IT Automation
Ansible Driver
Tutorial

※In this document, “Exastro IT Automation” will be written as “ITA”

Version 1.3

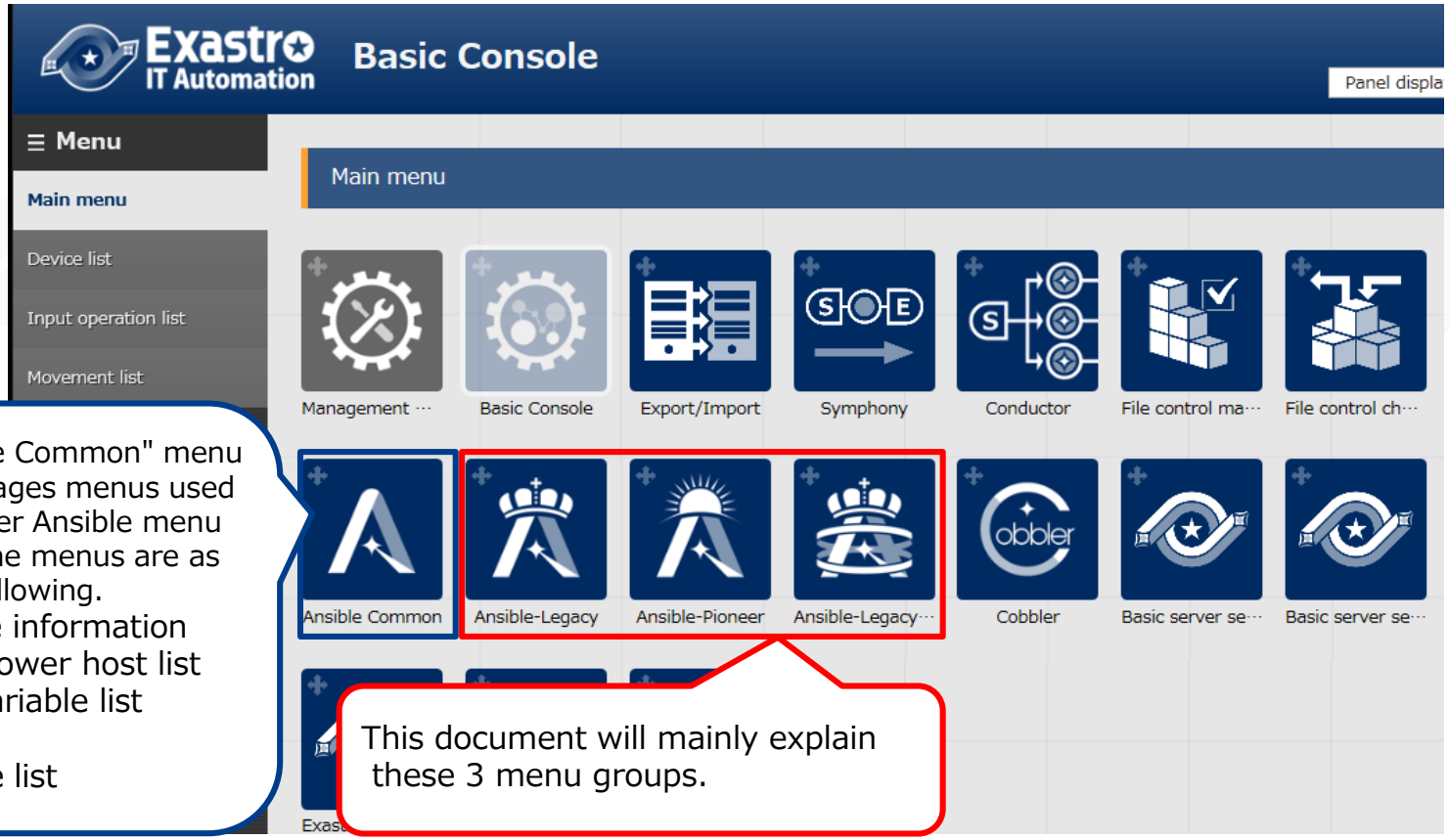
Exastro developer

Table of contents

1. Introduction

Main Menu

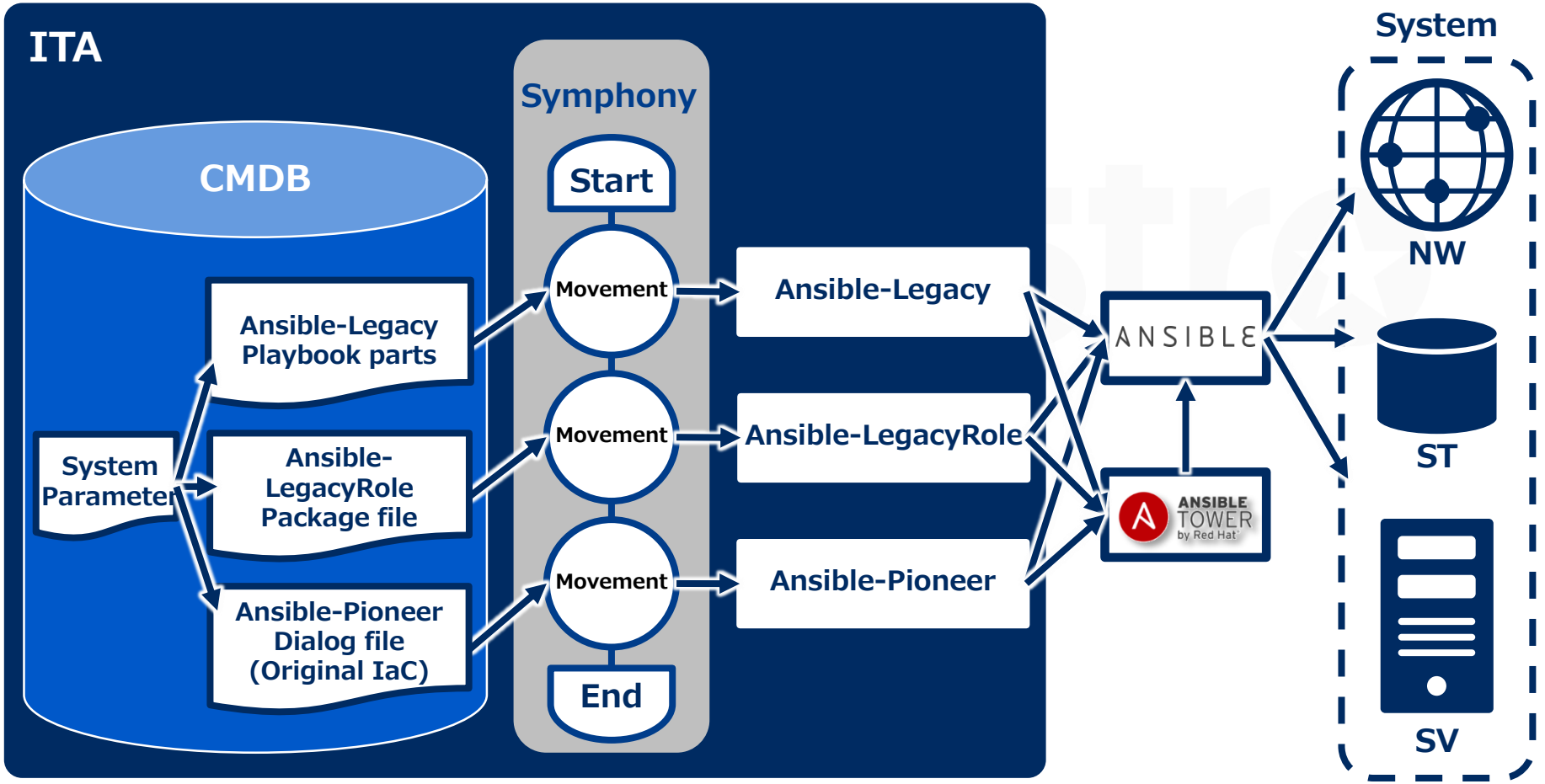
- This document will explain the functions andconcept of the menu groups: "Ansible-Legacy", "Ansible-LegacyRole" and "Ansible-Pioneer".
- We`ve prepared a practice document that uses screenshots to explain, so feel free to use that together with this document.



