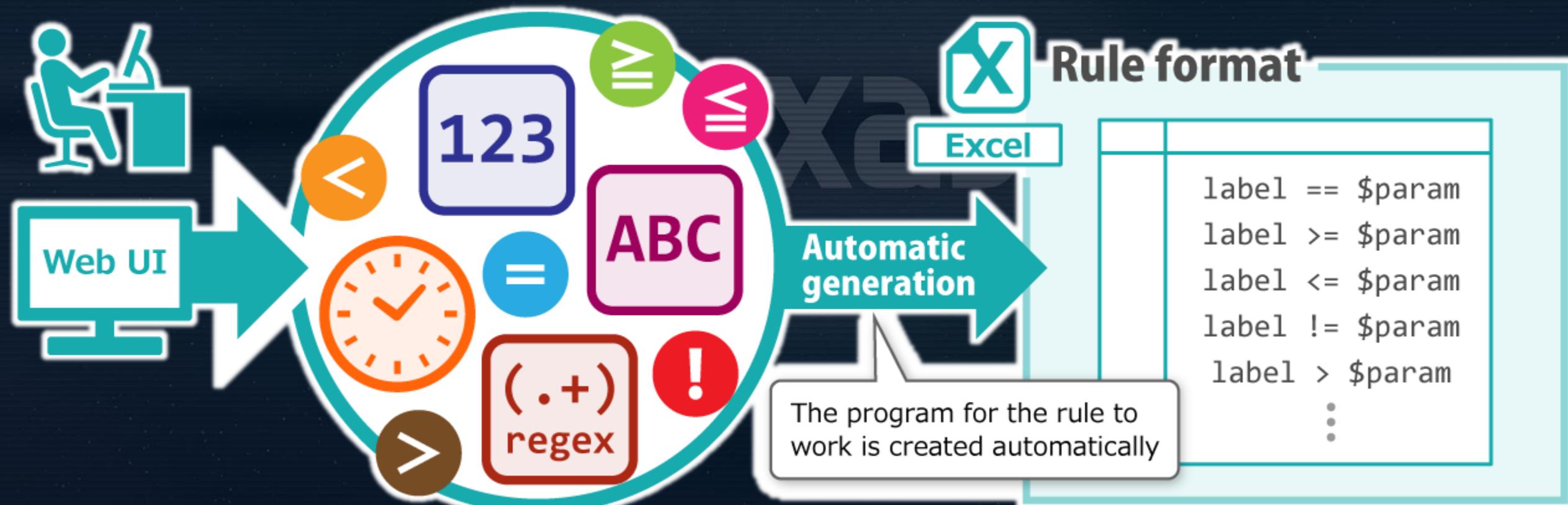
2nd feature – Linking multiple monitoring and automation software

OASE is equipped with drivers and RestAPI for linking with a variety of tools, allowing for linkage with multiple softwares.



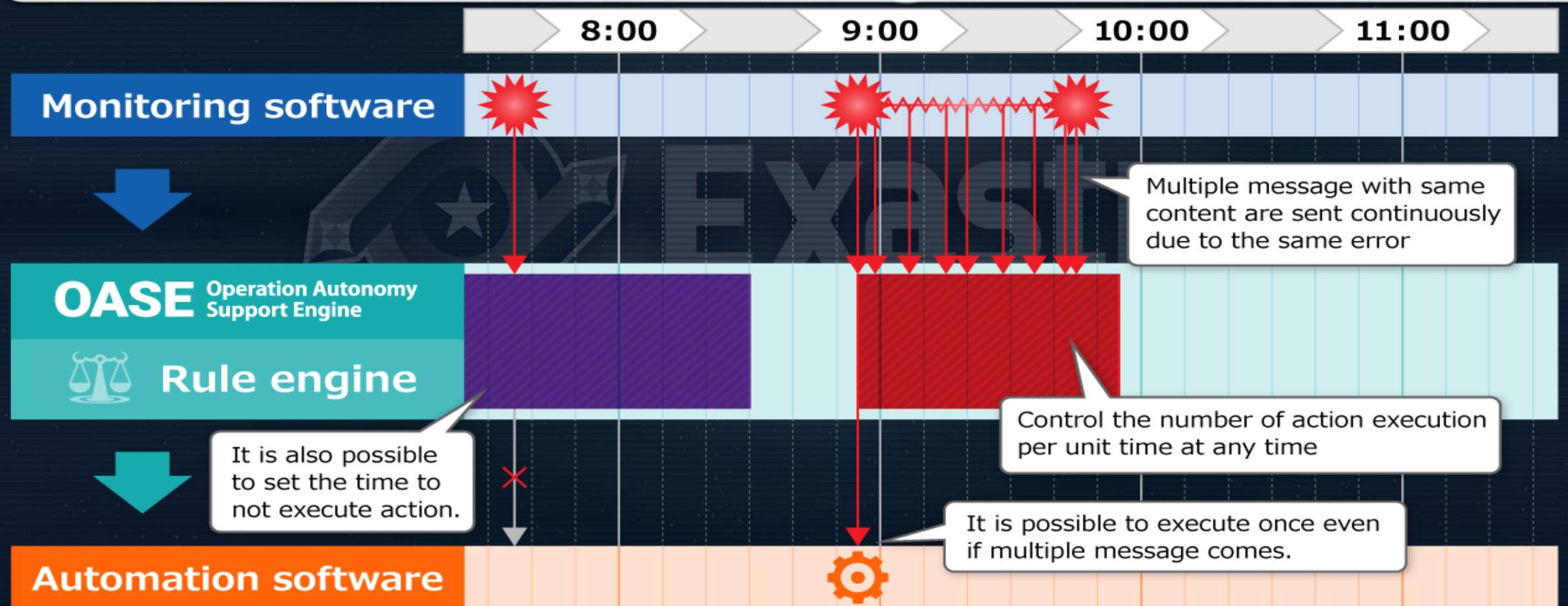
3rd feature – Codeless rule defining

OASE uses **Excel** for ruling format. It is possible to create an excel file with an automatically written program from the Web UI. In other words, **define rule without any coding.**

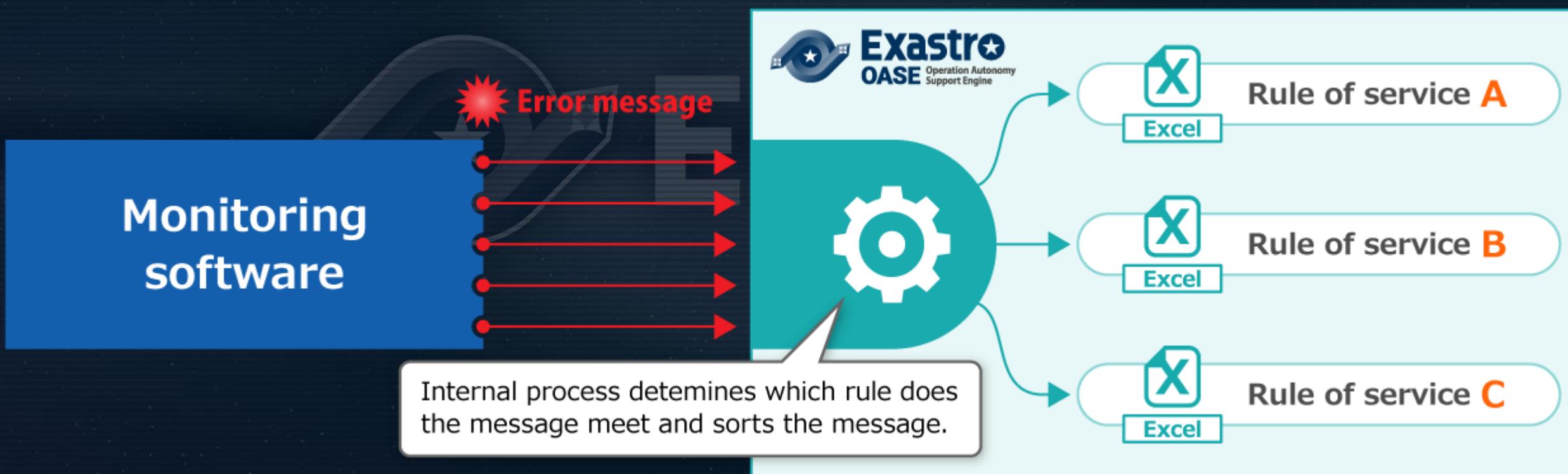


4th feature – Controlling the number of executed actions

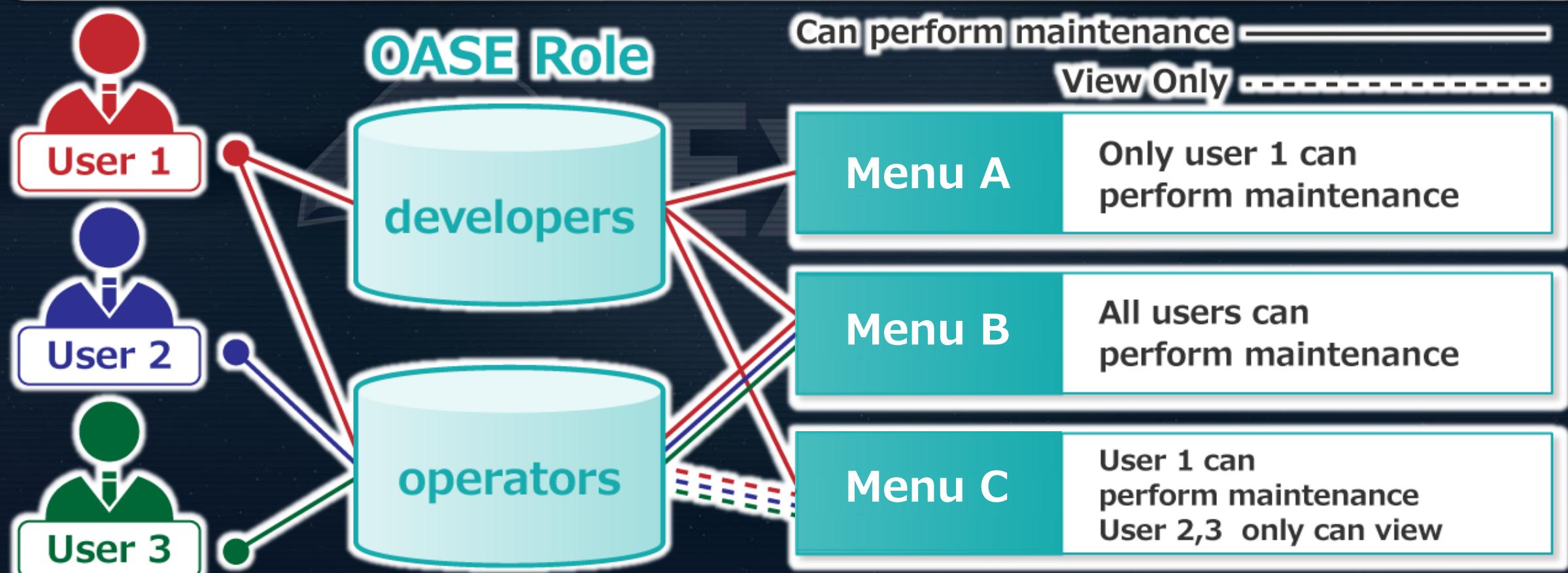
Even if the user is receiving multiple messages from the monitoring tool, it is possible to set the number of executions and the execution interval according to the user's need.



Rules can be divided into units, such as “by project” or “by system” and manage them. It is also possible to distribute messages according to the divided rules.



With Role-based Access control (RBAC) users can assign and manage roles (Reference/Update/Execute) for each group

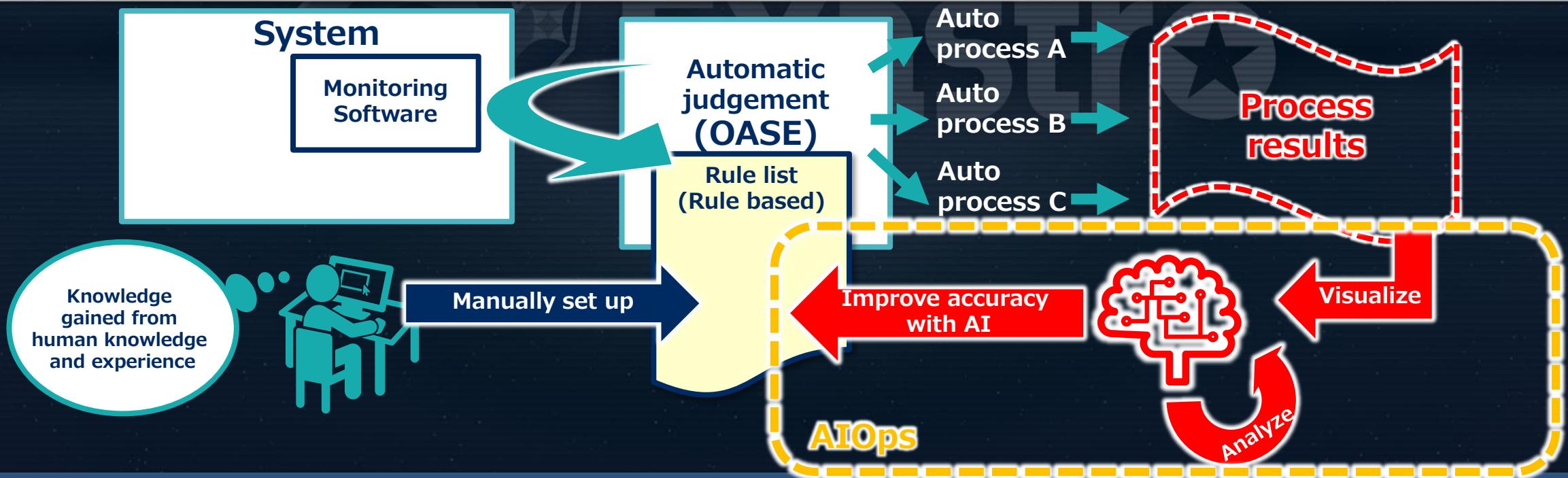


Approach to AIOps

Rule-base is effective for early automation/saving labor of frequently occurring events.

On the other hand, we can expect AIOps to further automate and save labor beyond the early stages. However, as it requires training data for the AI to read, it takes time to start up.

