Applying Object Oriented Programming Concepts

to Networks Security Automation

Jordy van Leersum
Automation Consulting Architect

07/12/2023

vmware Confidential | © VMware, Inc.

Agenda

- Problem Statement & Goals

- Object Oriented Programming concepts
- Understanding Cloud Services
- Defining a Framework for building a Cloud Service

- Find a problem to solve! (Network Security Automation)

Problem Statement

What are we trying to solve?

There is often a lack in the quality in the Code produced for Automation.

Leads to:

- Lack of flexibility
- Slow Time to Market
- High Resolution Time
- High Operational Effort

Most common reasons:

- Lack of Best Practices
- Lack of (Programming) Experience
  - Scripting vs. Coding
  - Developers vs. IT Experts

Scope Today: Aria Automation & NSX

Goal

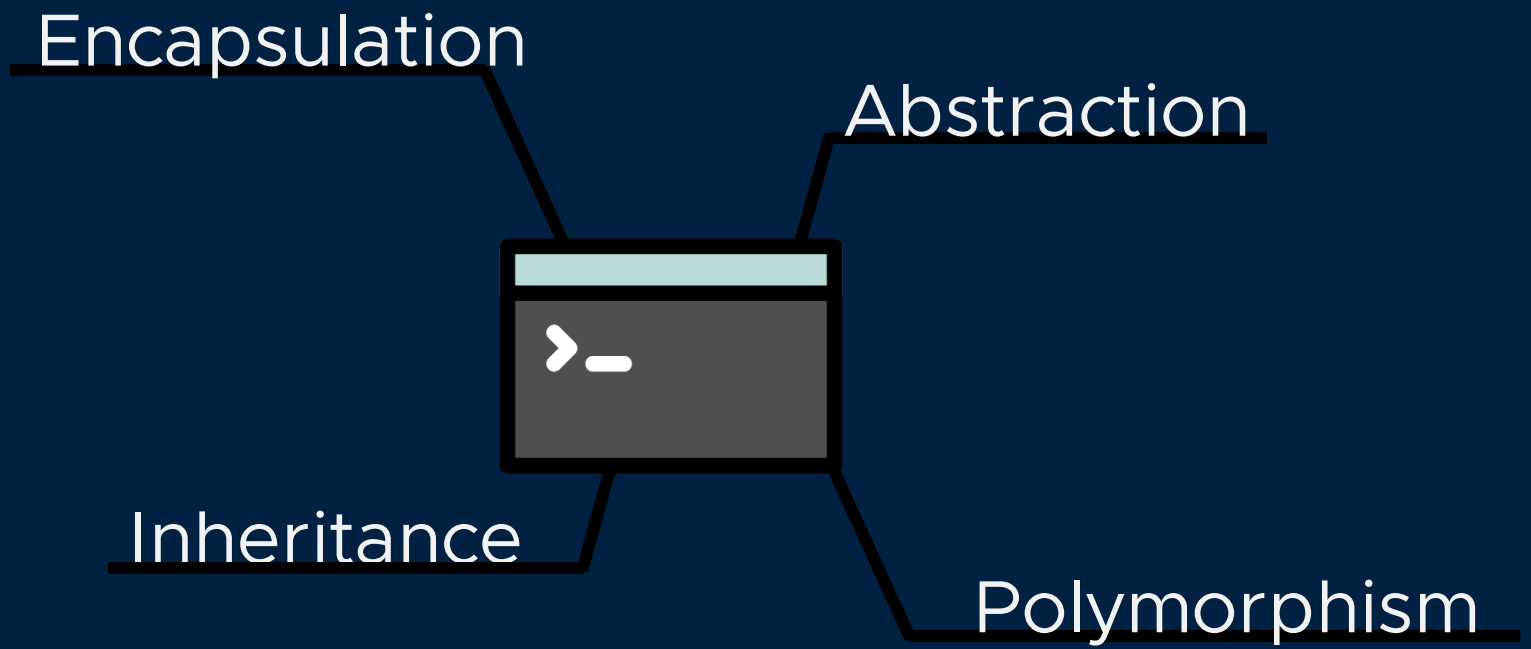
Improve Quality of the Code by applying concepts of Object Oriented Programming

Sub Goals:

- Can we make the code better maintainable?
- Can we make the code easily extendable?
- Can we make the code more modular?

Object Oriented Programming Concepts

The four pillars of Object Oriented Programming

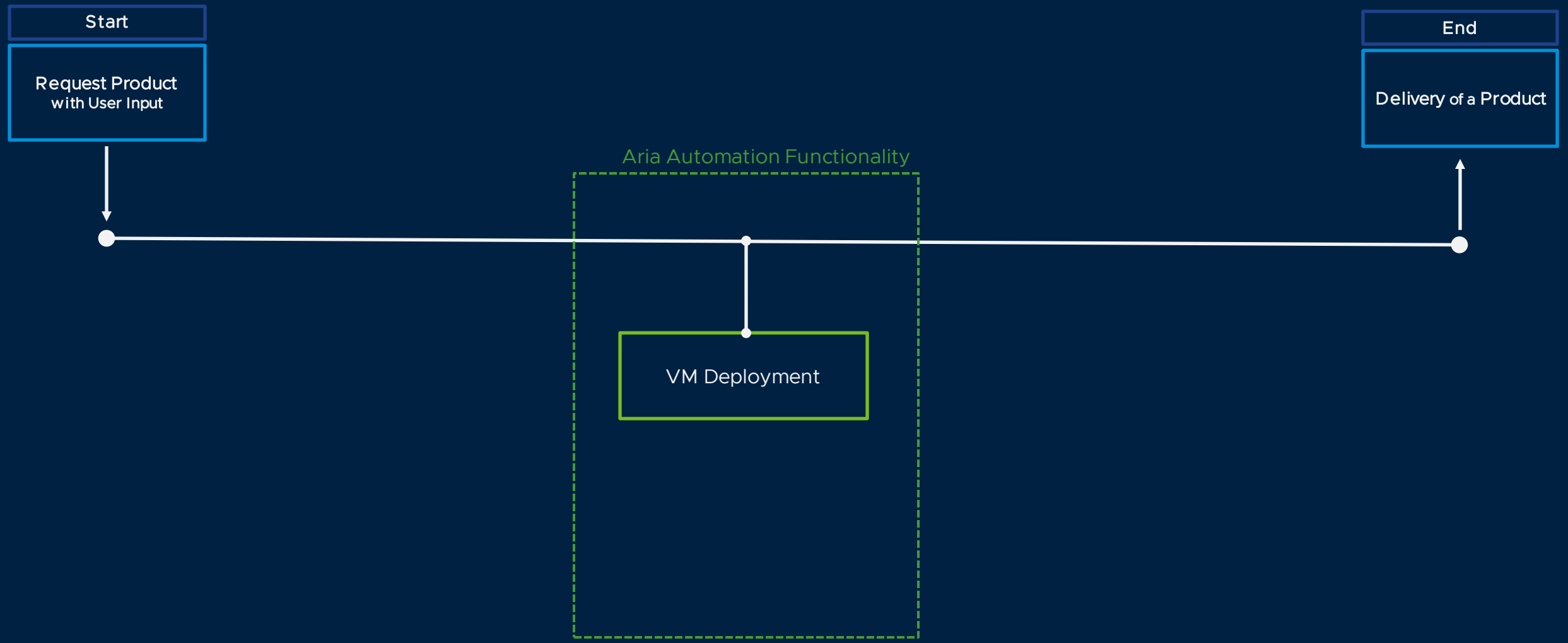


This gives us:

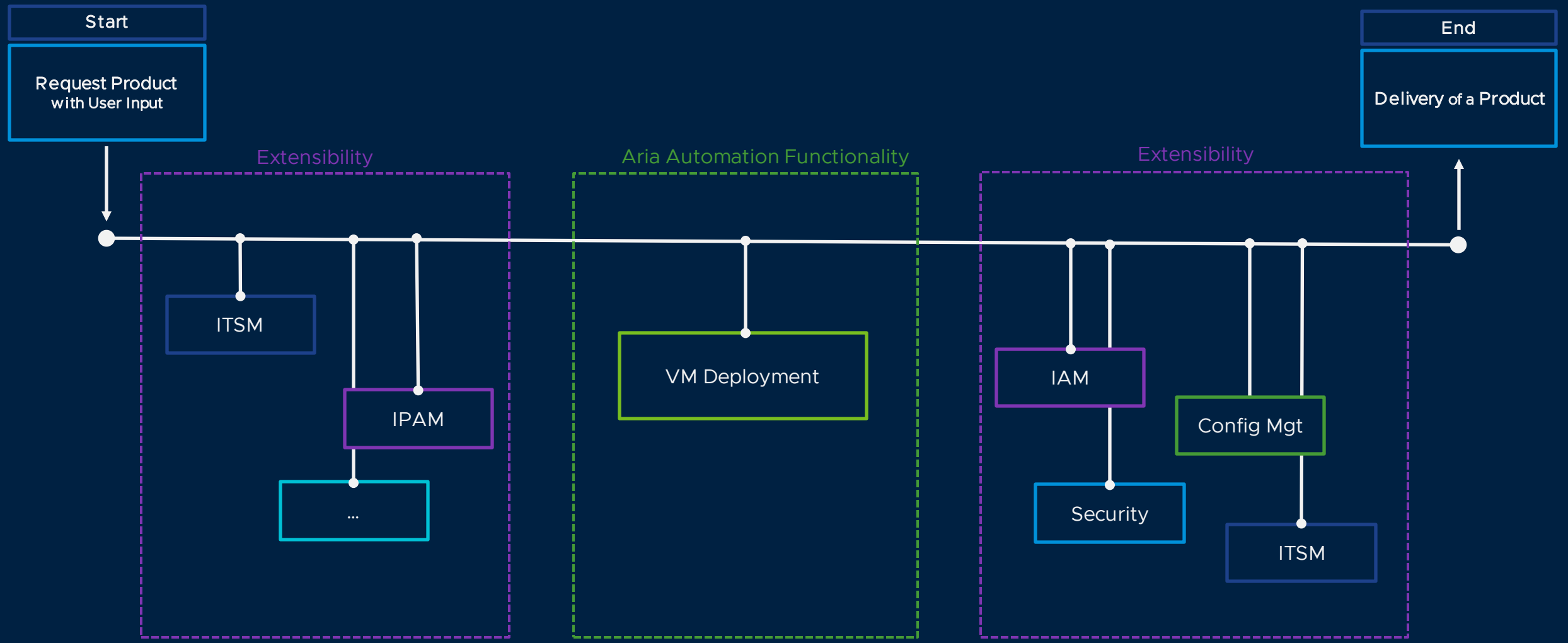
- Modularity
- Code Reusability
- Flexibility & Scalability
- Ease of Maintenance