2. What is Ansible Driver?

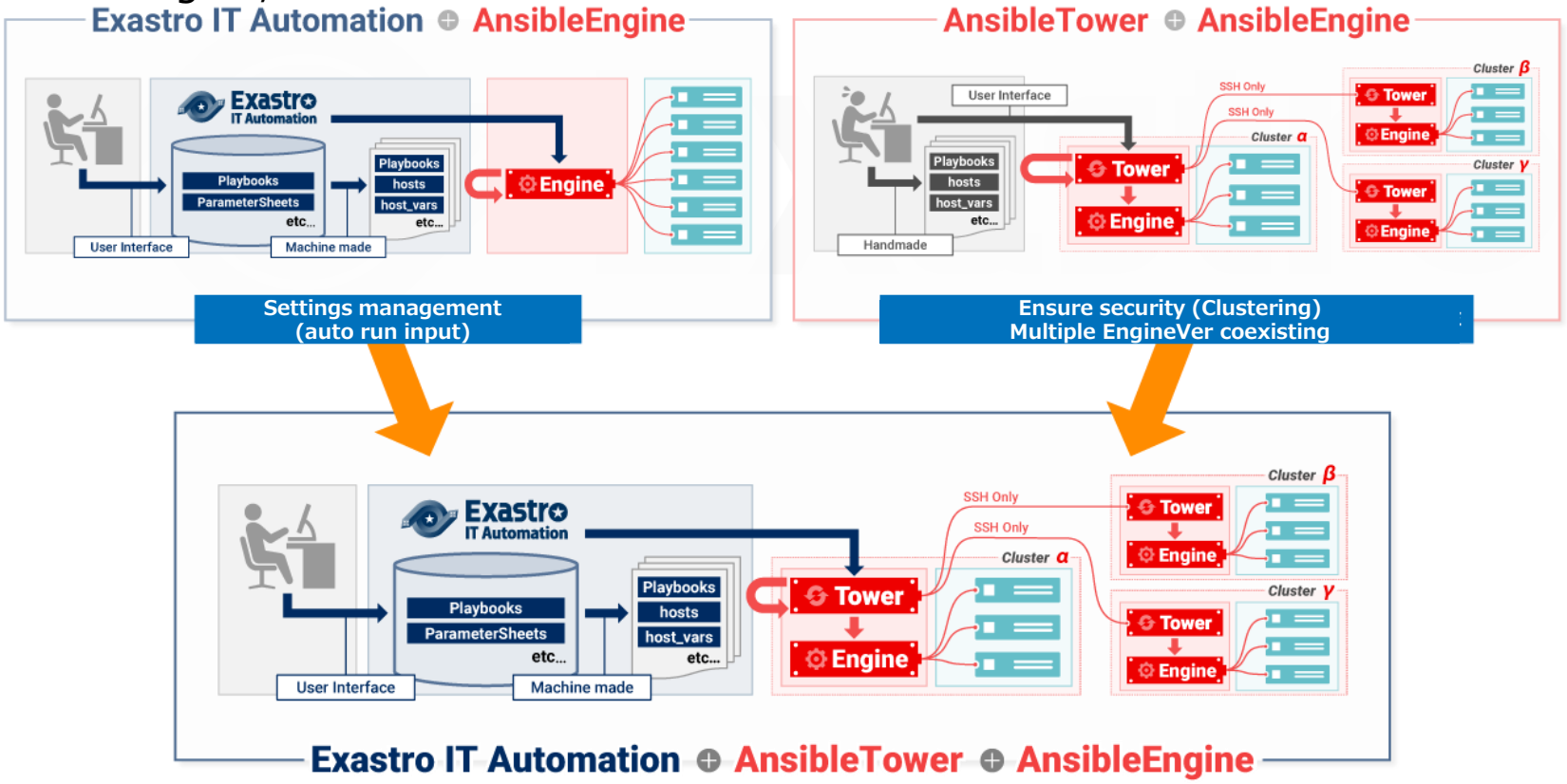
Ansible Driver links the system parameters that ITA centrally manages and the variables of IaC (Playbooks and such) and allows the system to cooperate with Ansible.

※We will explain the merits of using Ansible Tower in "3. Connecting to Ansible Tower".



3. Linking to Ansible Tower

- IT Automation stores and manages settings data and creates the directories and configuration files necessary to operate Ansible.
- AnsibleTower makes the signals between clusters more secure and controlls Ansible Engine
- Executing operations on an automatically constructing system created by combining the features of the 3 modes, IT Automation + AnsibleTower + AnsibleEngine, can be both save labour and be more efficient.



1.4. Explanation of the different Ansible Modes

Ansible Driver provides three modes with features that can handle different situations.

- The three modes, Ansible-Legacy, Ansible-LegacyRole and Ansible Pioneer, and their features are explained below.