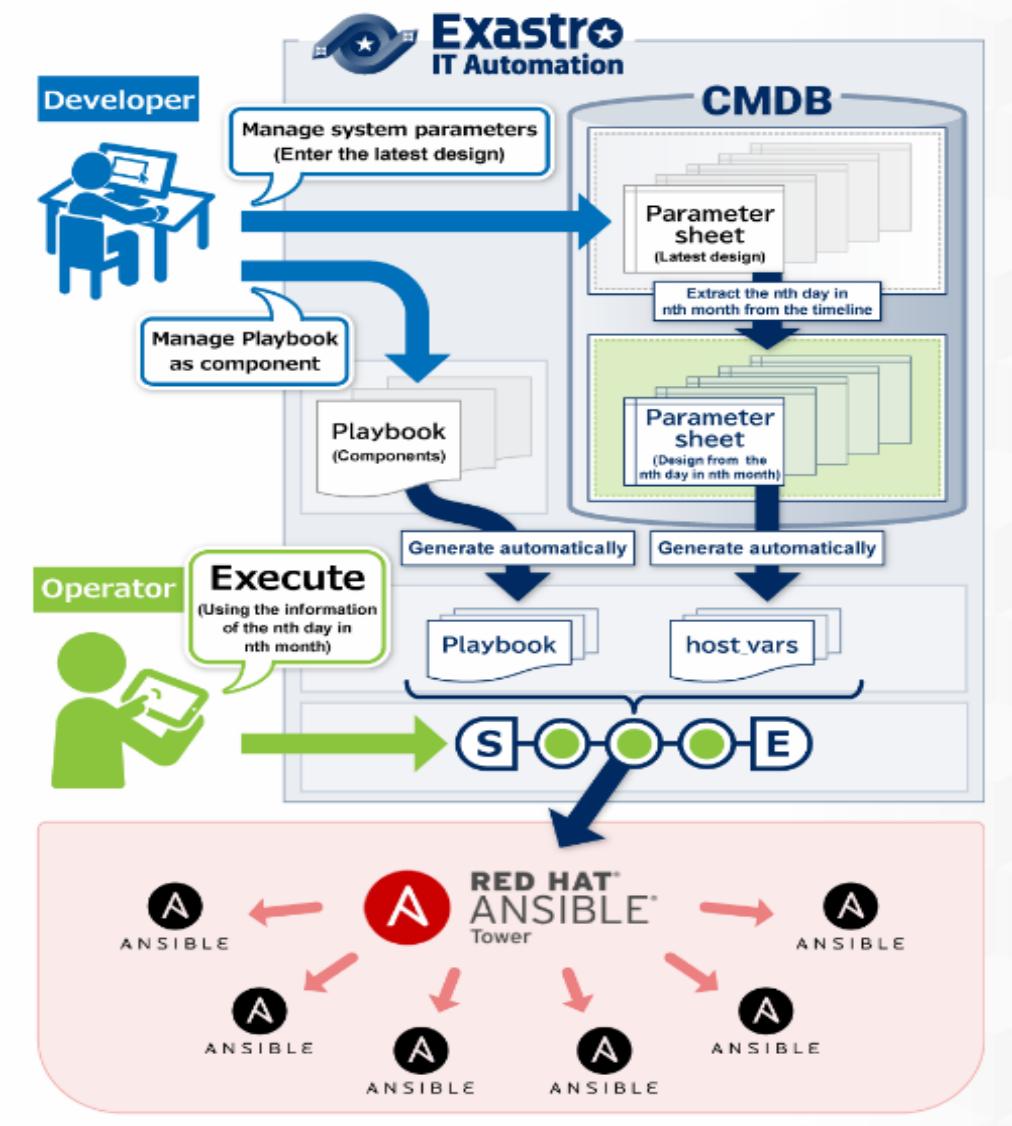
Exastro OASE supports the realization of AIOps by gathering process results while producing results in rule-based automation.



Case



[Case 1] Streamlining various operation monitoring settings for a large-scale system(1/2)



【Case 1】 Streamlining various operation monitoring settings for a large-scale system(2/2)

Effects

Example 1: Used in monitoring setup operation

Before configuration management / automation

Workload of 4 billion yen is carried out per year.

4 billion yen / year

After configuration management / automation

Can now be implemented in a workload of 1 billion yen per year. (75% reduction)

1 billion yen / year

75% reduction



Example 2: Used in command list maintenance operation

Before configuration management / automation

Workload of 40 million yen is carried out per year.

40 million yen / year

After configuration management / automation

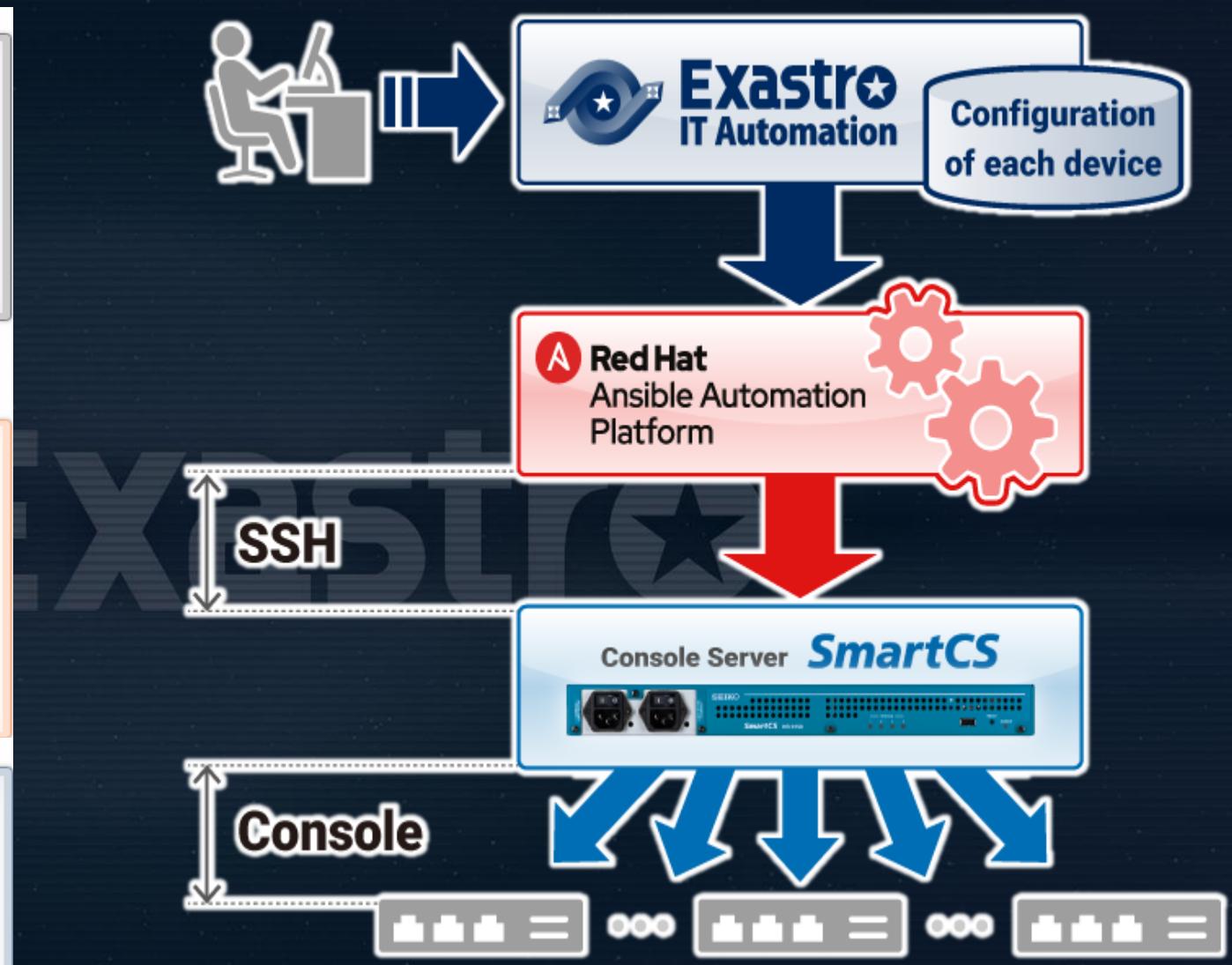
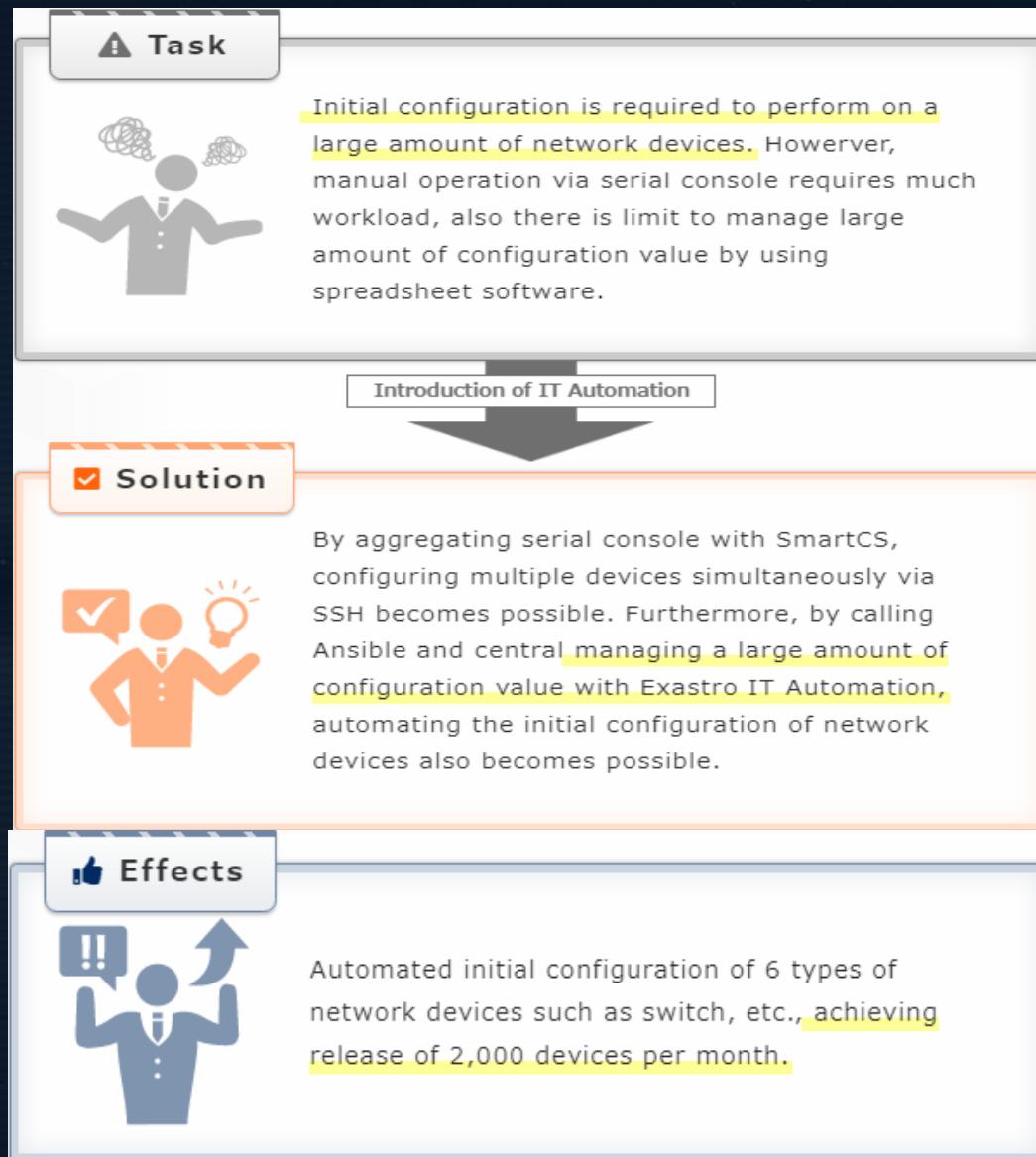
Can now be implemented in a workload of 8 million working load per year. (80% reduction)