Cloud Service

Provisioning Process



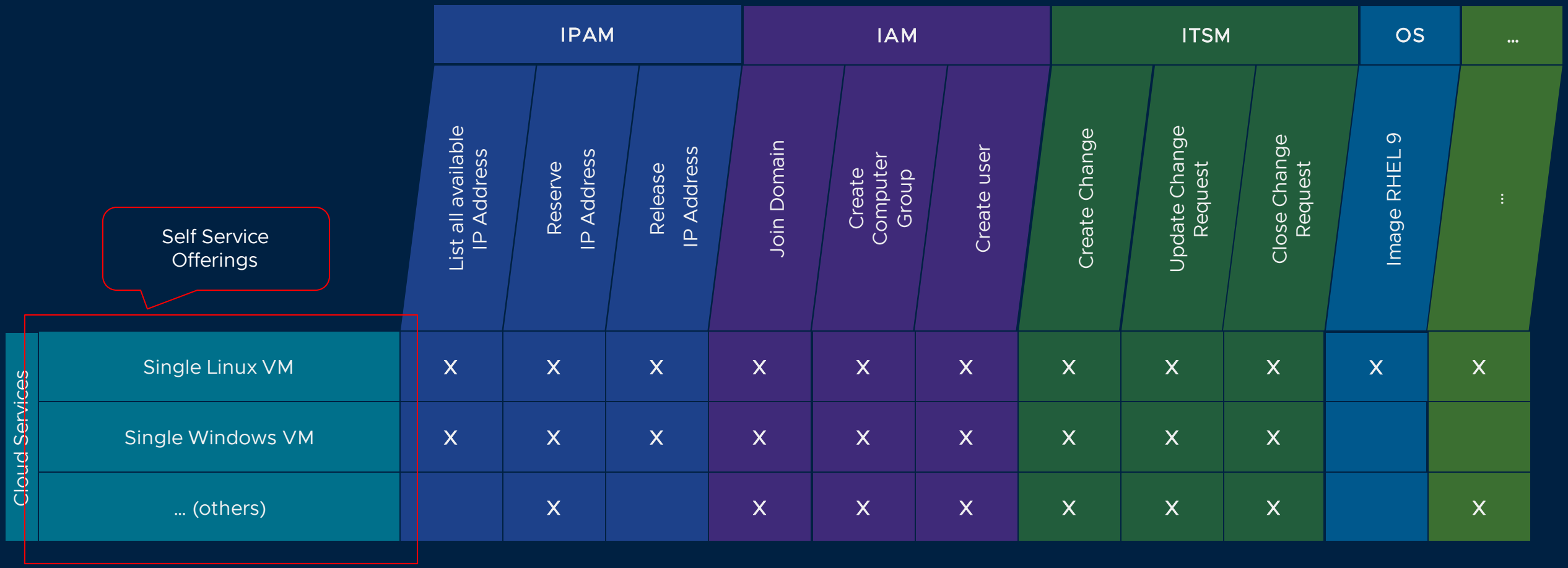
Provisioning Process



Cloud Services

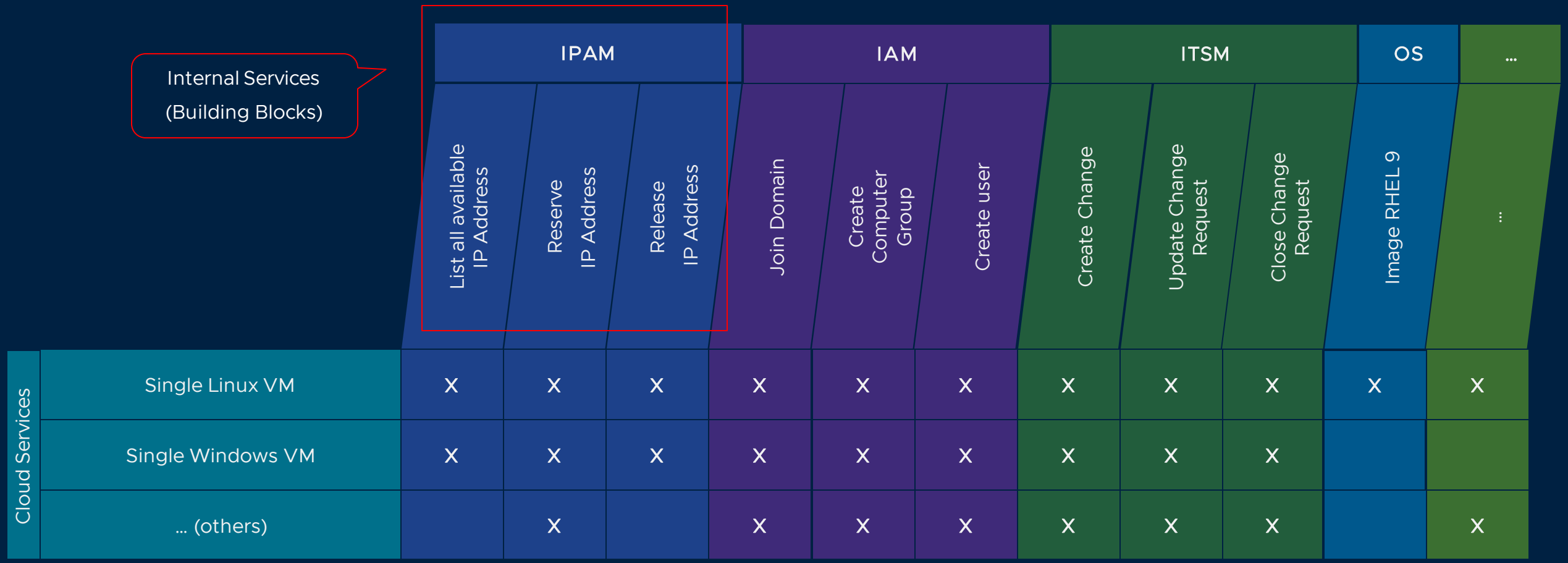
		IPAM			IAM			ITSM			OS	...
		List all available IP Address	Reserve IP Address	Release IP Address	Join Domain	Create Computer Group	Create user	Create Change	Update Change Request	Close Change Request	Image RHEL 9	...
Cloud Services	Single Linux VM	X	X	X	X	X	X	X	X	X	X	X
	Single Windows VM	X	X	X	X	X	X	X	X	X		
	... others		X		X	X	X	X	X	X		X

Cloud Services



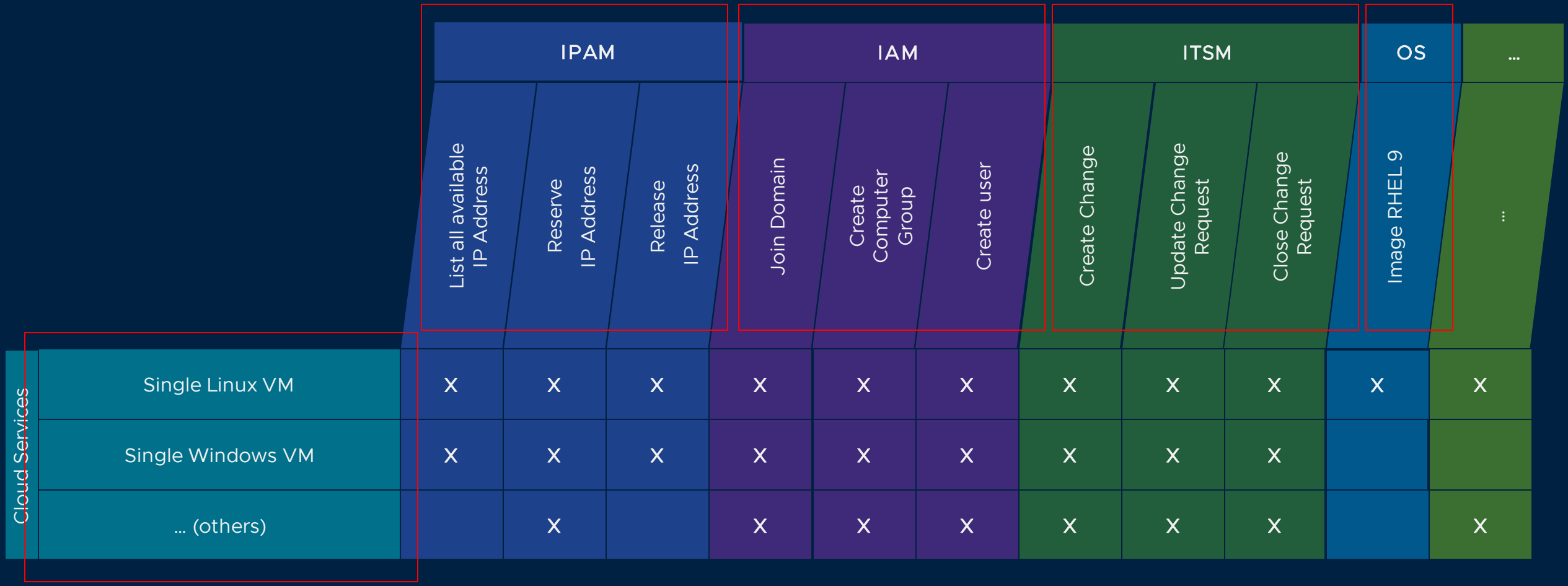
Cloud Services	List all available IP Address	Reserve IP Address	Release IP Address	Join Domain	Create Computer Group	Create user	Create Change	Update Change Request	Close Change Request	Image RHEL 9	...
Single Linux VM	X	X	X	X	X	X	X	X	X	X	X
Single Windows VM	X	X	X	X	X	X	X	X	X		
... (others)		X		X	X	X	X	X	X		X