◆IaC reusability
1. **Ansible-Legacy**
2. **Ansible-Pioneer**
3. **Ansible-LegacyRole**

◆Know-how application
1. **Ansible-Legacyrole**
2. **Ansible-Legacy**
3. **Ansible-Pioneer**

◆Application range
1. **Ansible-Pioneer**
2. **Ansible-LegacyRole**
3. **Ansible-Legacy**

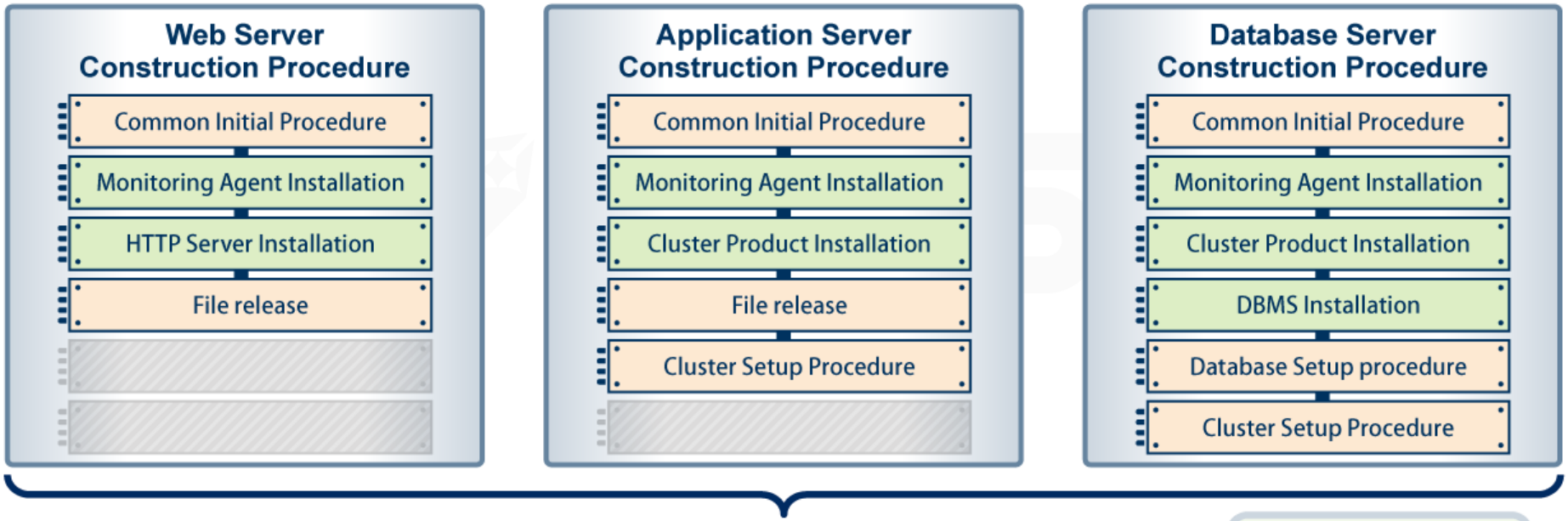
※Remarks
◎ : Strong point/ Specialty
○ : Normal / Possible to do
× : Not possible/Possible with the help of other modes

	Description	Ansible -Legacy	Ansible-Legacyrole	Ansible-Pioneer
IaC reusability	Turning Playbooks into modules and reuse them in Exastro.	◎	×	○
Applying know-how	Apply the many features that Ansible provides. Use Playbook Roles released by Ansible Galaxy and such.	○	◎	×
Range of application	Big variety of what kind of operation that can be operated.	○	○	◎

5. Features of each mode
5.1 Ansible-Legacy mode

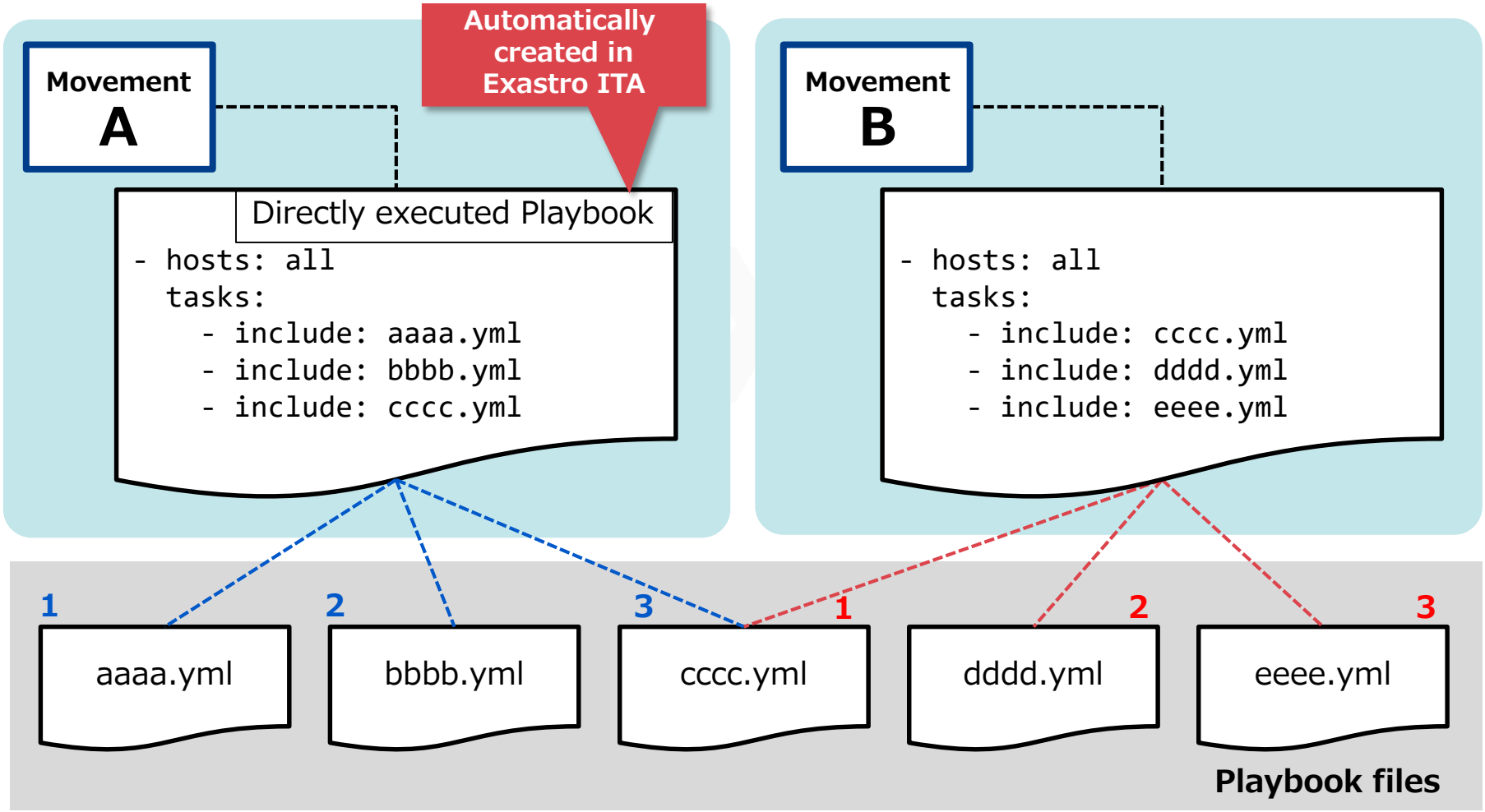
The real charm of using IT as base – Ansible Legacy Mode

- **The prime feature of Legacy is modulating IaC and reusing them.**
- **By reusing registered IaC, it is possible to construct more efficient systems.**