8 million yen / year

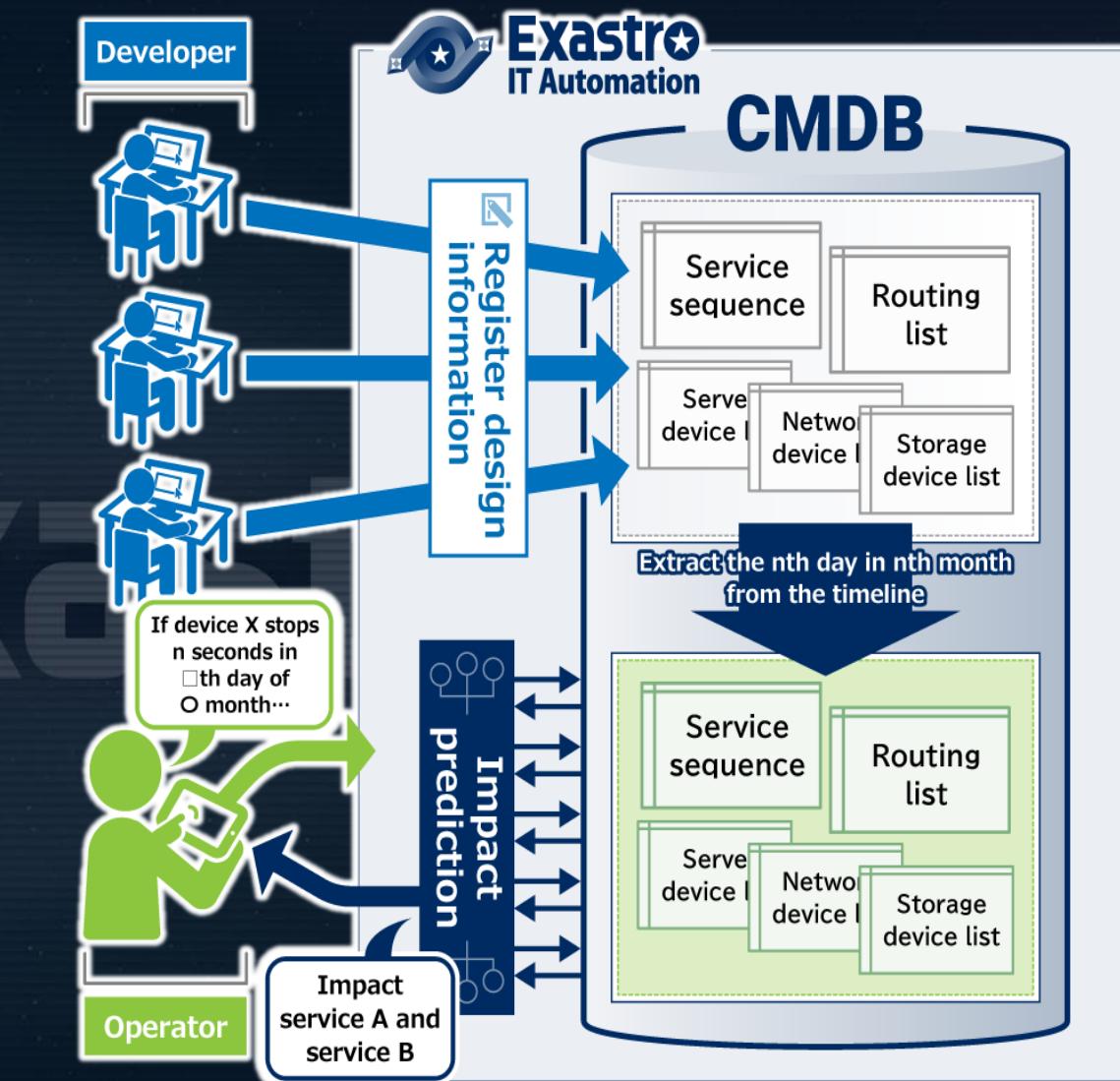
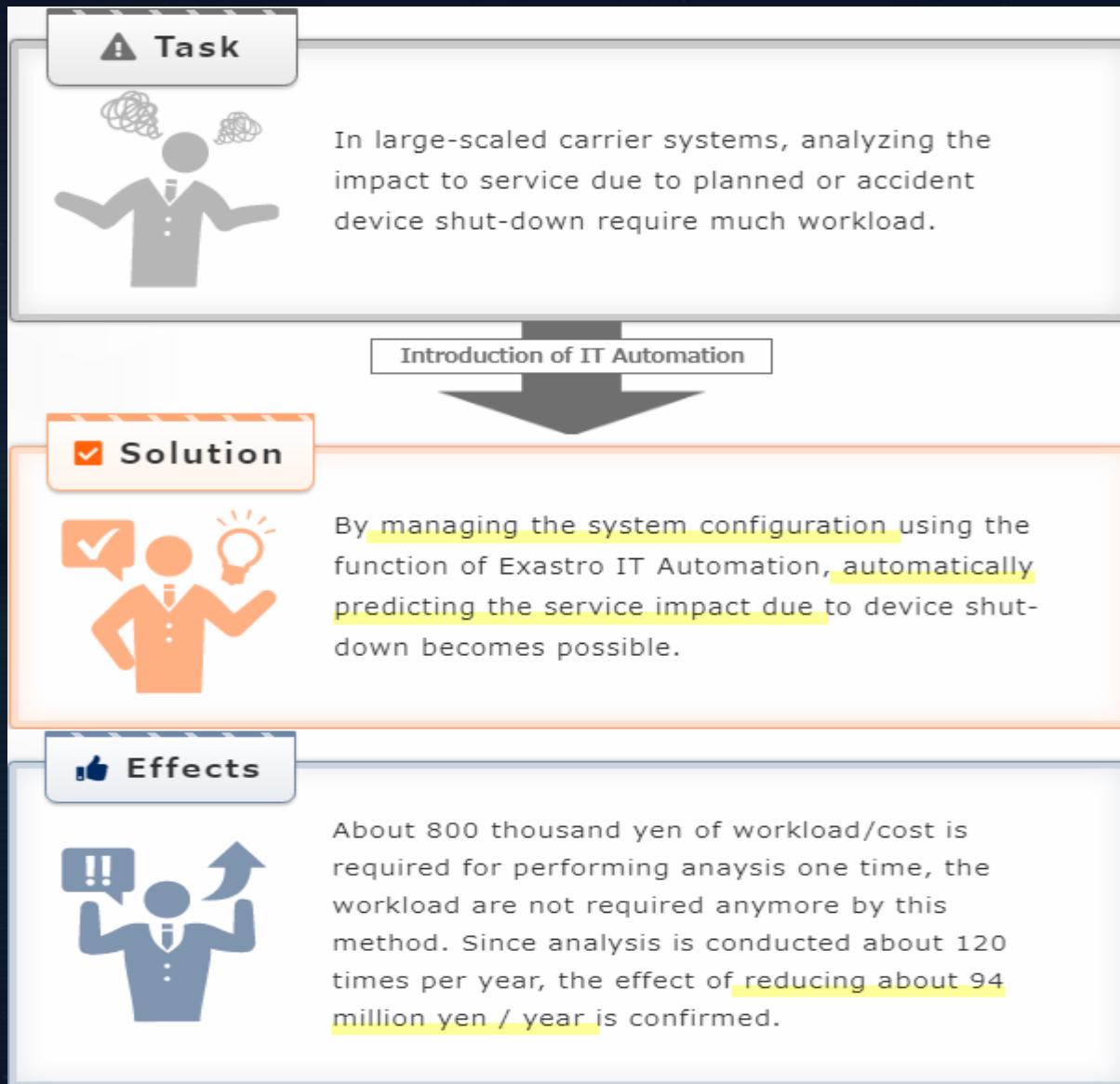
80% reduction

The convenience was also improved by the immediate reflection because the lead time until the maintenance is no longer required.

【Case 2】Automatic kitting 30 000 devices for a large-scale event.

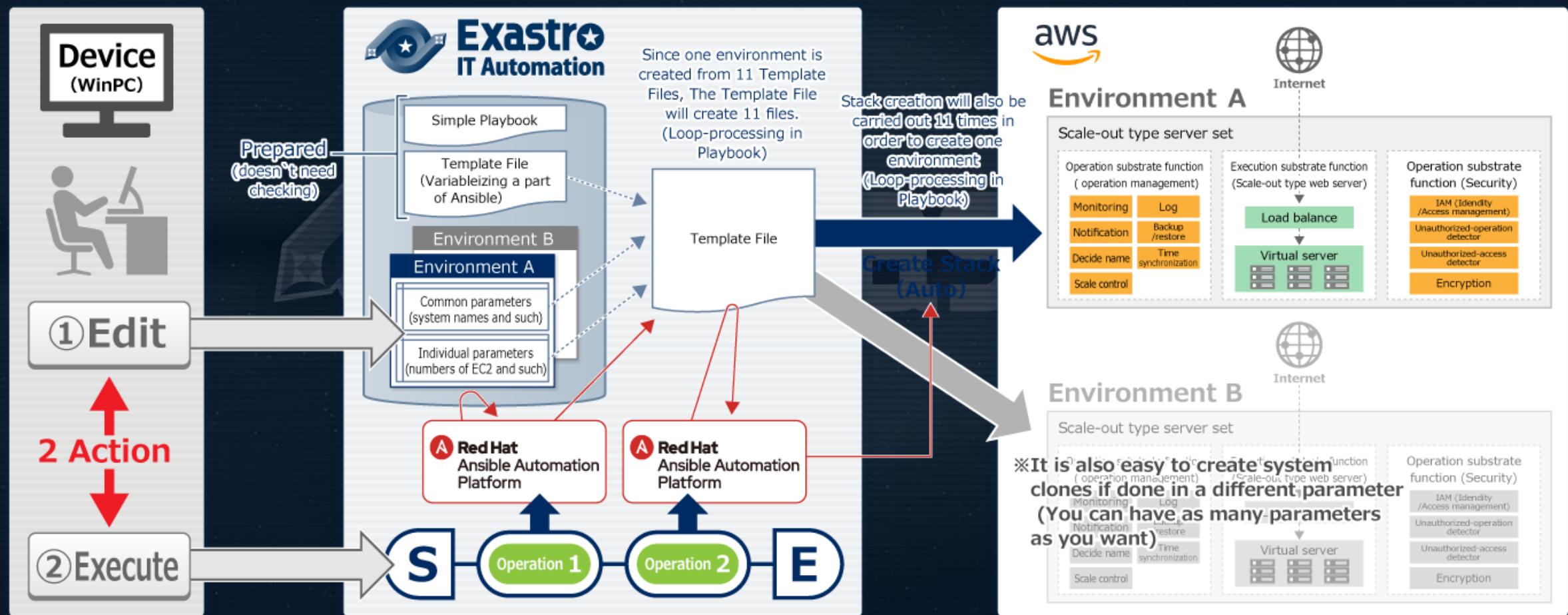


【Case 3】 Using system configuration lists to predict service impact of equipment outages.

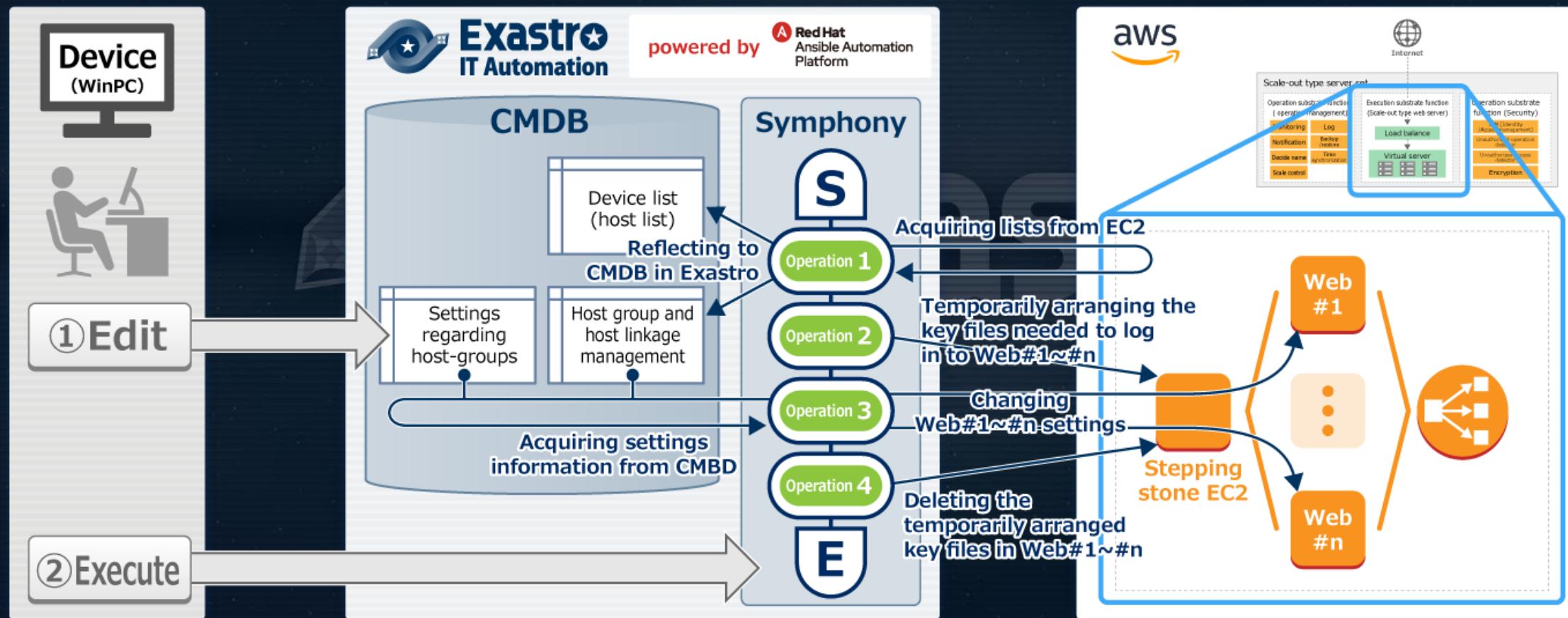


【Case 4】 How corporation's operation division controls each division's AWS system (1/2)

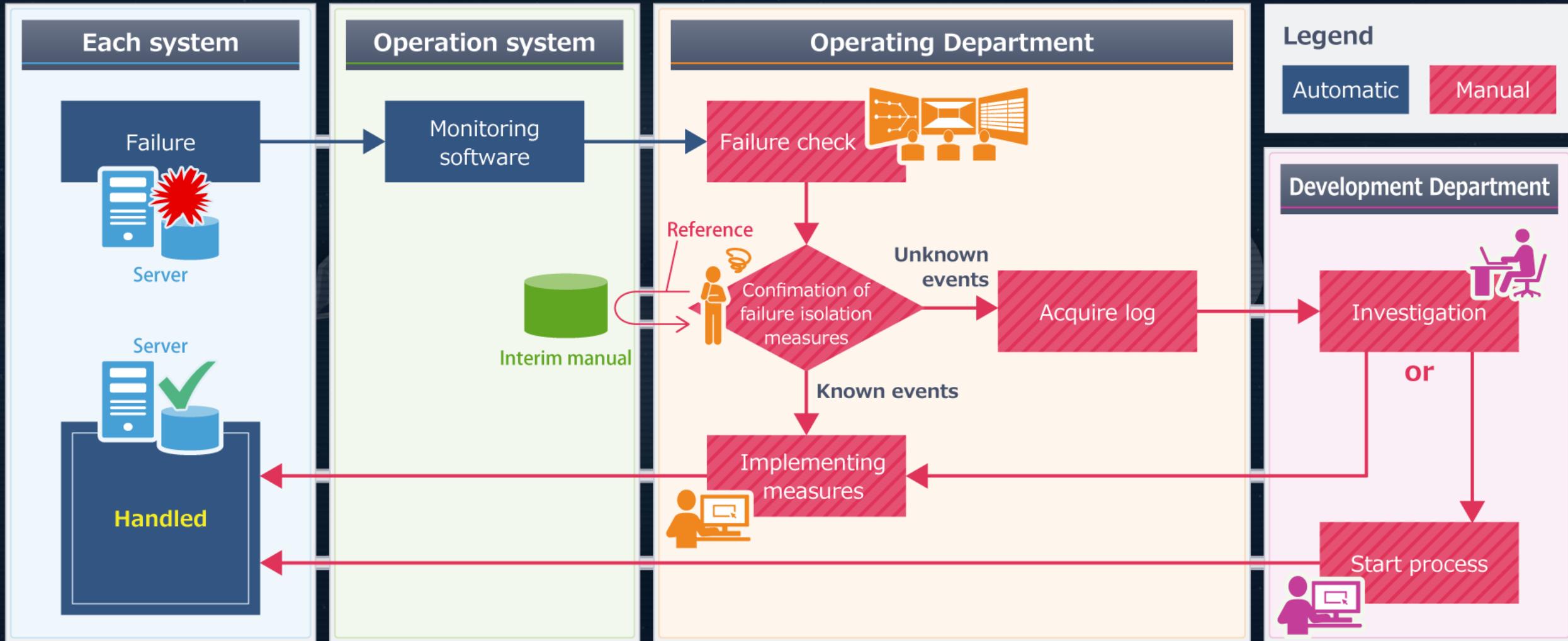
Exastro IT Automation managed a “CloudFormation template” and provided a mechanism that delivers a well-governed AWS environment to each department.