Cloud Services



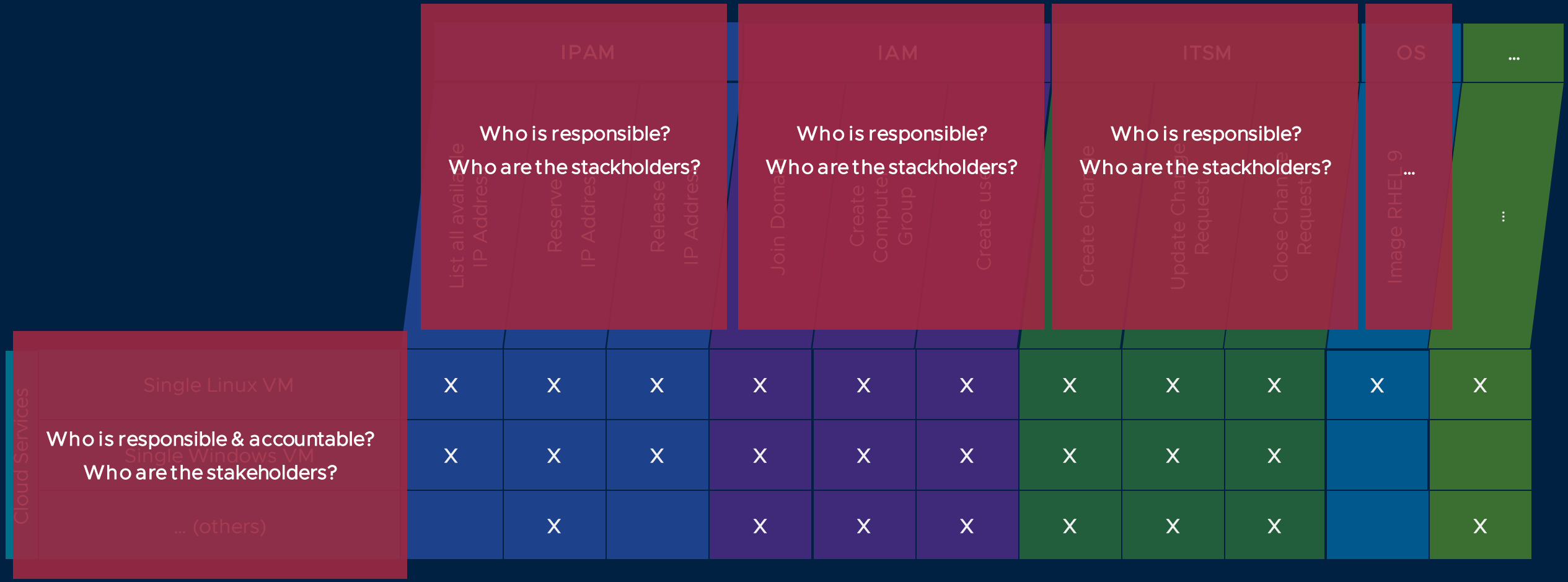
Cloud Services	IPAM: List all available IP Address	IPAM: Reserve IP Address	IPAM: Release IP Address	IAM: Join Domain	IAM: Create Computer Group	IAM: Create user	ITSM: Create Change	ITSM: Update Change Request	ITSM: Close Change Request	OS: Image RHEL 9	...: ...
Single Linux VM	X	X	X	X	X	X	X	X	X	X	X
Single Windows VM	X	X	X	X	X	X	X	X	X		
... (others)		X		X	X	X	X	X	X		X

Cloud Services



Cloud Services	IPAM			IAM			ITSM			OS	...
	List all available IP Address	Reserve IP Address	Release IP Address	Join Domain	Create Computer Group	Create user	Create Change	Update Change Request	Close Change Request	Image RHEL 9	...
Single Linux VM	X	X	X	X	X	X	X	X	X	X	X
Single Windows VM	X	X	X	X	X	X	X	X	X		
... (others)		X		X	X	X	X	X	X		X

Cloud Services



Types of Code

And their purpose

Linking Events

Business
Process Flow

Capability
which exposes
Functionality

Types of Code

And their purpose

"when the deployment is requested…"

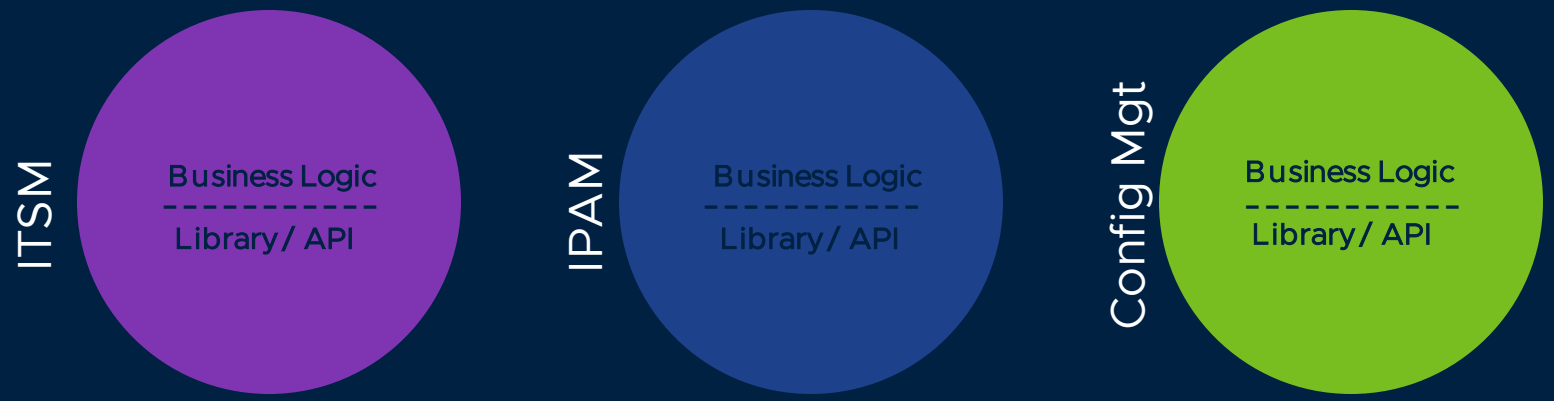
"I want to create Change Request"