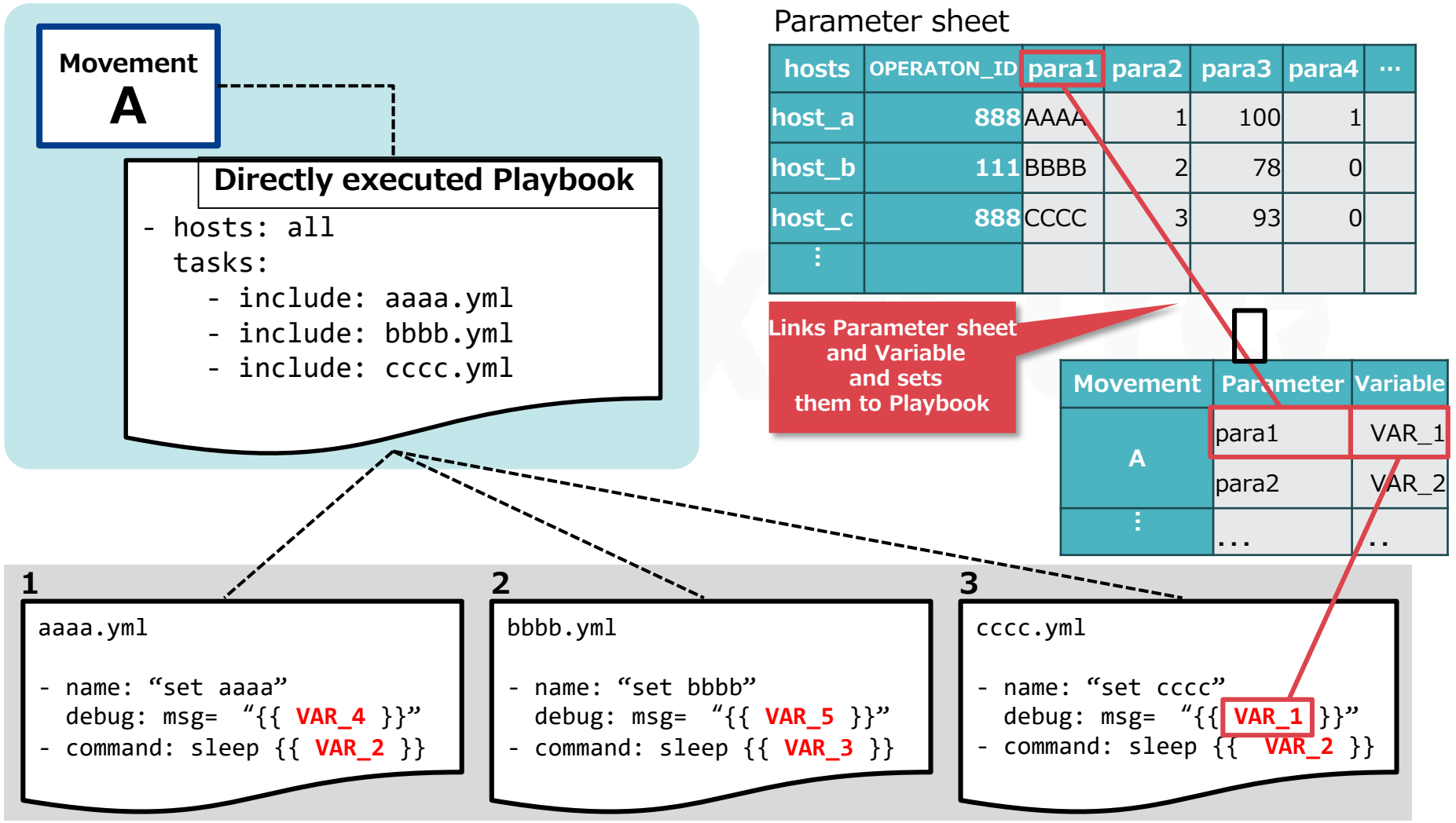
5.1 Ansible-Legacy mode (2/5)

The relation ship between the operation execution unit used in Exastro, "Movement" and Playbook are regulated on two levels.



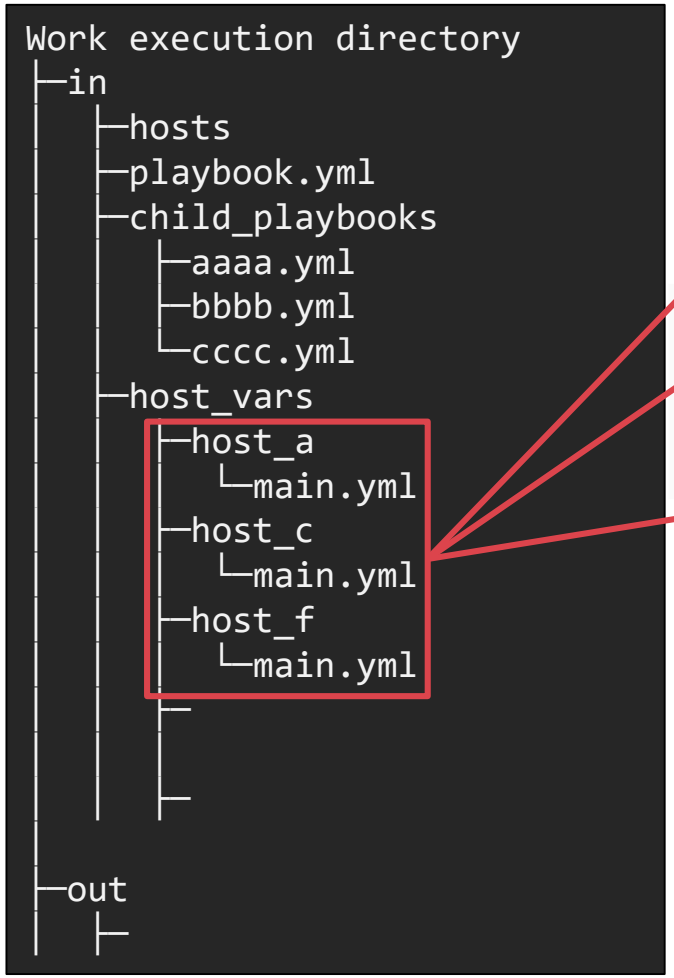
5.1 Ansible-Legacy mode (3/5)

- When executing, the parameters given to variables can be managed in the parameter sheets in Exastro ITA.



5.1 Ansible-Legacy mode (4/5)

While it isn't necessary to be aware about this when using ITA, this is what happens in the background.



Parameter sheet

hosts	OPERATON_ID	para1	para2	para3	para4	…
host_a	888	AAAA	1	100	1	
host_b	111	BBBB	2	78	0	
host_c	888	CCCC	3	93	0	
host_d	222	DDDD	10	78	0	
host_e	333	EEEE	5	84	1	
host_f	888	FFFF	4	80	0	
host_g	111	GGGG	3	90	1	
⋮						

- **hosts** : The operation target movement list. (OPERATION_ID=888)
- **playbook.yml** : Directly executed Playbook
- **child_playbooks** : Stores Playbook files
- **host_vars** : Stores all variables unique to every host defined by playbooks

※For more information about Operation ID, Please check Learn:BASE.