It also provided support for operational scenarios such as emergency patching of an running EC2 (Autoscale)

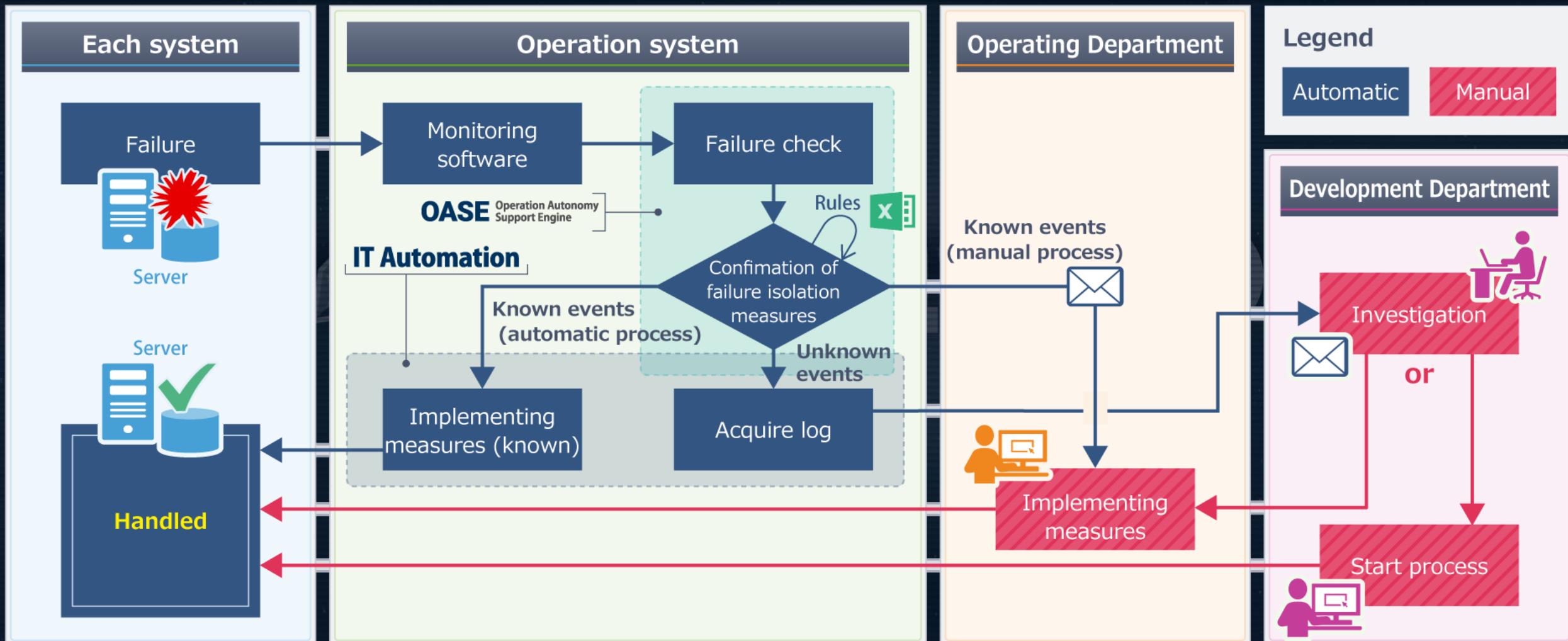


The whole operation of handling problems had to be done manually.



【Case ⑤】 Automated fault handling in operational tasks – Before –

We were able to minimize the amount of manual labor by using Exastro





Reference 1) Our goal of making Exastro OSS

Our goal of making Exastro OSS.

The aim of making Exastro Suite OSS was to be able to share it widely in the current market and collaborating with other software and its developers to solve the “pain” of IT Engineers.

NEC has developed IT Automation, a software that supports automating system construction and released it as an open source software (OSS) in April 2019.

One of the challenges of building large-scale systems has been the repetition of manual routine tasks, such as setting up the devices that make up the system, as well as the variation of quality and efficiency of the workers.

NEC has been building and developing mission-critical large scale-systems and has accumulated SI knowledge and experience. Exastro IT Automation is a software built to handle problems built from that same experience and previous knowledge and have been using it to automate a wide variety of system construction projects. However, recently, large-scale system construction have been done by multiple companies, meaning that the need for automation tools, such as Exastro IT Automation, is increasing.

By making Exastro IT Automation OSS, we aim to share it widely in the market and form an ecosystem in collaboration with other software to create a new business model where we can practice social value creation.

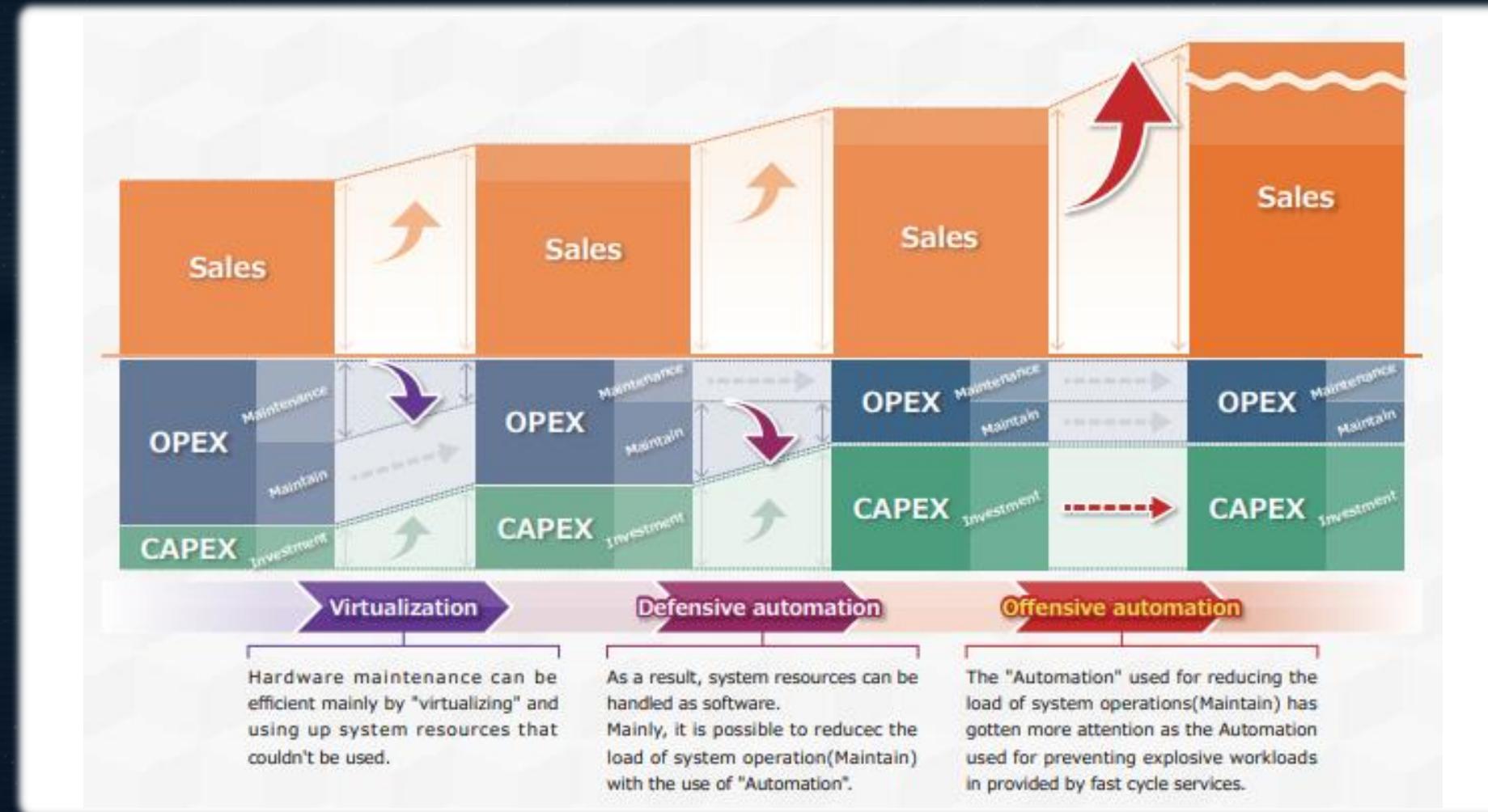
Source : <http://www.news2u.net/releases/165517/>

Note : Old name 「astroll IT Automation」は「Exastro IT Automation」に書き直している

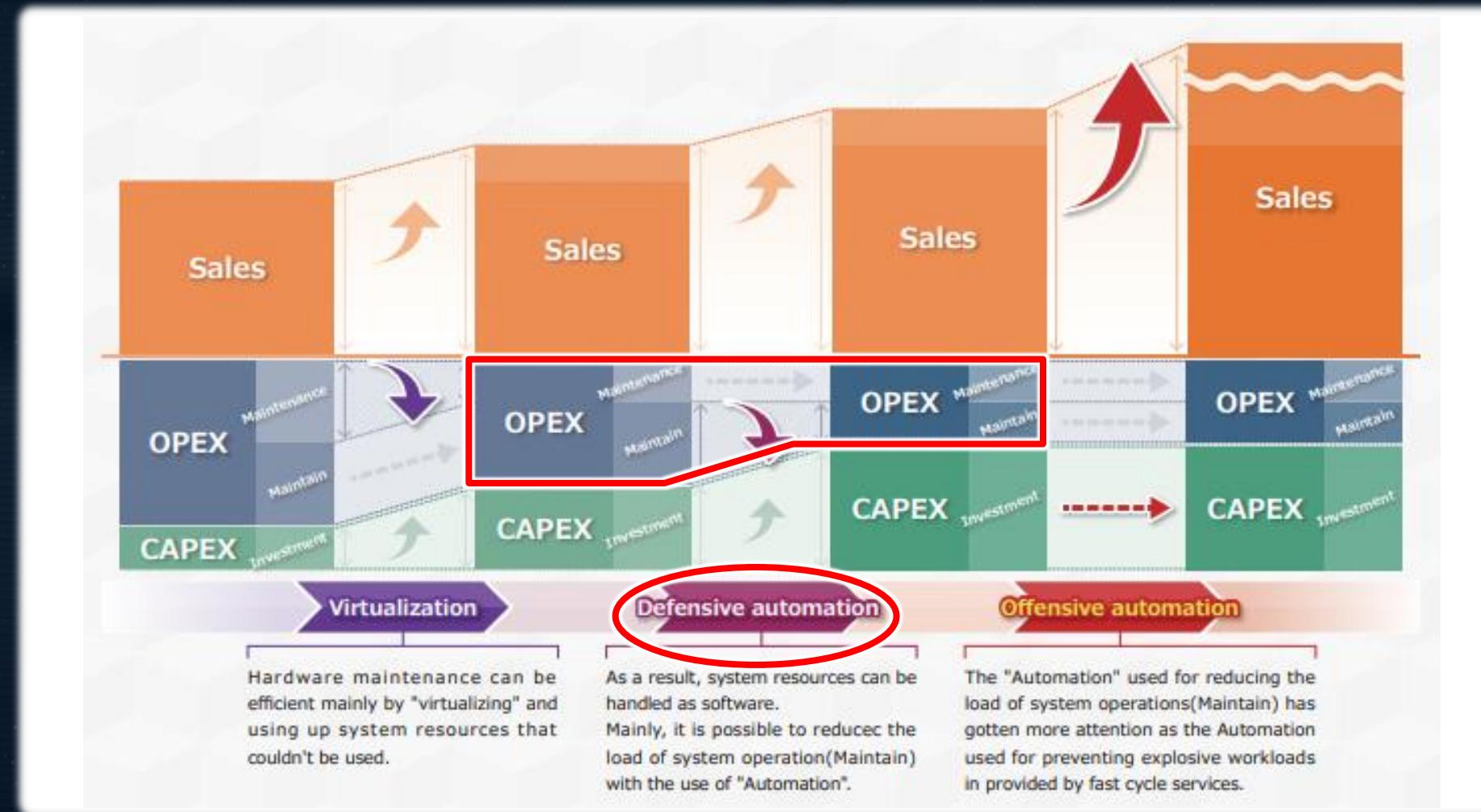
Reference2) Offensive and Defensive Automation



The 2 different areas of automation have different objectives. Even from a financial perspective.

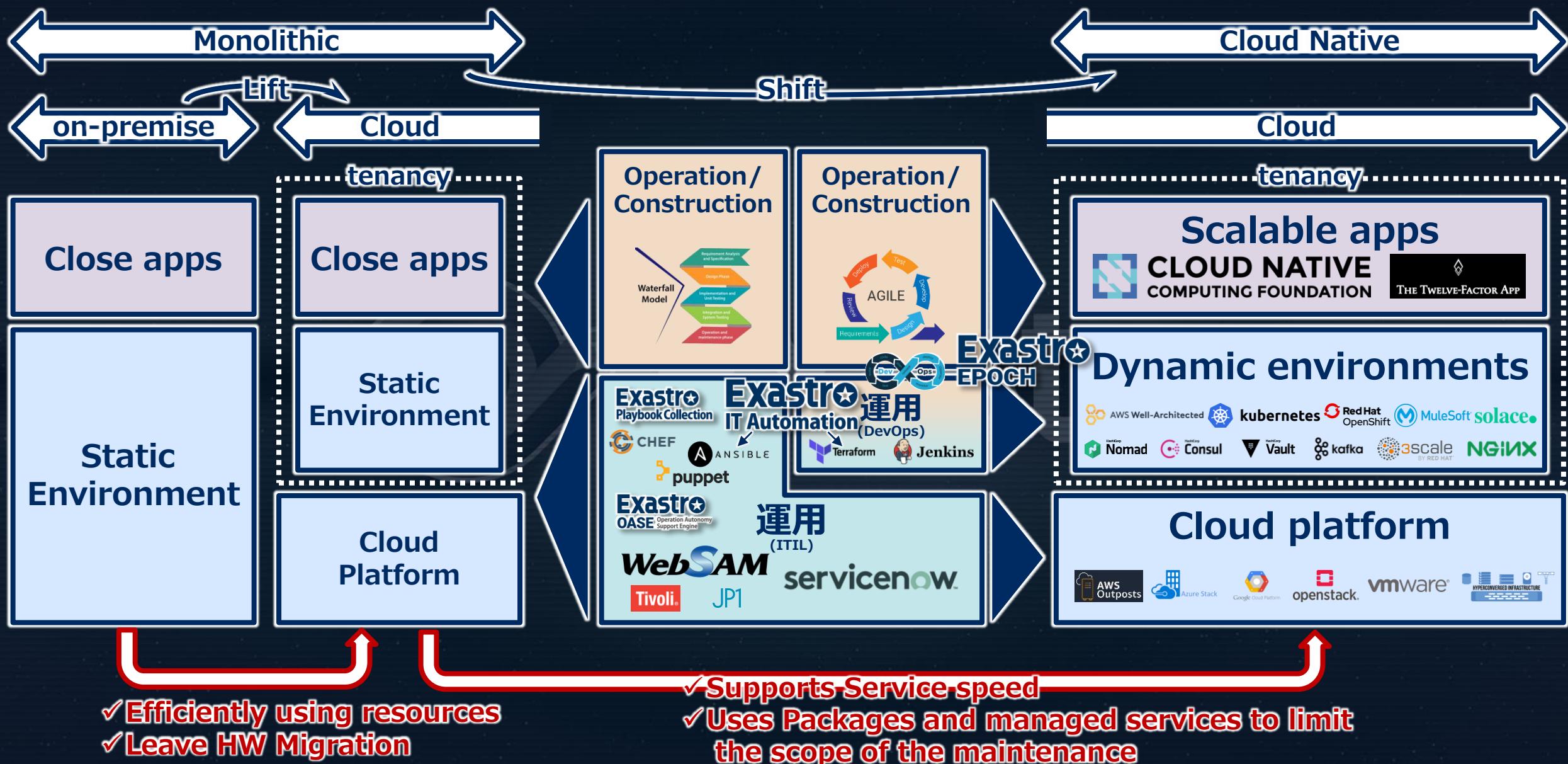


Defensive automation often aims to improve OPEX efficiency (maintenance costs).