Creates the Change Request

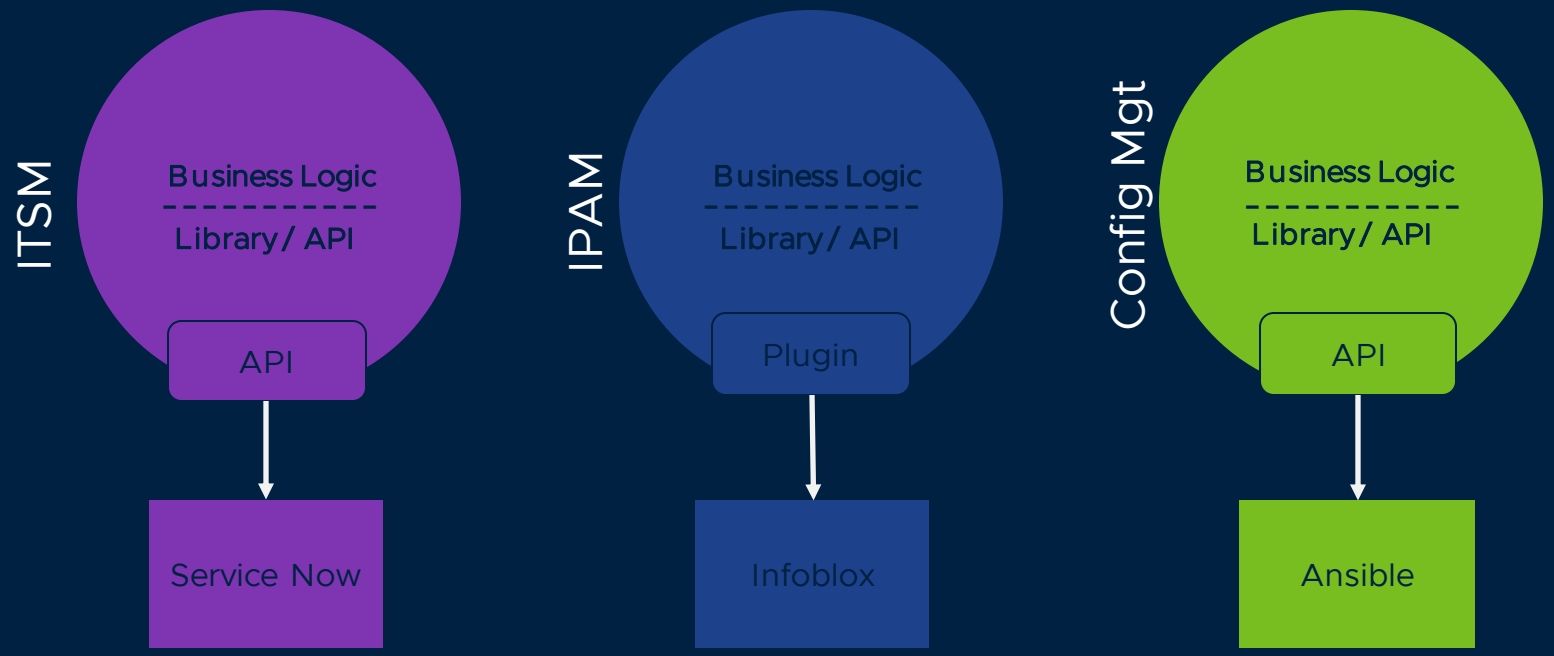
Coding Methodology

Coding Methodology

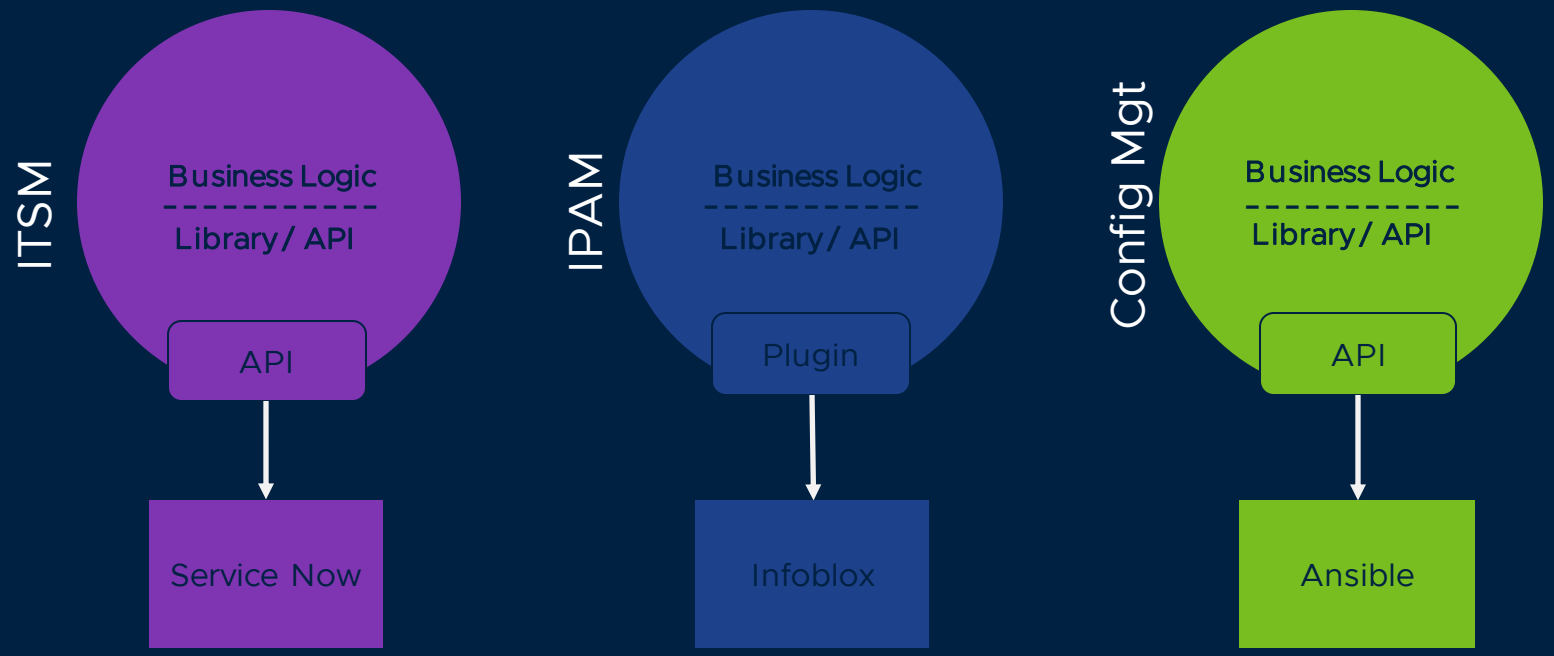
Coding Methodology



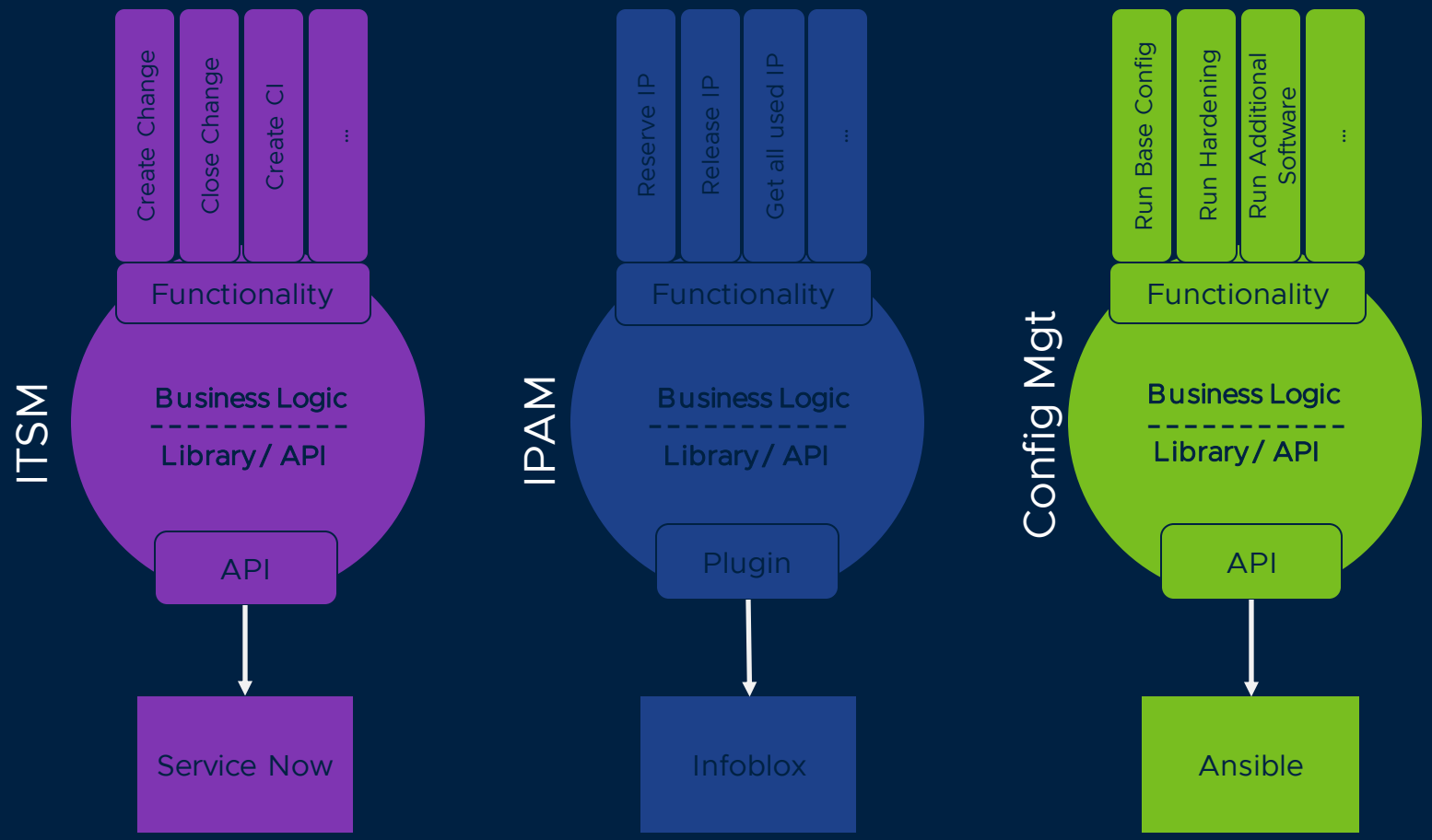
Coding Methodology



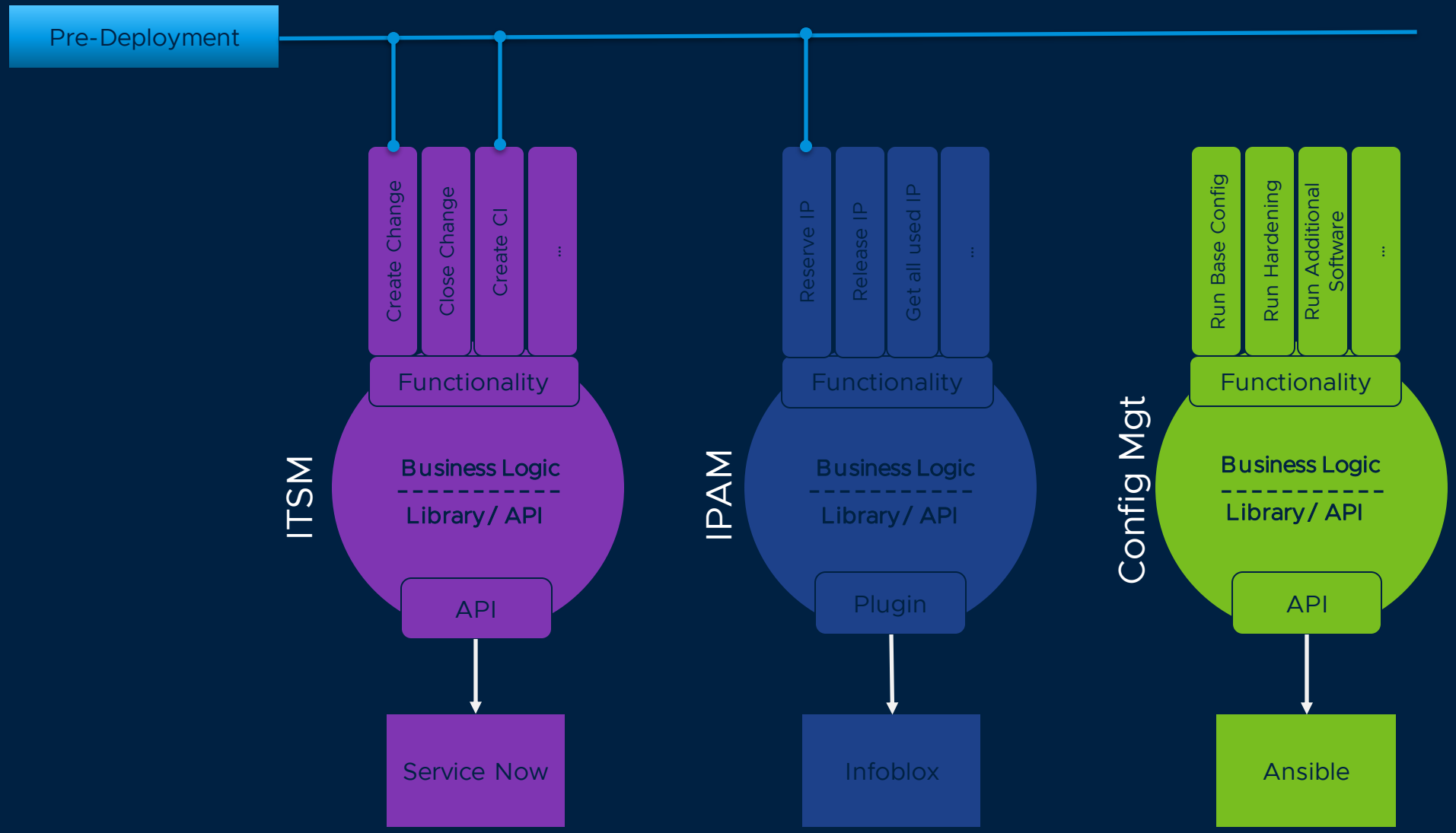
Coding Methodology



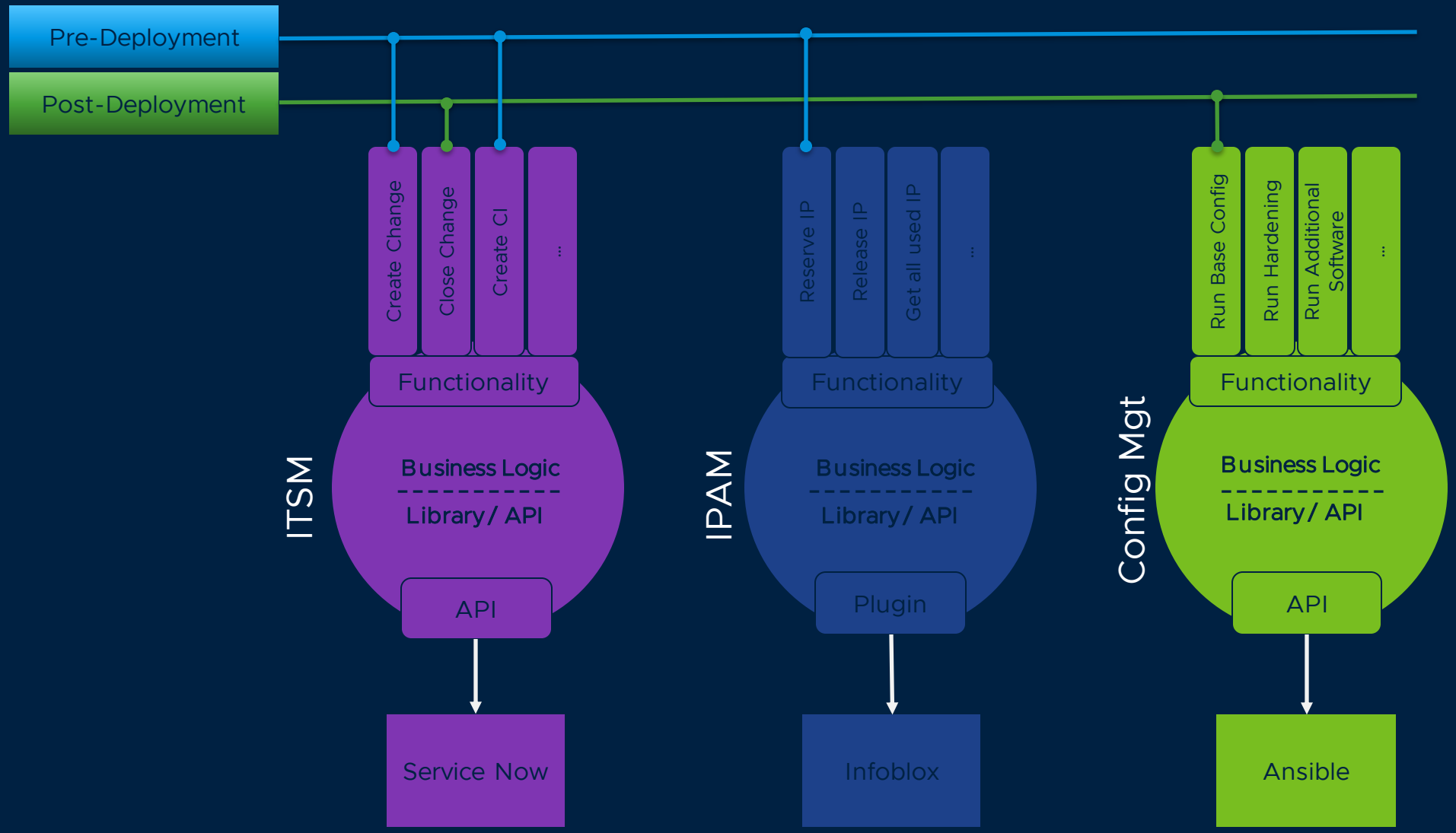
Coding Methodology