Menu functions explanation

- **Movement list**

  Create Movements and check them in a list.
- **Playbook files**

  Register IaC and check them in a list.
- **Movement details**

  Manage the playbooks that are going to be included in Movement.
- **Substitution value auto-registration settings**

  Link the set value of the item of registered operations and every host.
  Also possible to manage movement variables.
- **Target host**

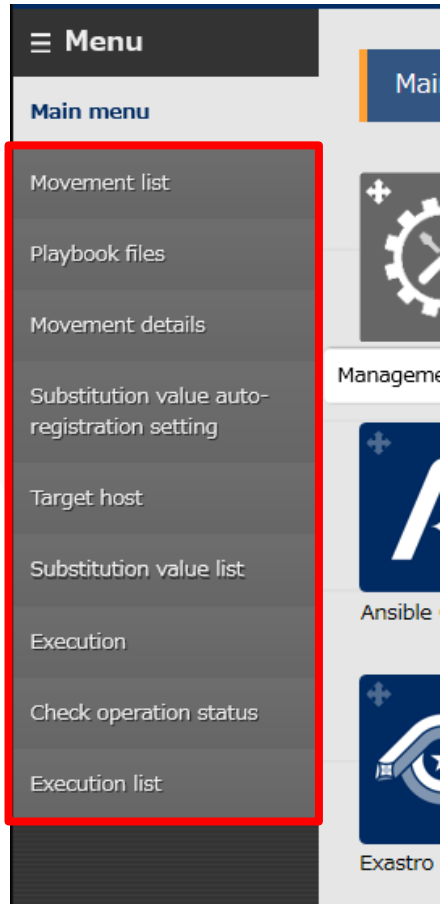
  Manage host and movements linked to Operation.
- **Substitution value list**

  Manage the substitute values for Playbooks and Variables (VAR_) used in Movements.
- **Execution**

  Execute created movements.
- **Check operation status**

  Check detailed status of executed Movement.
- **Execution list**

  Check created and executed operations and the execution history list.