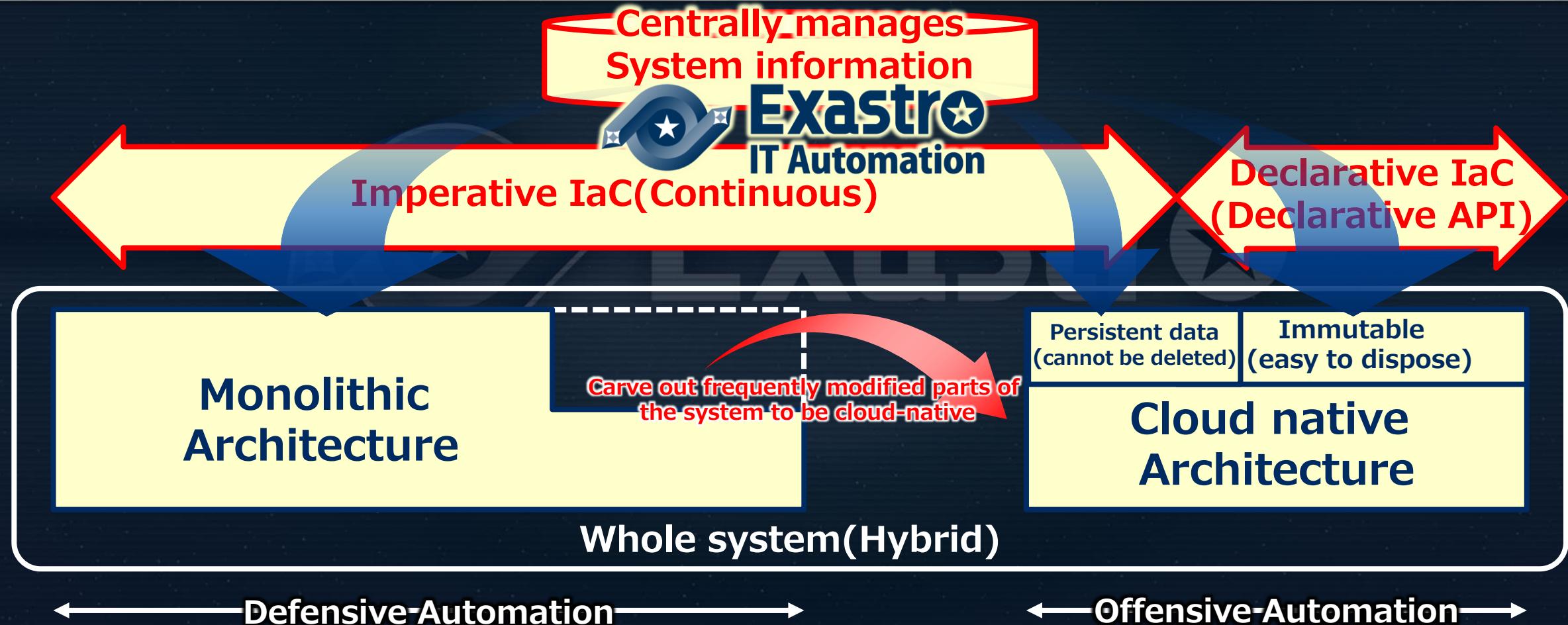


Offensive automation often aims to increase ROI (Return on Investment)





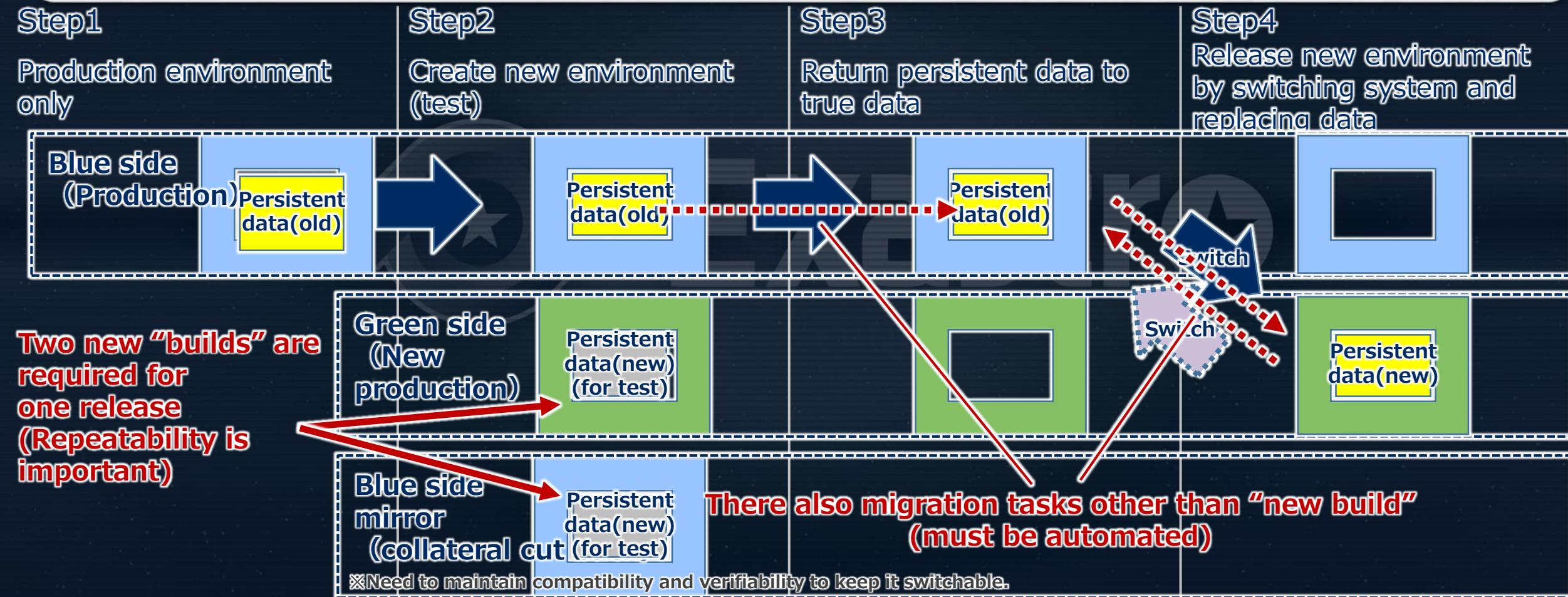
Exastro ITA centrally manages information for the whole system, not separately.





Reference 3) Offensive Automation

**Blue green deployment partially repeats “new builds”.
If done manually, the amount will be too much to handle, so an automatic construction is a must.**

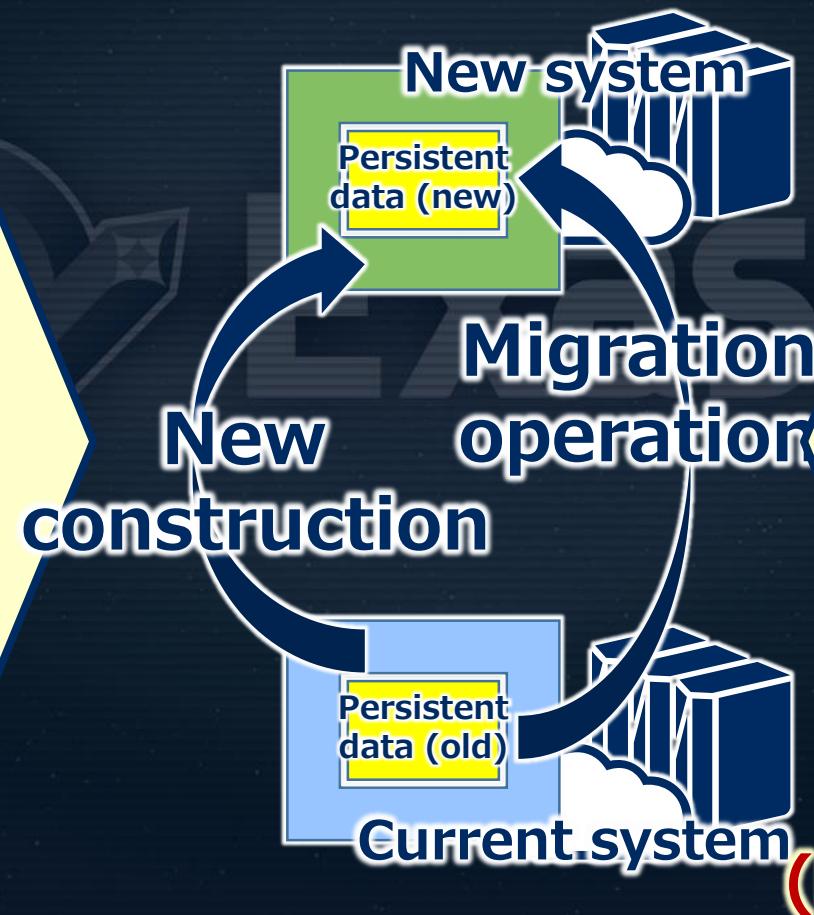


Declarative APIs (that defines) are the best for “new builds”
On the other hand, the “migration work” needs to be automated with imperative IaC.

Declarative IaC (Declarative API)

The final result will be as following
1 of OO
2 of □□
3 of △△

Declarative IaC suits building new cloud resources



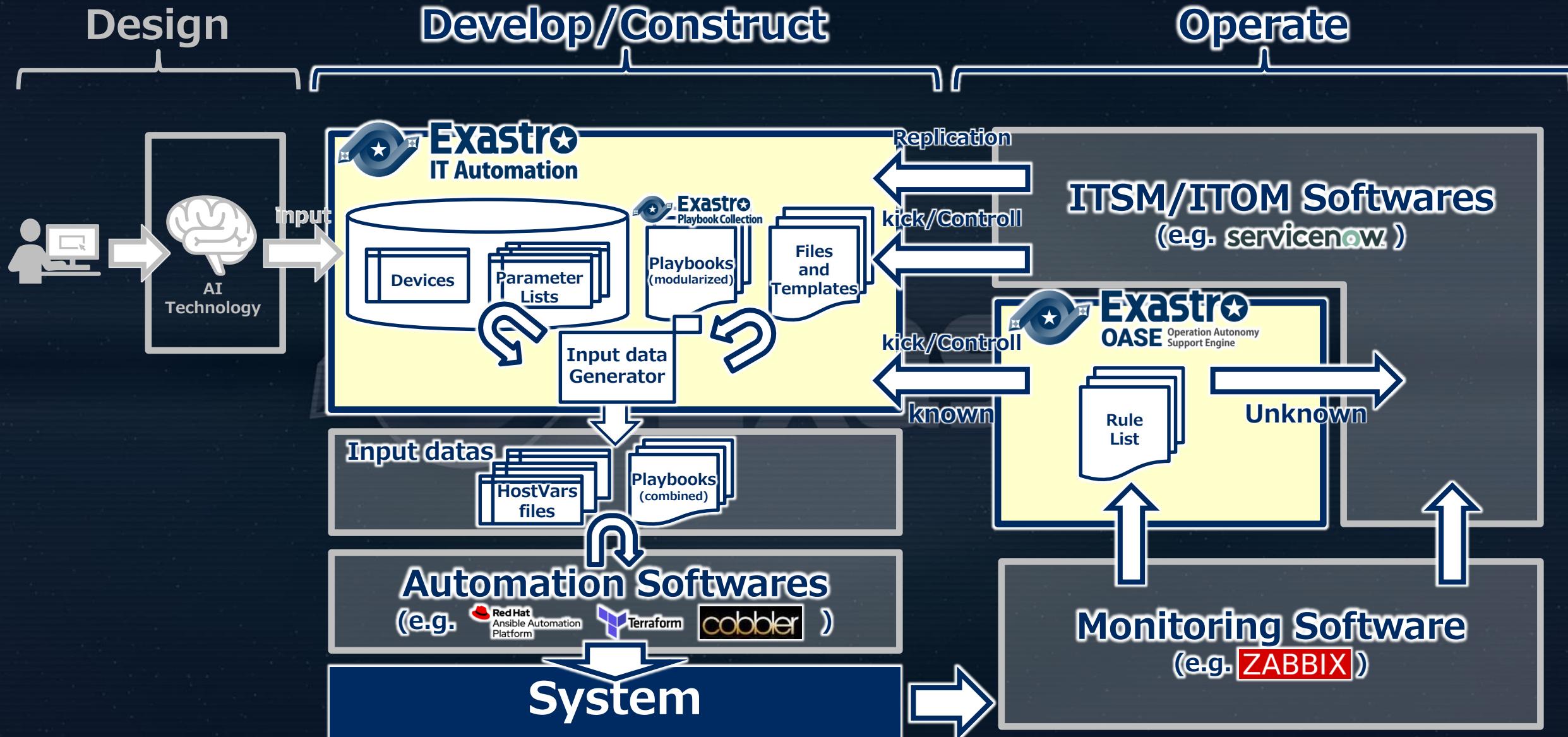
Imperative IaC (Continuous)