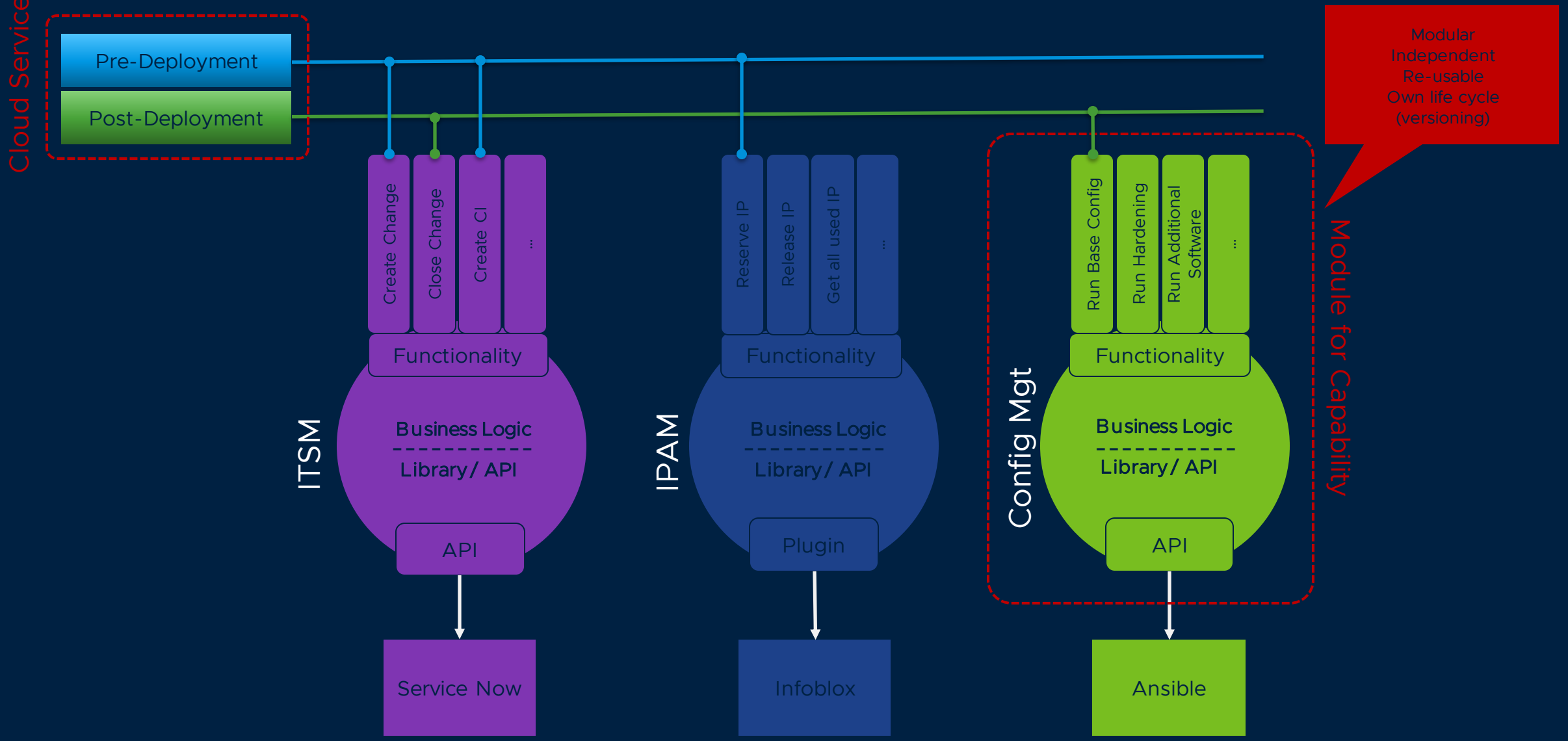
Coding Methodology



Coding Methodology



Coding Methodology

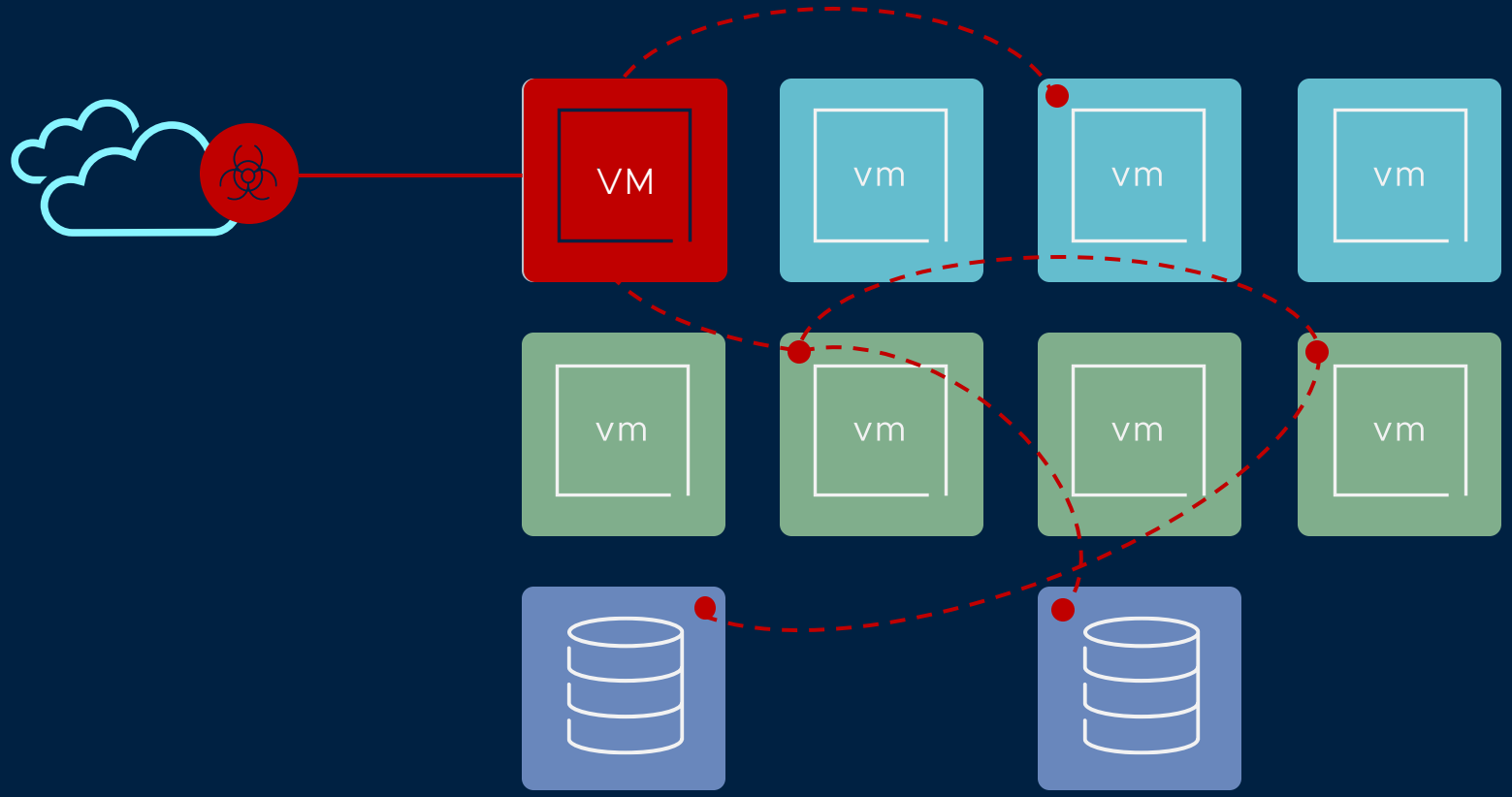


Let’s design…

Building Network Security

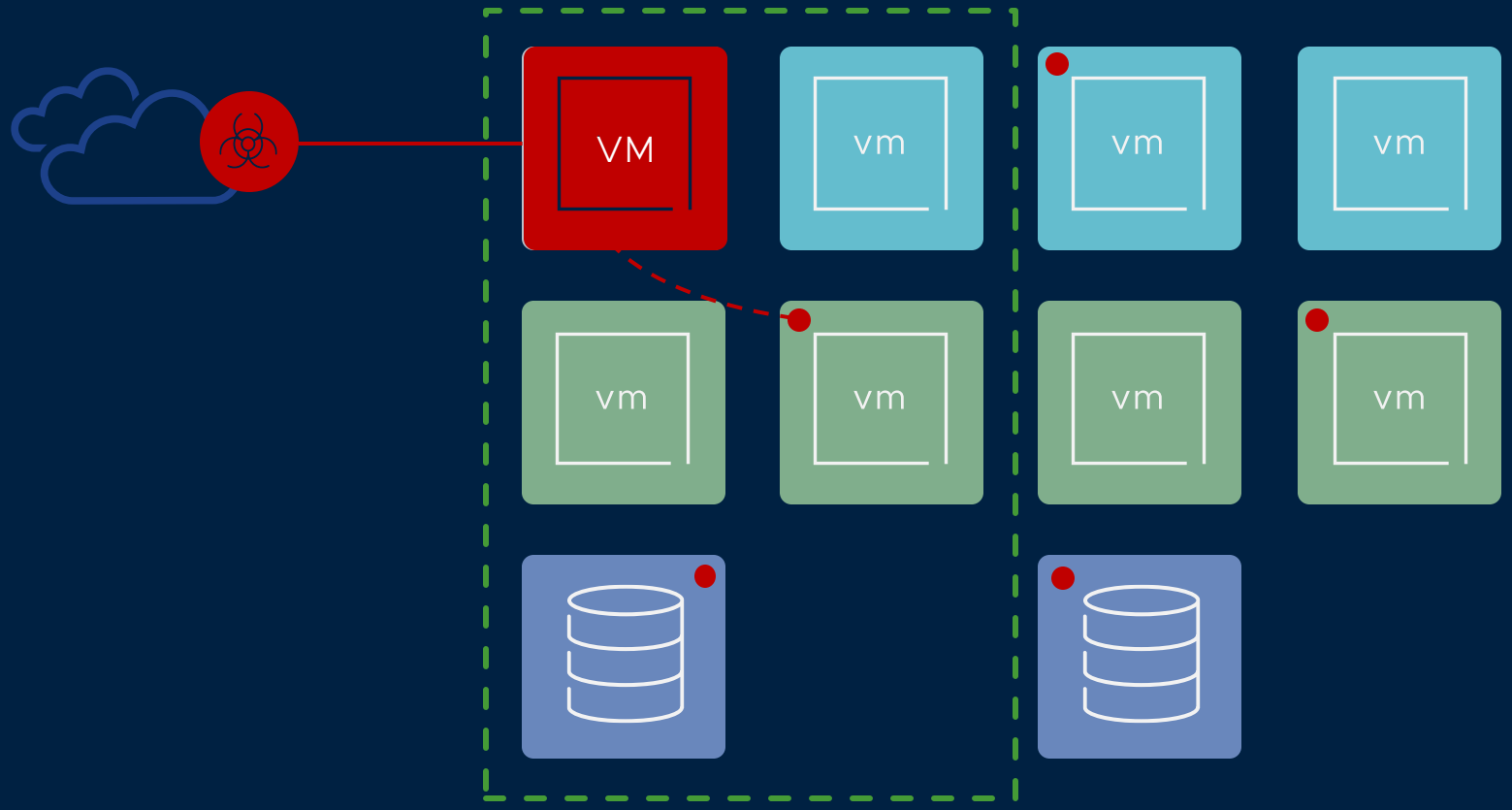
Micro-segmentation

Lateral movement of attacks is key to a successful breach



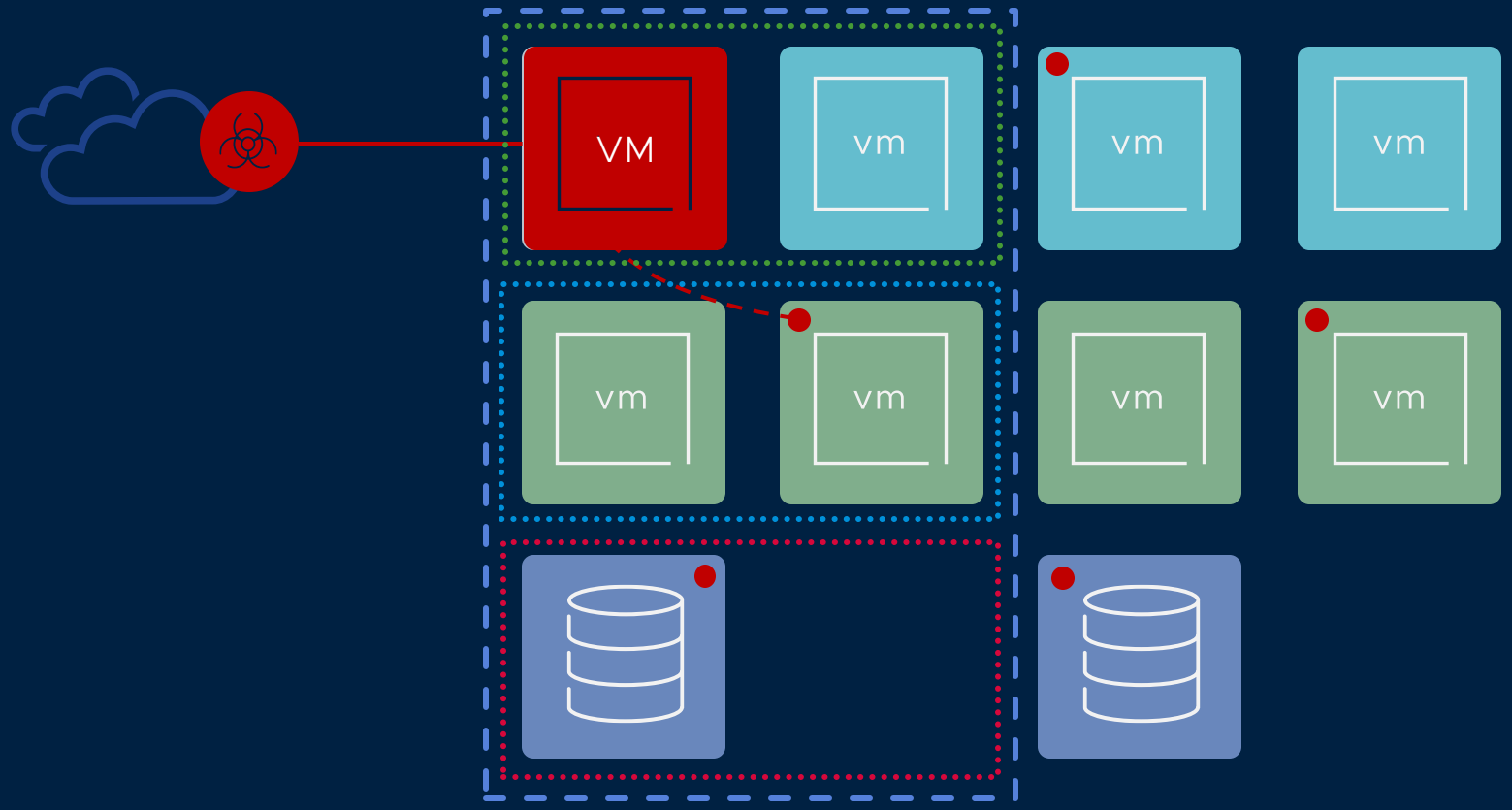
Micro-segmentation