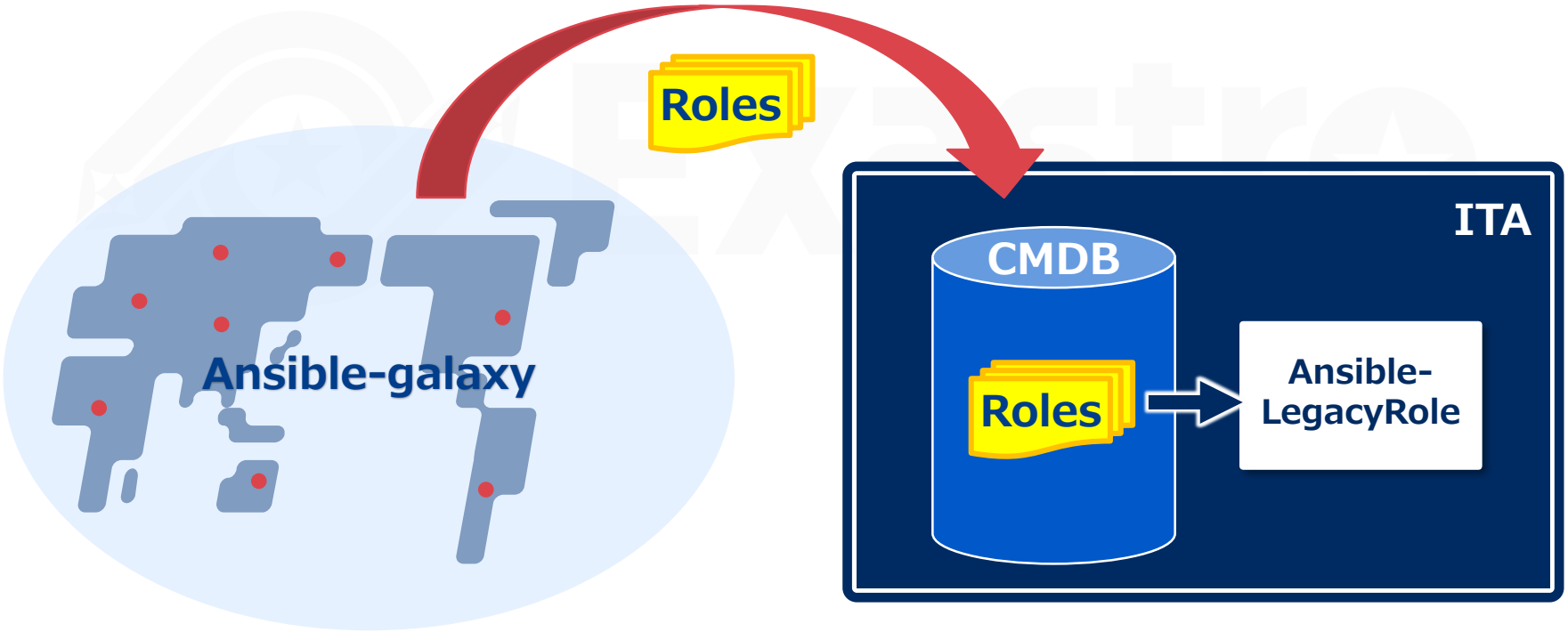


5. Features of each mode
5.2 Ansible-LegacyRole mode

5.2 Ansible-LegacyRole mode (1/4)

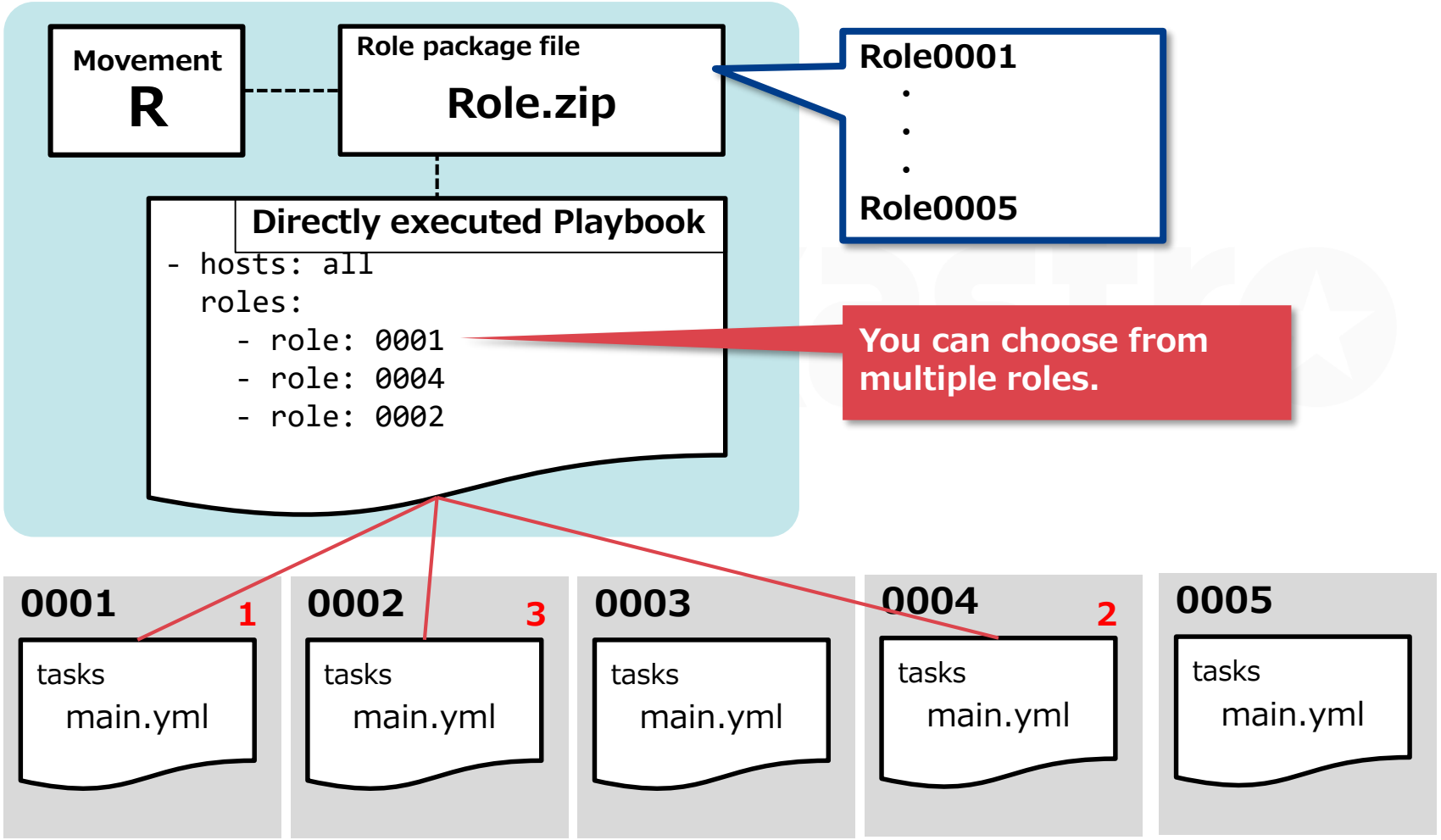
All the wisdom in the whole world – Ansible LegacyRole mode.

- The prime feature of Ansible-Legacy is to register and use role packages.
- You can use roles that you`ve created yourself or roles acquired from Ansible-Galaxy



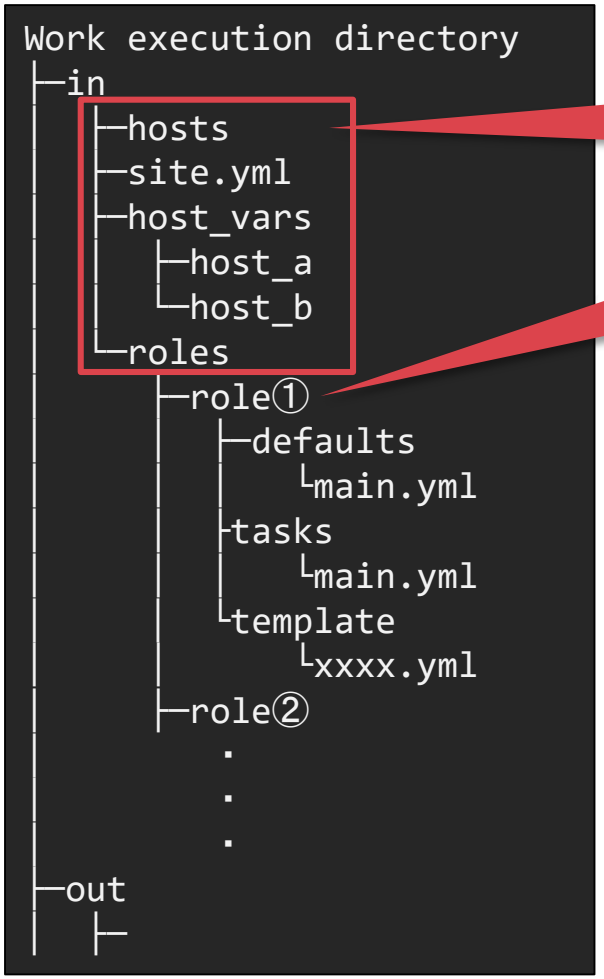
5.2 Ansible-LegacyRole mode (2/4)

Link the roles inside Role packages and Movements in ITA.



5.2 Ansible-LegacyRole mode (3/4)

- While it isn't necessary to be aware about this when using ITA, this is what happens in the background.



The role packages compresses directories that has roles in it into a zip file.

The name of the execution role directory will be stay the same , Site.yml (Directly executed playbook)

- **hosts** : Operation target host list(created by ITA)
- **site.yml** : Directly executed Playbook (created by ITA)
- **host_vars** : Stores playbooks that defines host-unique variables.
- **roles** : Stores Roles that executes playbooks.

⇒**Under each role**

- **defaults** : Lists the parameters of the variant part of playbooks.
- **tasks** : Execute playbook
- **template** : The text file used for the playbook getting executed

※The directory to the left is an example.

Menu function explanation.

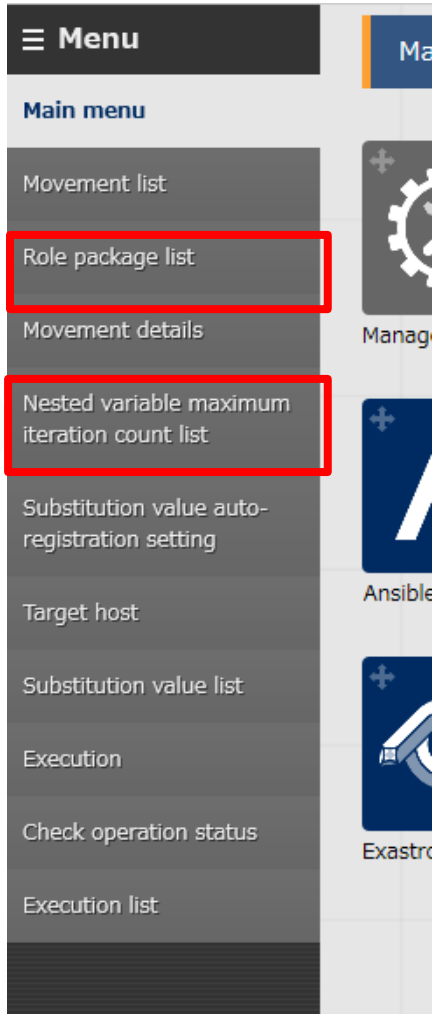
(Here we will explain the similarities it has to Ansible-Legacy mode.)

- **Role package list**

  Manage already created role package files.

- **Nested variable maximum iteration count list**

  Manage the Maximum amount of cycles the of the arrays among the multistate variables defined inside role packages.