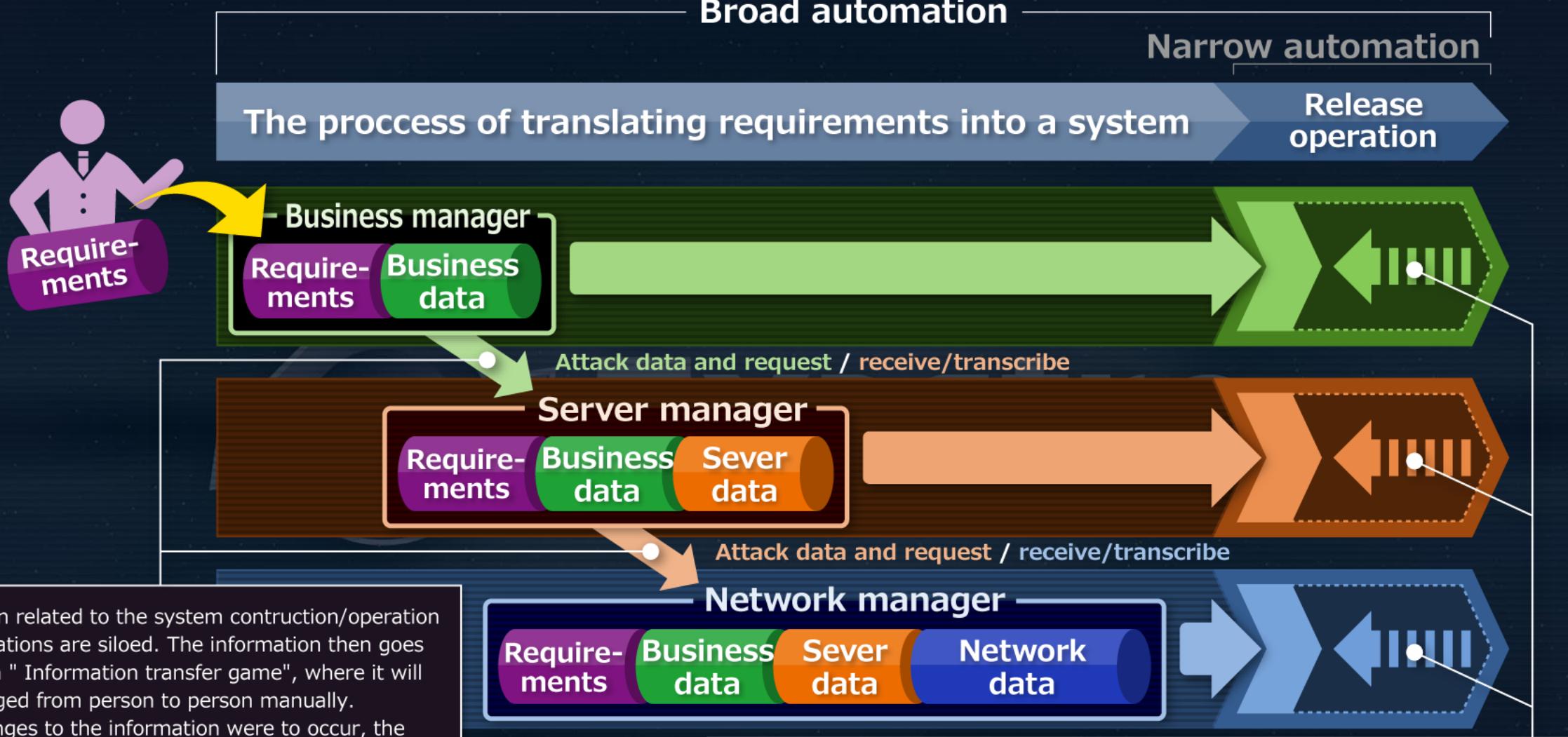
1. Prepare 1 of
2. Mix A and B to create 3 of △△
3. Mix C and D to create 2 of □□

There is a need to automate operations with Imperative IaC (Except for cloud resources)

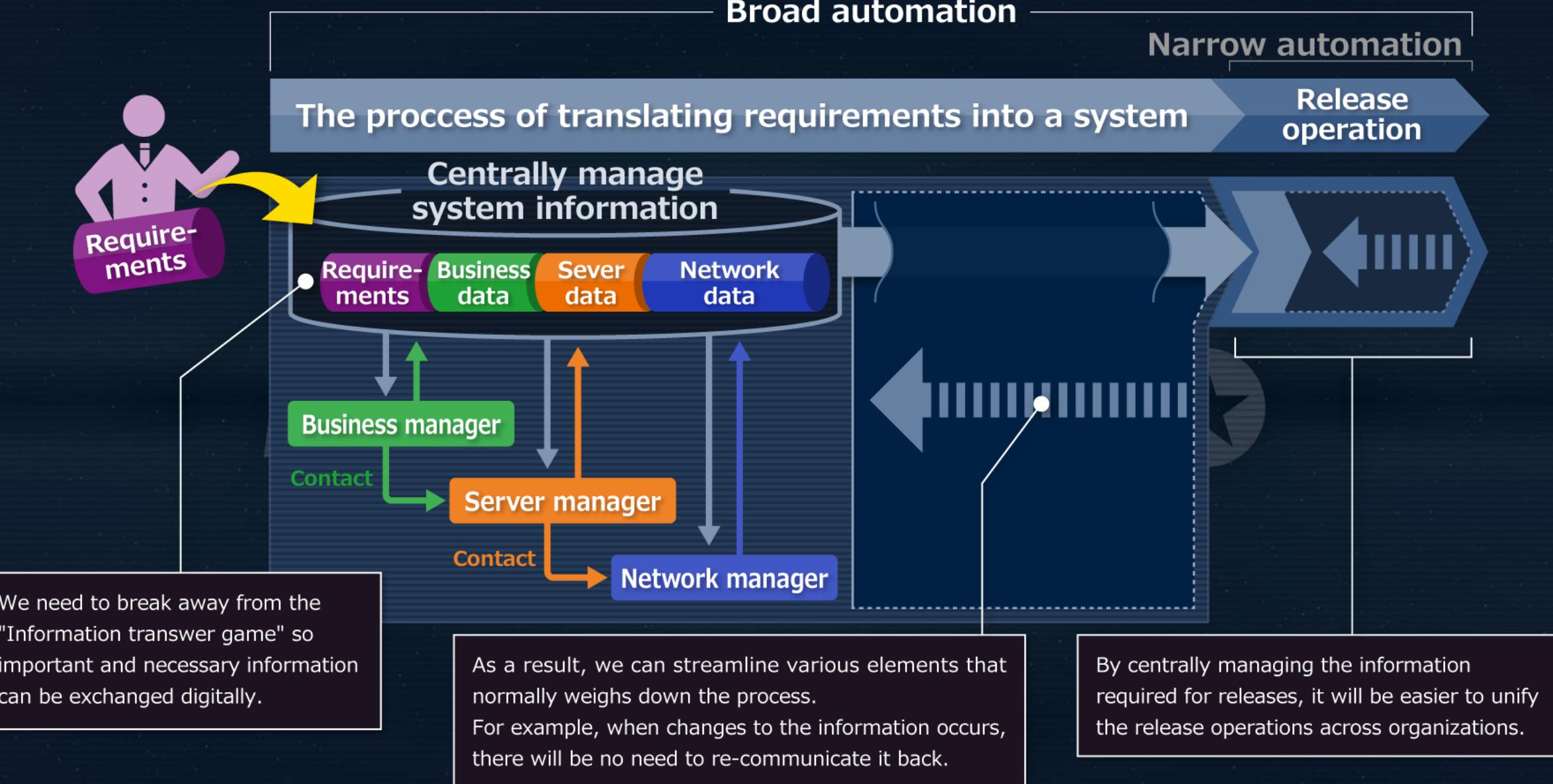


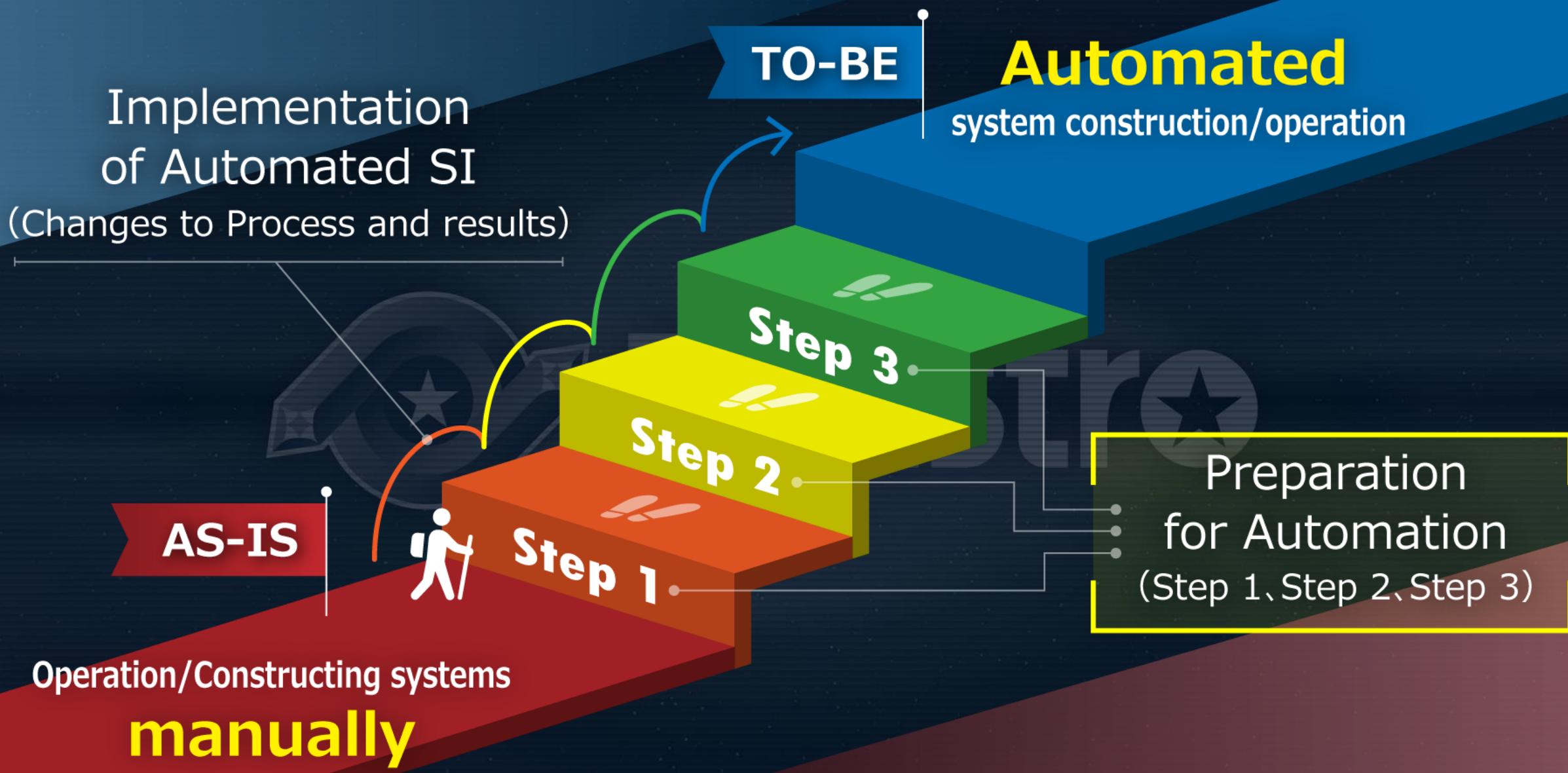
Reference 4) Defensive automation





While it is not a bad thing that the release process is heavily streamlined, but it only streamlines a part of the system construction/operation.





These problems can generally be solved in 3 steps



Design



Preparation

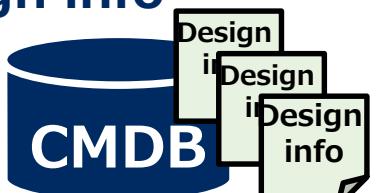


Execution



Solution

Step 1
Centrally manage
Design info

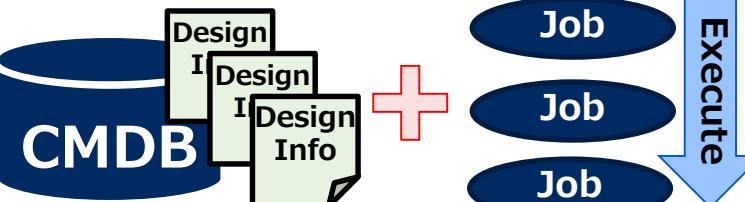


Solution

Step 2
Realize automated
Executions



Step 3
Link centrally management
and automated execution.



More specifically, it can be solved with **12 smaller tasks**.

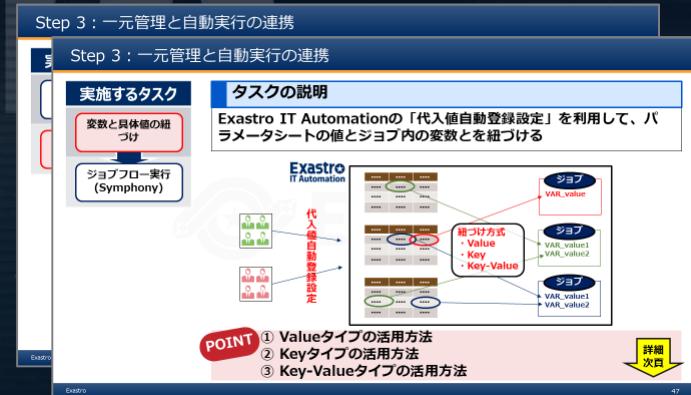
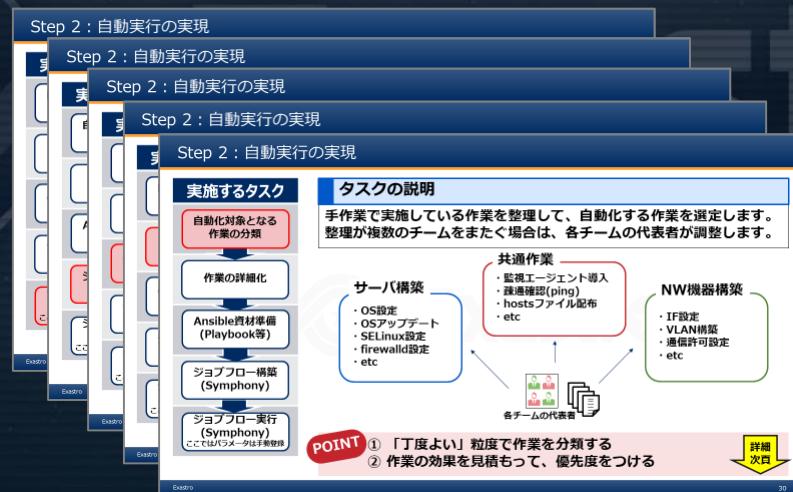
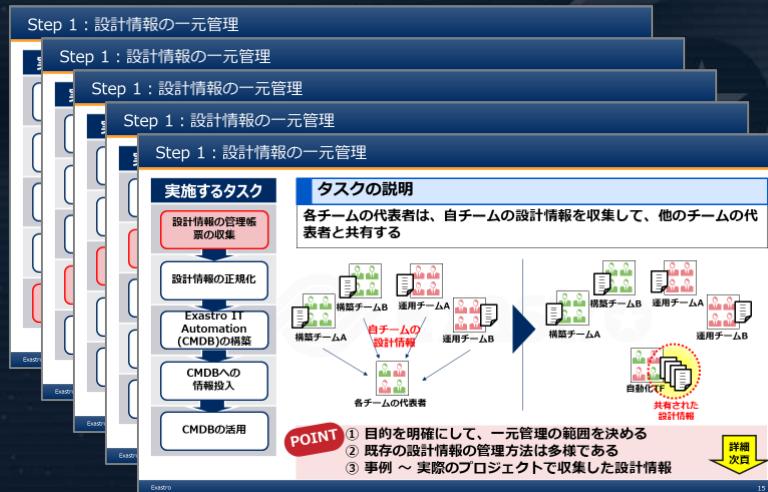
Step 1