Lateral movement of attacks is key to a successful breach

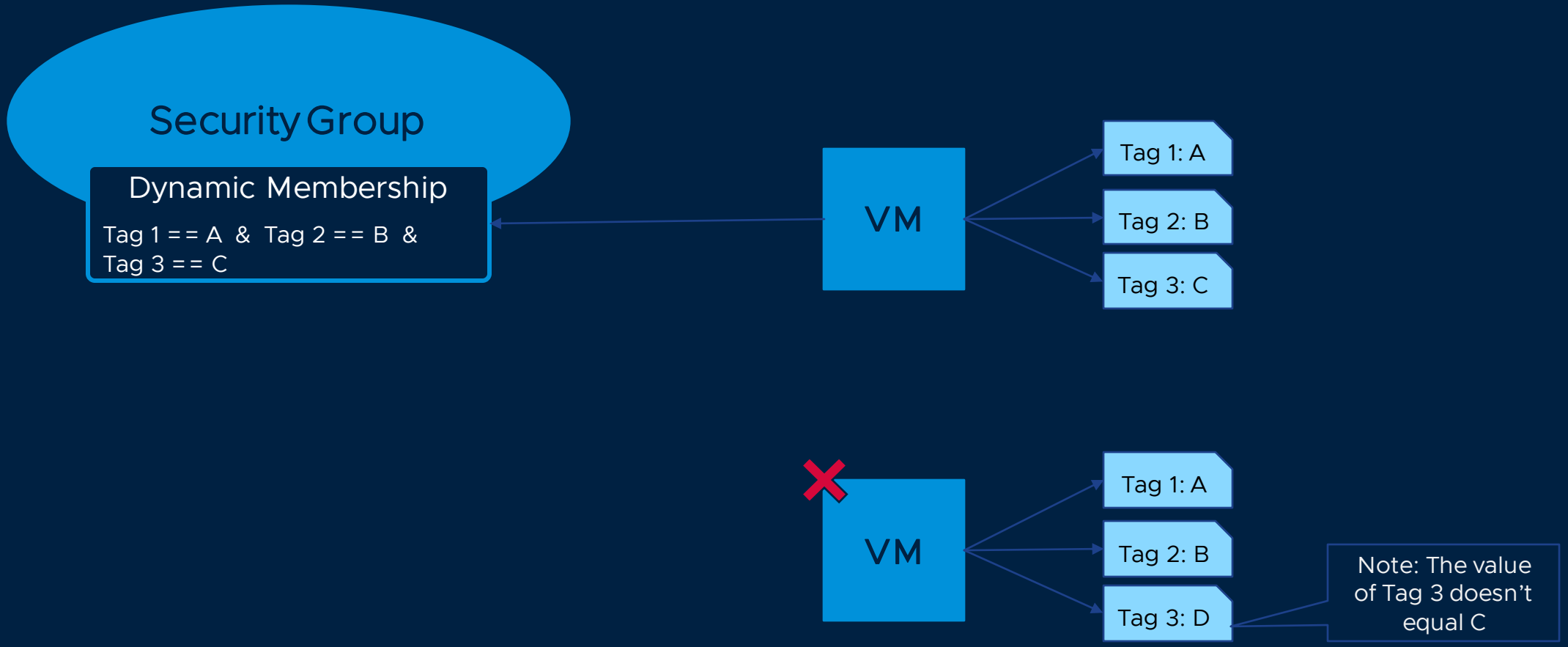


Micro-segmentation

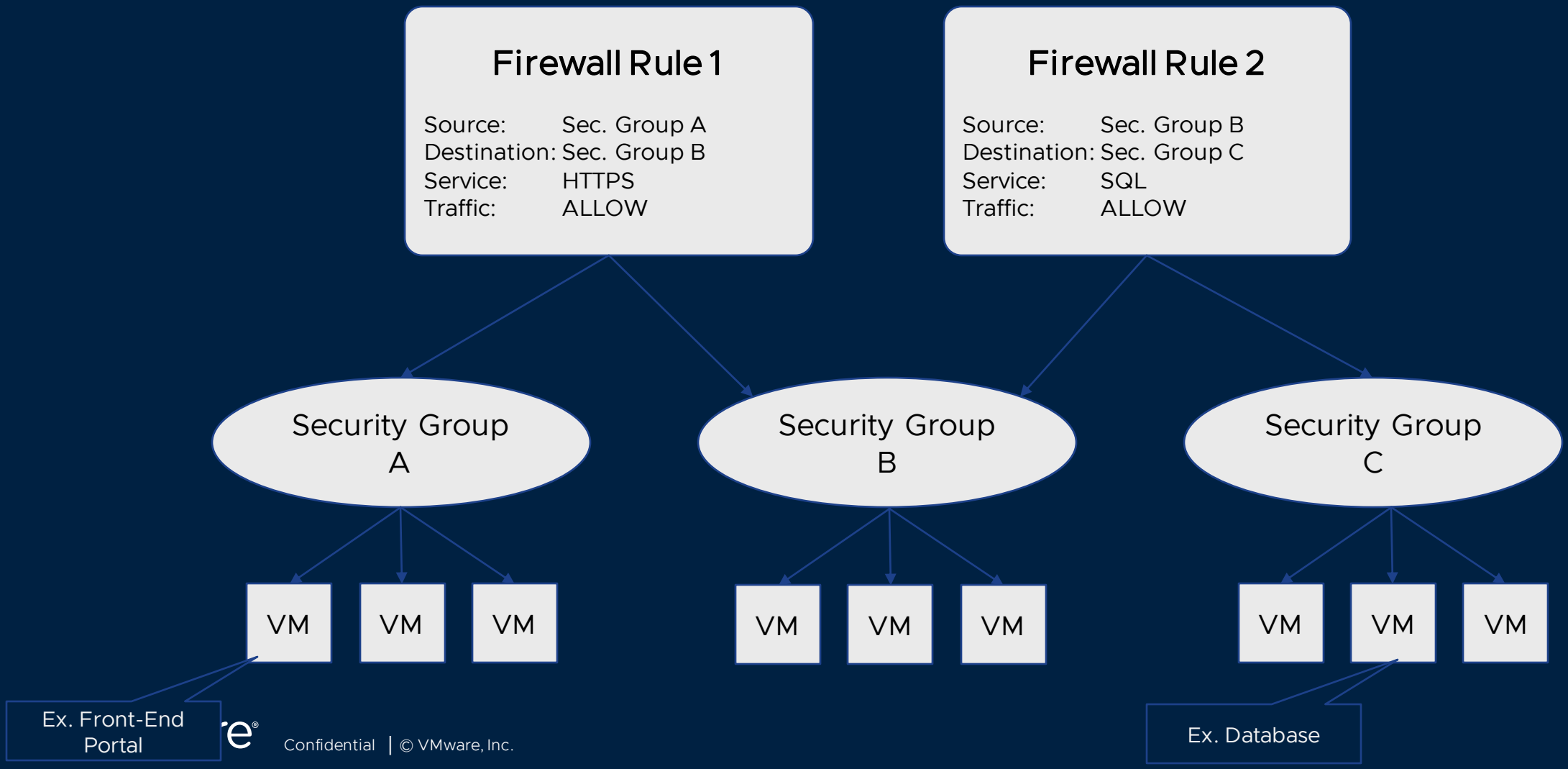
Lateral movement of attacks is key to a successful breach



Security Groups



Firewall Rules



Definitions

Security Zone

Application MyApp (Aria Automation Project)

Security Zone

- Firewall Rules:
  - SecGroup_myapp -> SecGroup_myapp : ALLOWED - HTTPS
  - SecGroup_mgt -> SecGroup_myapp : ALLOWED - HTTPS
- Security Groups:
  - New *SecGroup_myapp* with dynamic membership of *tag_app_name == myapp*

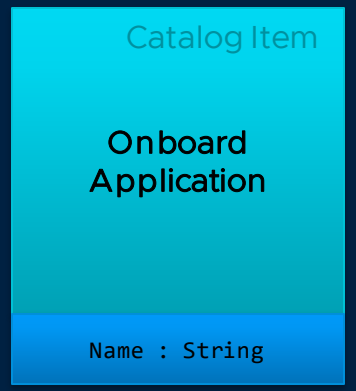
* We assume we already have a blocking firewall rule for all not approved traffic and a security group SecGroup_mgt predefined.