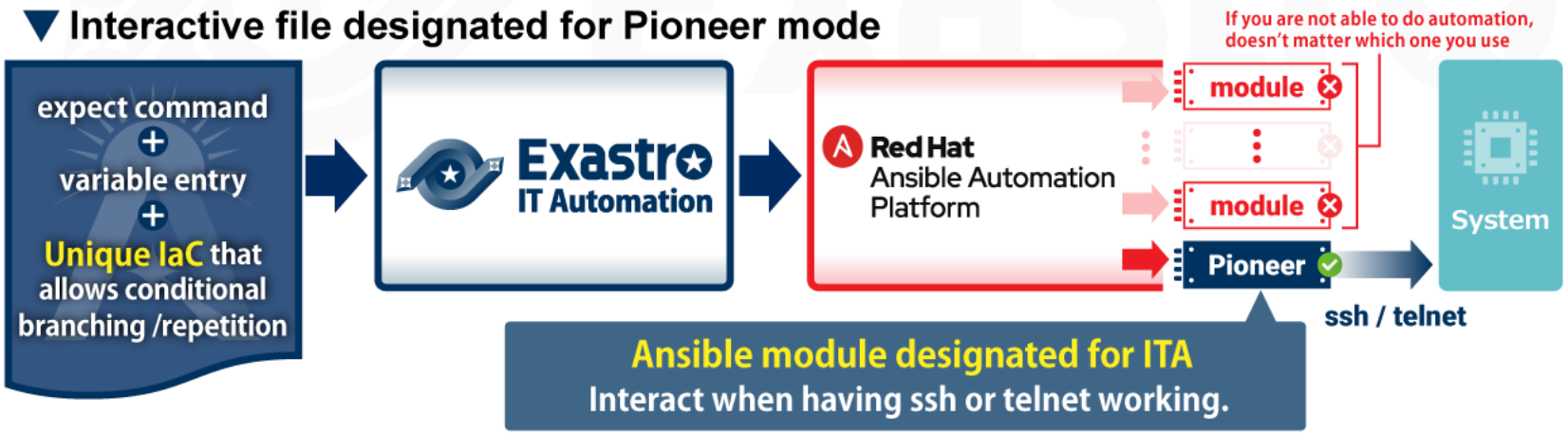


5. Features of each mode
5.3 Ansible-Pioneer mode

5.3 Ansible-Pioneer mode (1/5)

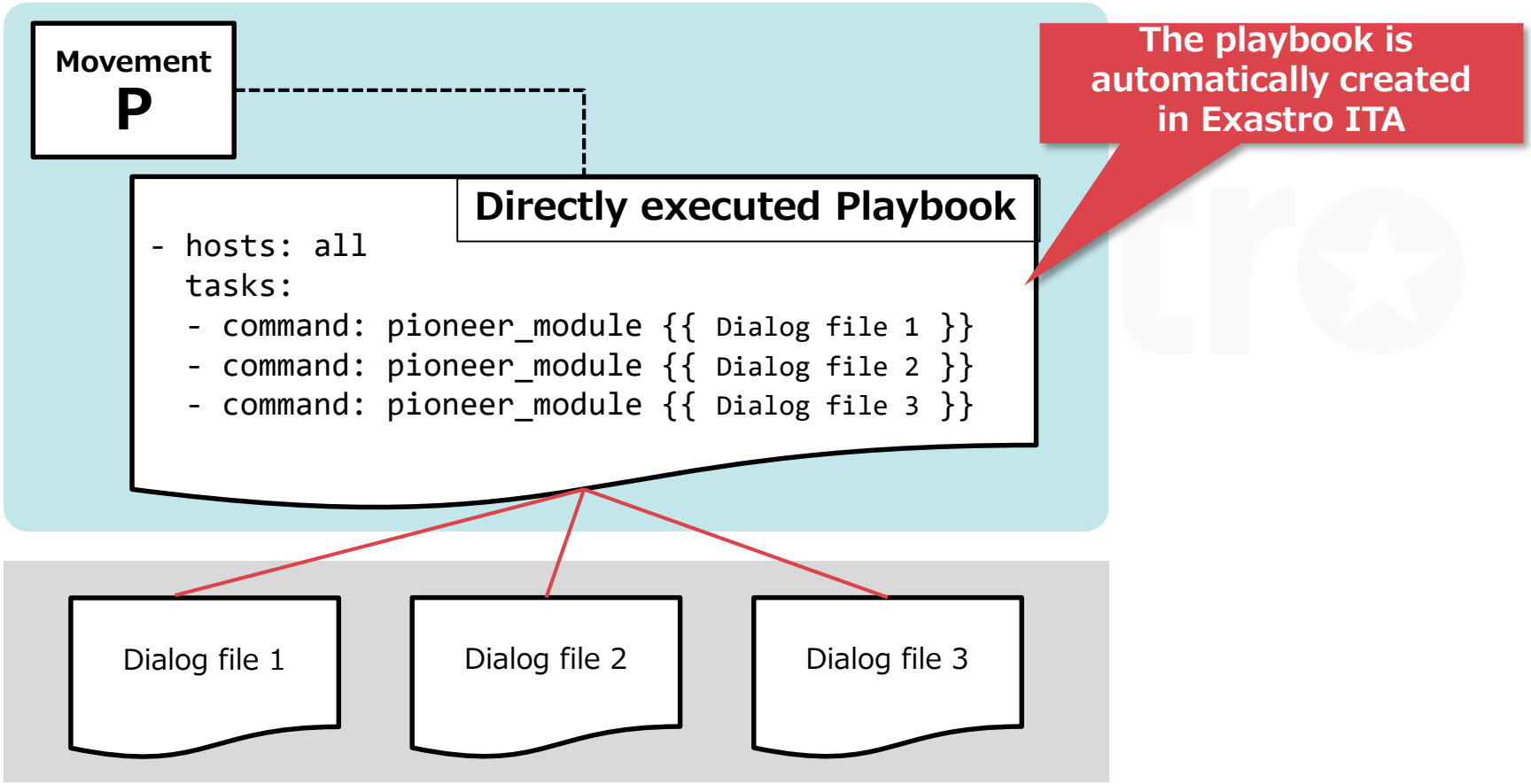
Ansible Pioneer mode – The final trump card that doesn't stop Automation

- **If users can't automate even by using all of the Ansible modules, the benefits of autimating will be cut down to half. Therefore, ITA offers Pioneer mode which acts as a trump card that doesn't stop the automation.**



5.3 Ansible-Pioneer mode (2/5)

Pioneer mode uses the Pioneer module (ITA original module)
from the directly executed Playbook to execute dialog files ※ in order.
※More about dialog files will be explained in the next slide.



5.3 Ansible-Pioneer mode (3/5)

In Ansible-Pioneer, users can write Target settings in a dialog format. Additionally, by comparing and cycling simple expect commands and using conditional processing and such, users can create high-grade dialogs.
※For more information regarding Dialog files, please refer to this manual.

Dialog file "Template" example

1. Log into the target system and check the status of the service defined by the variables.
2. If the status is "disable", post-error processing will start. If the status is anything other than "disable", the "complete" will be output to the prompt.