Step 2



Step 3



5 tasks

5 tasks

2 tasks

Implementation of Automated SI

(Changes to Process and results)

AS-IS

Operation/Constructing systems
manually

TO-BE

Automated

system construction/operation



Step 3

Step 2

Step 1

Preparation
for Automation
(Step 1, Step 2, Step 3)

The guide can be downloaded from the OSS Community site

Exastro IT Automation Community site:

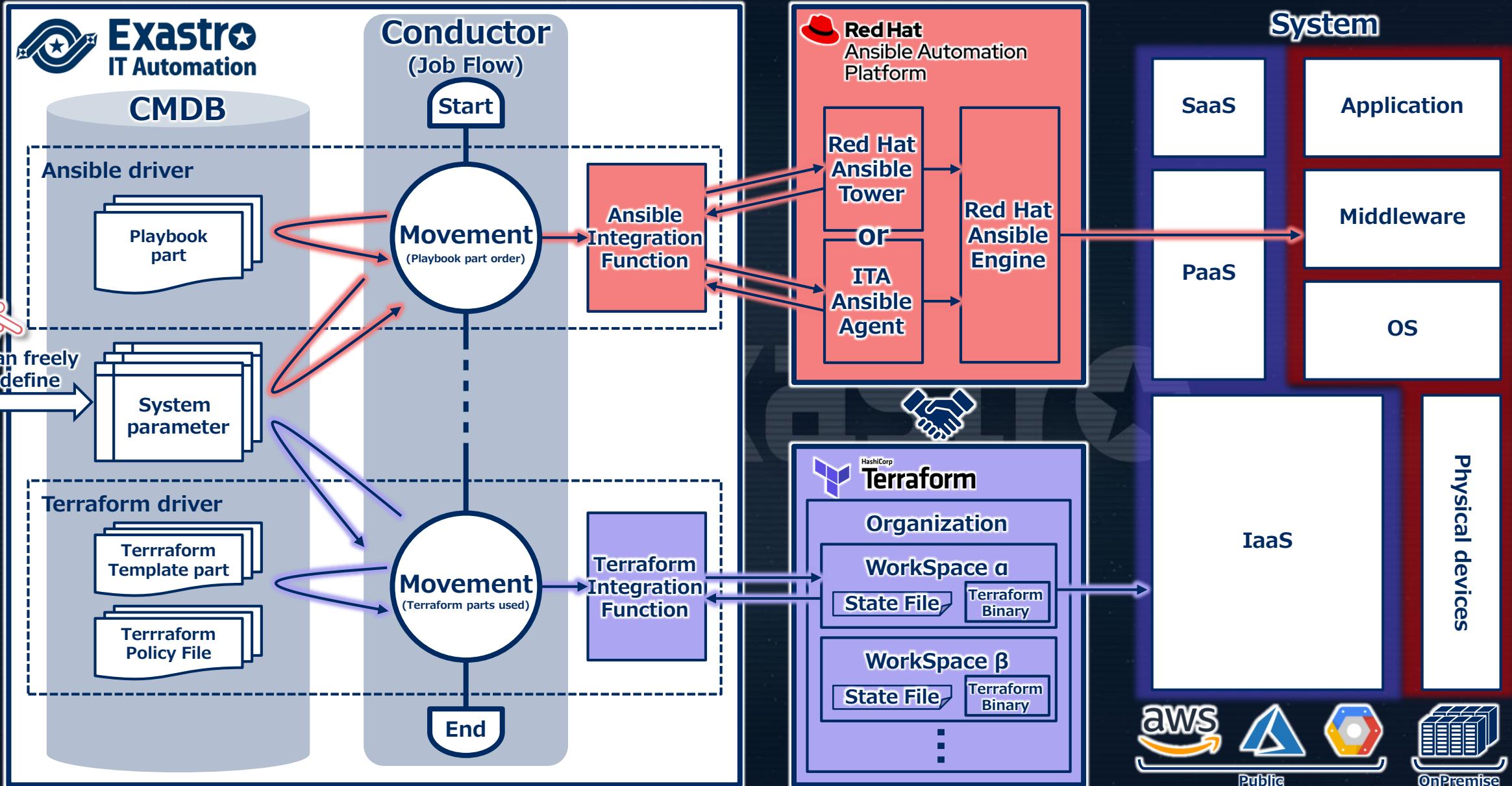
<https://exastro-suite.github.io/it-automation-docs/index.html>

The screenshot shows the Exastro IT Automation Community site. At the top, there is a navigation bar with tabs: Overview, Learn (highlighted), Case, Documents, Downloads, and FAQ. Below the navigation bar, there is a sub-navigation menu with items: Functions, Deploy, Installation, Quick start, BASE, Function introduction, and a dropdown menu for 'Guidebook' which is currently expanded to show 'System efficiency guide'. In the center of the page, there is a large box titled 'Exastro System Operation and Construction Efficiency Guide'. Inside this box, there is a paragraph of text and a red button labeled 'Download manual (PDF)'. To the right of this box, there is a section titled 'Exastro IT Automation' with a 'View PDF' button. At the bottom of the page, there is a footer with the Exastro logo, a brief description of the software, and links to Why Exastro, Exastro Suite, News Room, Asset, and IT Automation. The IT Automation section includes links for Overview, Learn, Case, Documents, Downloads, and FAQ. A QR code is located on the right side of the page.

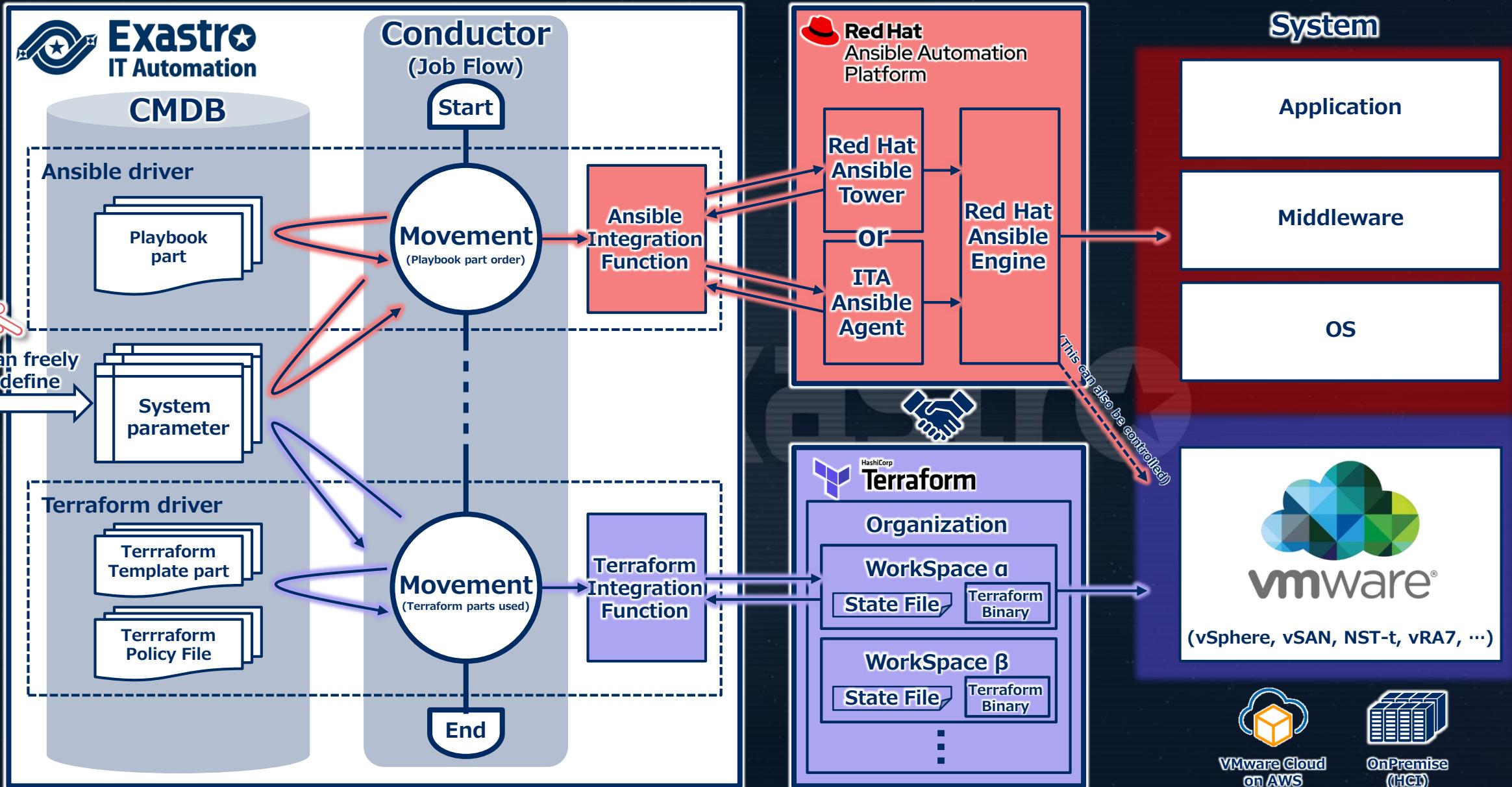


Reference 5) Ecosystem and related software

Exastro IT Automation : Where to use Automation software (cloud, etc.)



Exastro IT Automation : Where to use Automation software (Vmware Infrastructure)



Software certified by Red Hat OpenShift Operator is guaranteed to be able to run on OpenShift, enabling multi-cloud support and autonomous operation through the codification of operational know-how.

Exastro is participating in the "**Red Hat Kubernetes Operator Project**" and has operator support for deploying Exastro on Open shift.

NEC Corporation· OSS Promotion Center,
Advanced SI Technology
Development Division Leader : Suganuki Kimio

NEC has provided Openshift Products on an OEM basis since 2016, and is focused on Operator support for the latest Red Hat Openshift, including Webotx, Istorage and **Exastro**. We believe that this project will be a key initiative for improving the sophistication and convenience of container operations, and through collaboration with Red Hat, we will develop solutions that will promote DX for our customers.

Source : <https://www.redhat.com/ja/about/press-releases/red-hat-starts-kubernetes-operator-project-in-japan>

Red Hat Press release page



Benefits of linking Exastro ITA and Ansible Tower

Ref.

IT Automation

Stores/Manages configuration data and generates directories/config files needed for Ansible to run

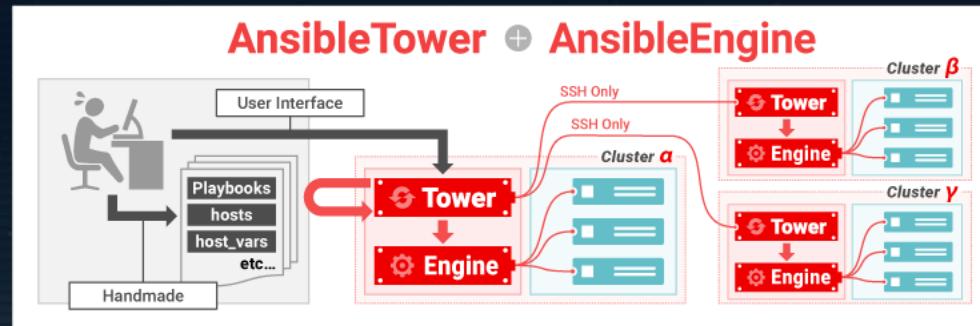
AnsibleTower

Secures communication between clusters and controls different versions of Ansible engine.

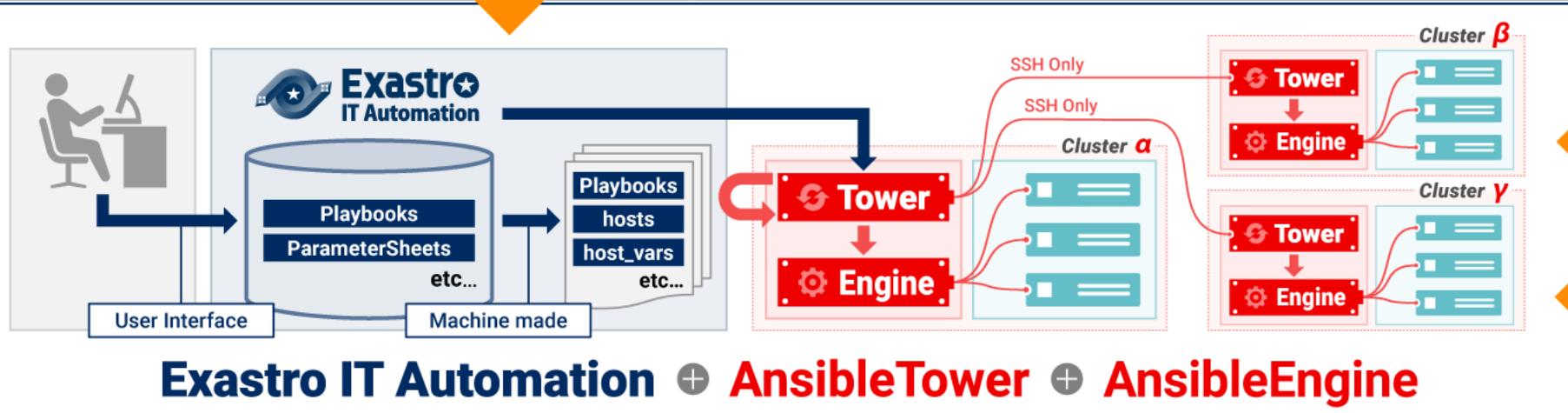
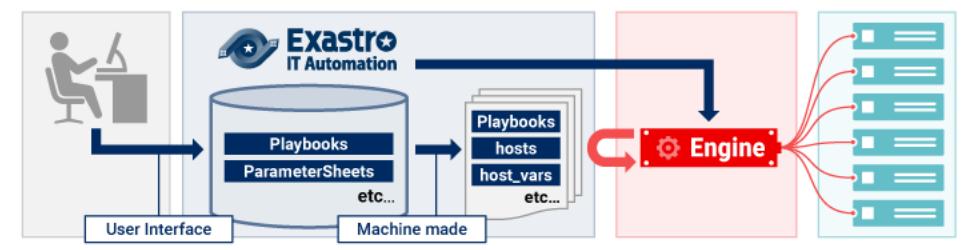
AnsibleEngine

Engine that executes AnsiblePlaybook

Combining the features of the software above creates an Automatic construction system that can improve efficiency and save labor.



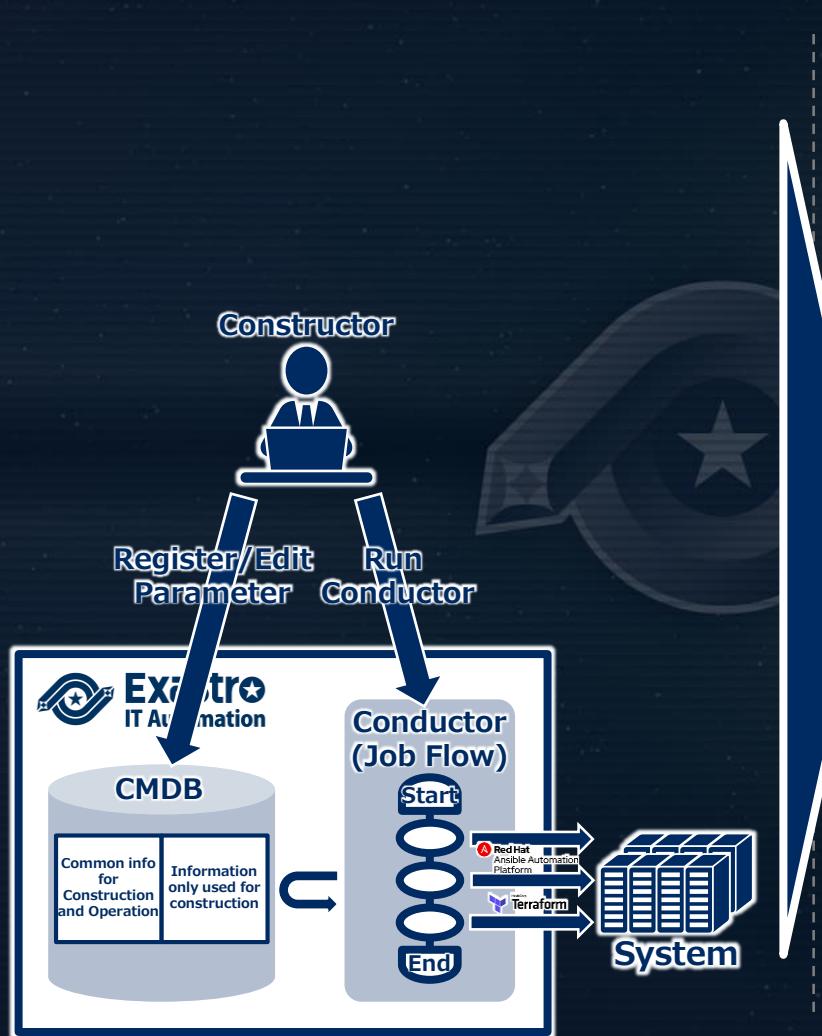
Exastro IT Automation + AnsibleEngine



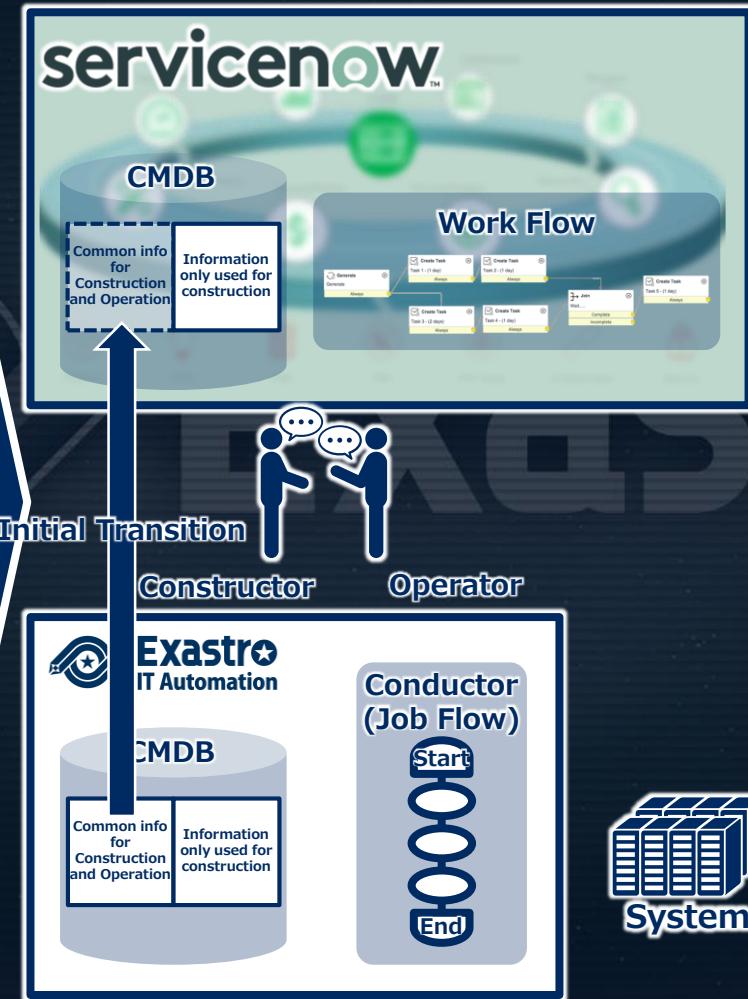
Design management
(Automatic input generation)

Assuring security (Clustering)
Multiple engine ver

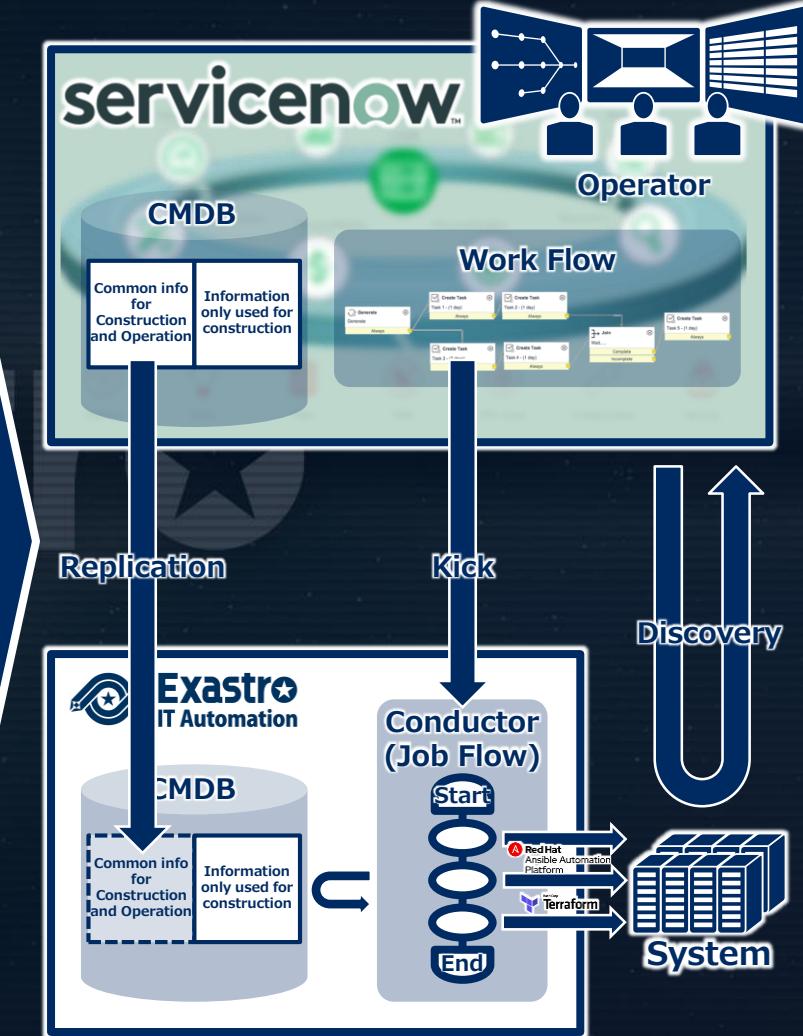
Construction



Operation prep



Operation

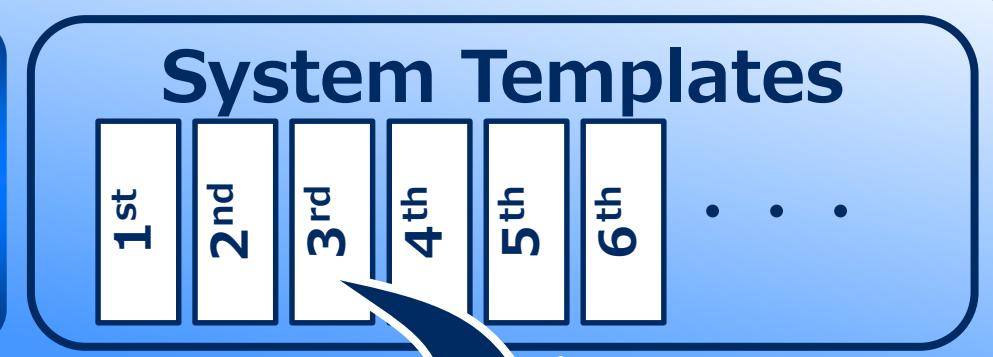




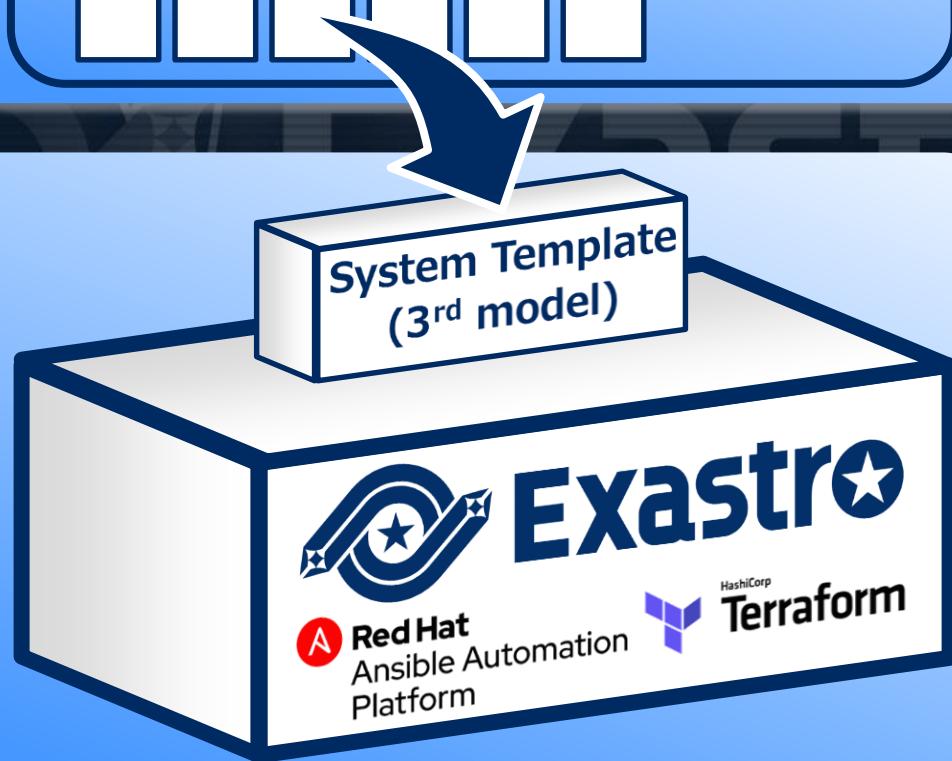
Reference 6) Setting Samples

Line up cartridge type system templates as Setting Samples.

Line up system templates and turn the SI into menus (Cartridge type)



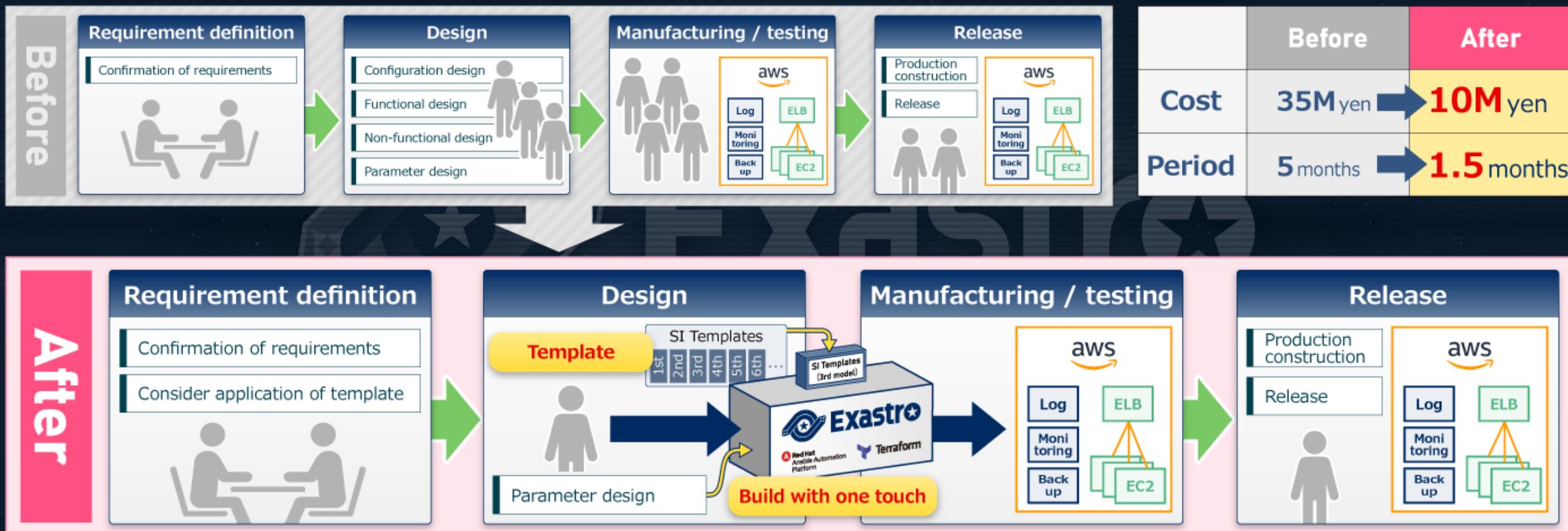
Exastro pursues ease of use as an execution base for system templates



Deploying SI Menus to various cloud environments



Effects of incorporating Setting Samples (1st model)





Exastro 