※The red text in the dialog file represents the variables that refers to parameter sheets.

```
01  - expect: 'password:'
02    exec: “{{ Login password }}“
03  - command: 'systemctl status {{ item.0 }}'
04    prompt: '{{ Login user }}@'
05    with_items:
06      - ‘{{ Service name 1 }}‘
07      - ‘{{ Service name 2}}‘
08    failed_when:
09      - stdout match(disable)
10  - command: 'echo complete!'
```

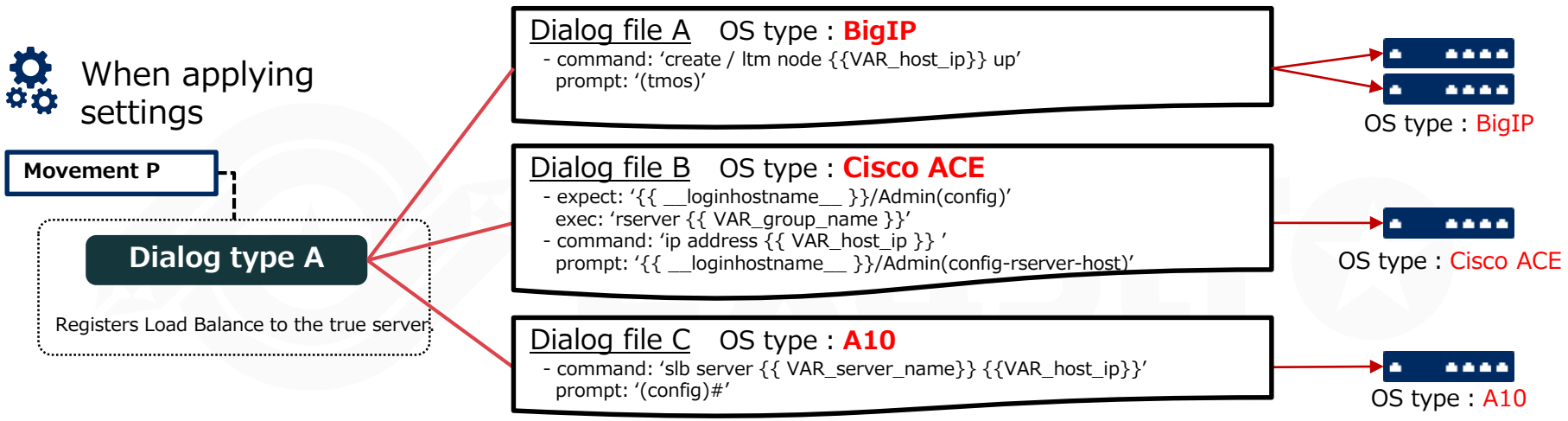
① • Iterative processing by “with_items”

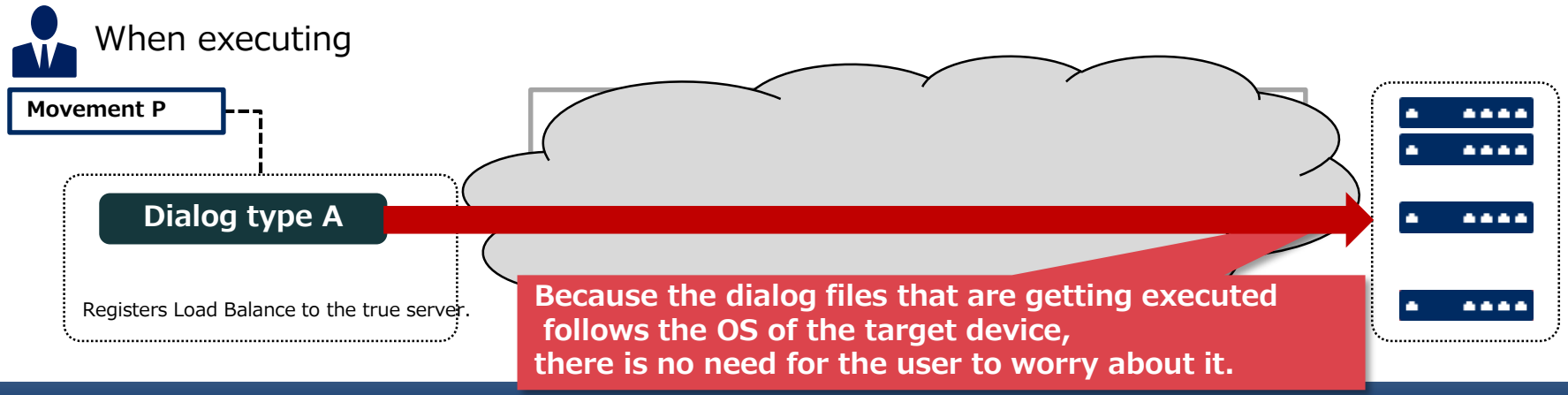
② • Branch processing by “Failed_when”

5.3 Ansible-Pioneer mode (4/5)

In Pioneer , it is possible to execute operations that doesnt get affected when there are different types of OS by setting"os type" and "Dialog type“.

- **OS Type** - Set to Dialog dile and Target device. It will come handy when you`re choosing what dialog file is getting executed.
- **Dialog type** - link dialog files with the same objective.





5.3 Ansible-Pioneer mode (5/5)

Menu function explanation

(We will now explain the similarities it has to Ansible-Legacy mode)

- **OS type master**

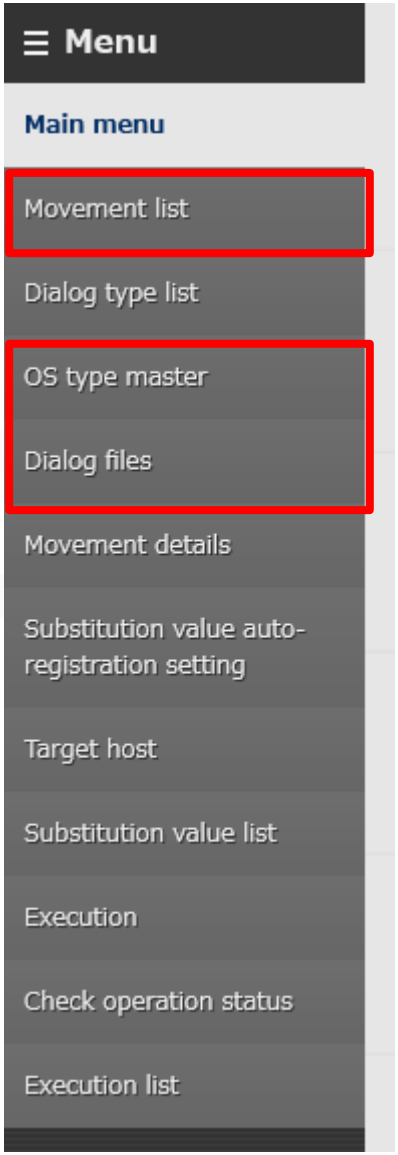
  Maintain OS types (monitor/register/update/abolish)

- **Dialog type list**

  Maintain Dialog types (monitor/register/update/abolish)

- **Dialog files**

  Manage dialog files per host




Exastro