Use-Cases

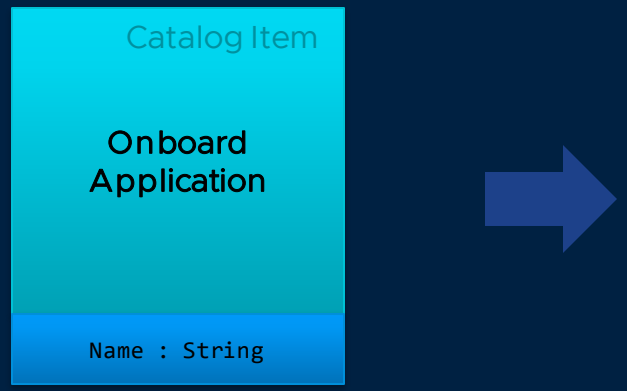
In order to achieve network segmentation we create the following use-cases:

1. Create Security Zone during Onboarding (Project Creation)
2. Add VM to the Security Zone of the Project
3. Remove VM from the Security Zone of the Project

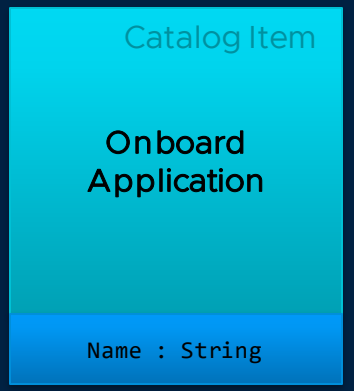
Use Case: Onboard Application



Use Case: Onboard Application



Use Case: Onboard Application



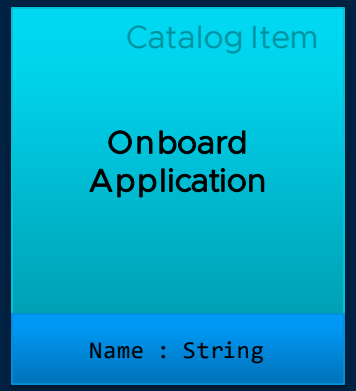
Workflow

```
OnboardApplication(name){
            // Create Change for Onboarding New Application
            var ChangeMgr = ITSM.ChangeManager(name);
            var requestId = ChangeMgr.createApplicationChange();

            // Creating the Project in Aria Automation
            var ProjectMgr = Aria.ProjectManager();
            ProjectMgr.onboardApplication(name, requestId);

            // Closing the Change
            ChangeMgr.closeApplicationChange(requestId);
}
```

Use Case: Onboard Application



Workflow

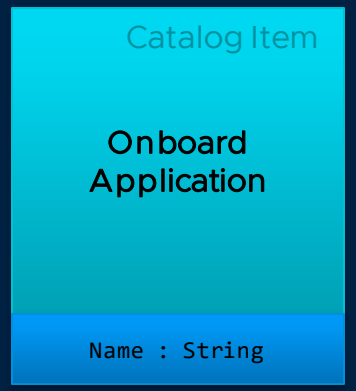
```
OnboardApplication(name){
            // Create Change for Onboarding New Application
            var ChangeMgr = ITSM.ChangeManager(name);
            var requestId = ChangeMgr.createApplicationChange();

            // Creating the Project in Aria Automation
            var ProjectMgr = Aria.ProjectManager();
            ProjectMgr.onboardApplication(name, requestId);

            // Create a Security for the new Application in NSX
            var SecZoneMgr = NSX.SecZoneMgr();
            SecZoneMgr.createApplicationSecZone(name);

            // Closing the Change
            ChangeMgr.closeApplicationChange(requestId);
}
```

Use Case: Onboard Application



Workflow

```
OnboardApplication(name){
        // Create Change for Onboarding New Application
        var ChangeMgr = ITSM.ChangeManager(name);
        var requestId = ChangeMgr.createApplicationChange();

        // Creating the Project in Aria Automation
        var ProjectMgr = Aria.ProjectManager();
        ProjectMgr.onboardApplication(name, requestId);

        // Create a Security for the new Application in NSX
        var SecZoneMgr = NSX.SecZoneMgr();
        SecZoneMgr.createApplicationSecZone(name);

        // Closing the Change
        ChangeMgr.closeApplicationChange(requestId);
}
```

Use Case: Onboard Application

```
SecurityZoneManager(){
  createApplicationSecZone(name){
          var sgMgr = NSX.SecurityGroupManager();
          var sgApp = sgMgr.createSecurityGroup(name);
          var sgMgt = sgMgr.getManagementSecurityGroup();

          var fwMgr = NSX.FirewallManager();
          fw.createFirewallRule(sgApp, sgApp, allow, https);
          fw.createFirewallRule(sgMgt, sgApp, allow, https);
  }
}
```

Use Case: Onboard Application

```
SecurityZoneManager(){
  createApplicationSecZone(name){
          var sgMgr = NSX.SecurityGroupManager();
          var sgApp = sgMgr.createSecurityGroup(name);
          var sgMgt = sgMgr.getManagementSecurityGroup();

          var fwMgr = NSX.FirewallManager();
          fw.createFirewallRule(sgApp, sgApp, allow, https);
          fw.createFirewallRule(sgMgt, sgApp, allow, https);
  }
}
```

```
SecurityGroupManager(){
  createSecurityGroup(name){
    // API call..
  }
}
```

```
FirewallRuleManager(){
  createFirewallRule(sgA, sgB, allowed, traffic){
    // API call..
  }
}
```

Use Case: Onboard Application

```
SecurityZoneManager(){
  createApplicationSecZone(name){
          var sgMgr = NSX.SecurityGroupManager();
          var sgApp = sgMgr.createSecurityGroup(name);
          var sgMgt = sgMgr.getManagementSecurityGroup();

          var fwMgr = NSX.FirewallManager();
          fw.createFirewallRule(sgApp, sgApp, allow, https);
          fw.createFirewallRule(sgMgt, sgApp, allow, https);
  }
}
```

```
SecurityGroupManager(){
  createSecurityGroup(name){
    // API call..
  }
}
```

```
FirewallRuleManager(){
  createFirewallRule(sgA, sgB, allowed, traffic){
    // API call..
  }
}
```

```
SessionManager(){
  getSessionToken(){
    // API call..
  }
}
```

Use Case: Onboard Application

External

```
SecurityZoneManager(){
  createApplicationSecZone(name){
          var sgMgr = NSX.SecurityGroupManager();
          var sgApp = sgMgr.createSecurityGroup(name);
          var sgMgt = sgMgr.getManagementSecurityGroup();

          var fwMgr = NSX.FirewallManager();
          fw.createFirewallRule(sgApp, sgApp, allow, https);
          fw.createFirewallRule(sgMgt, sgApp, allow, https);
  }
}
```

Internal

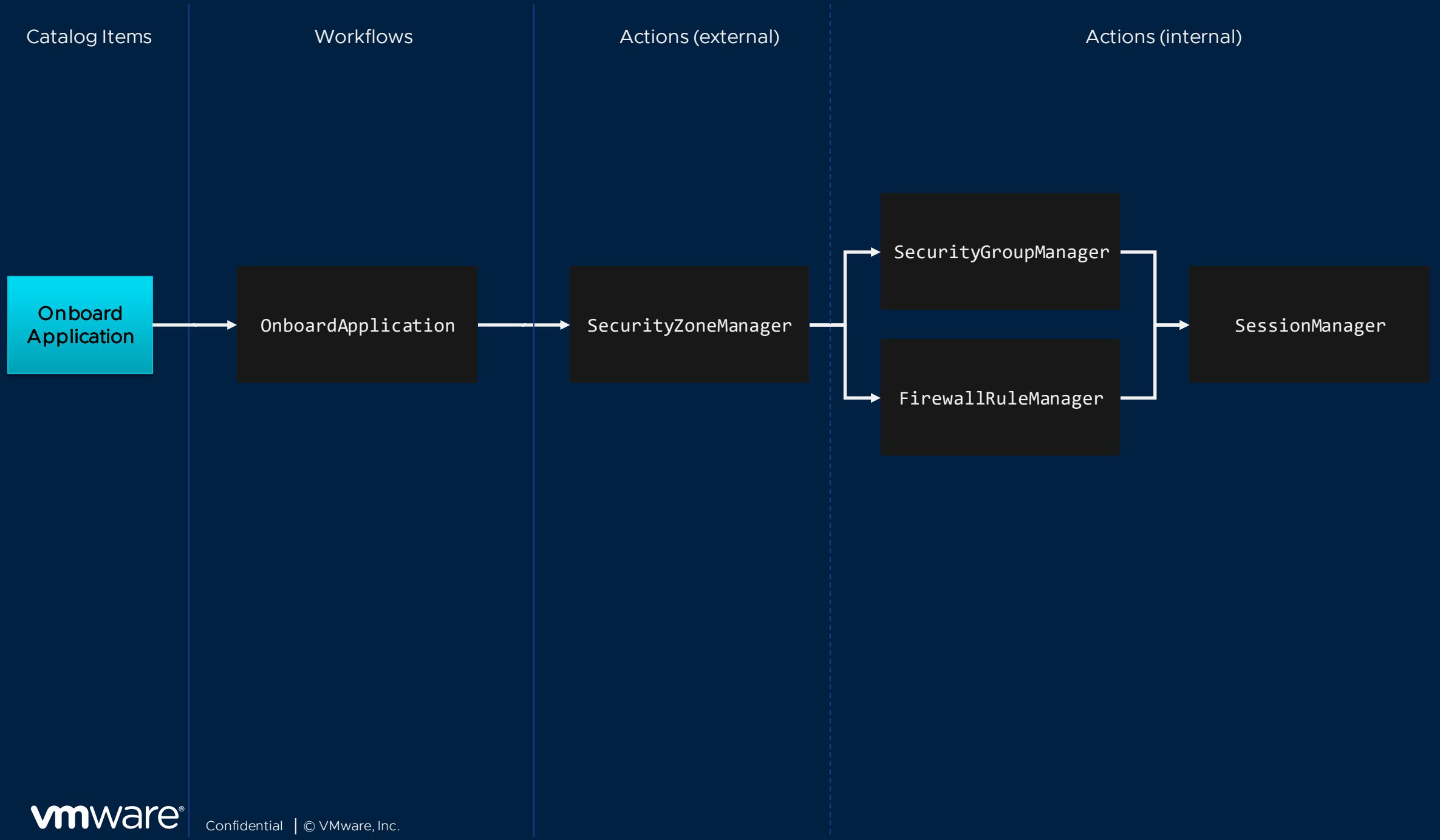
```
SecurityGroupManager(){
  createSecurityGroup(name){
    // API call..
  }
}
```

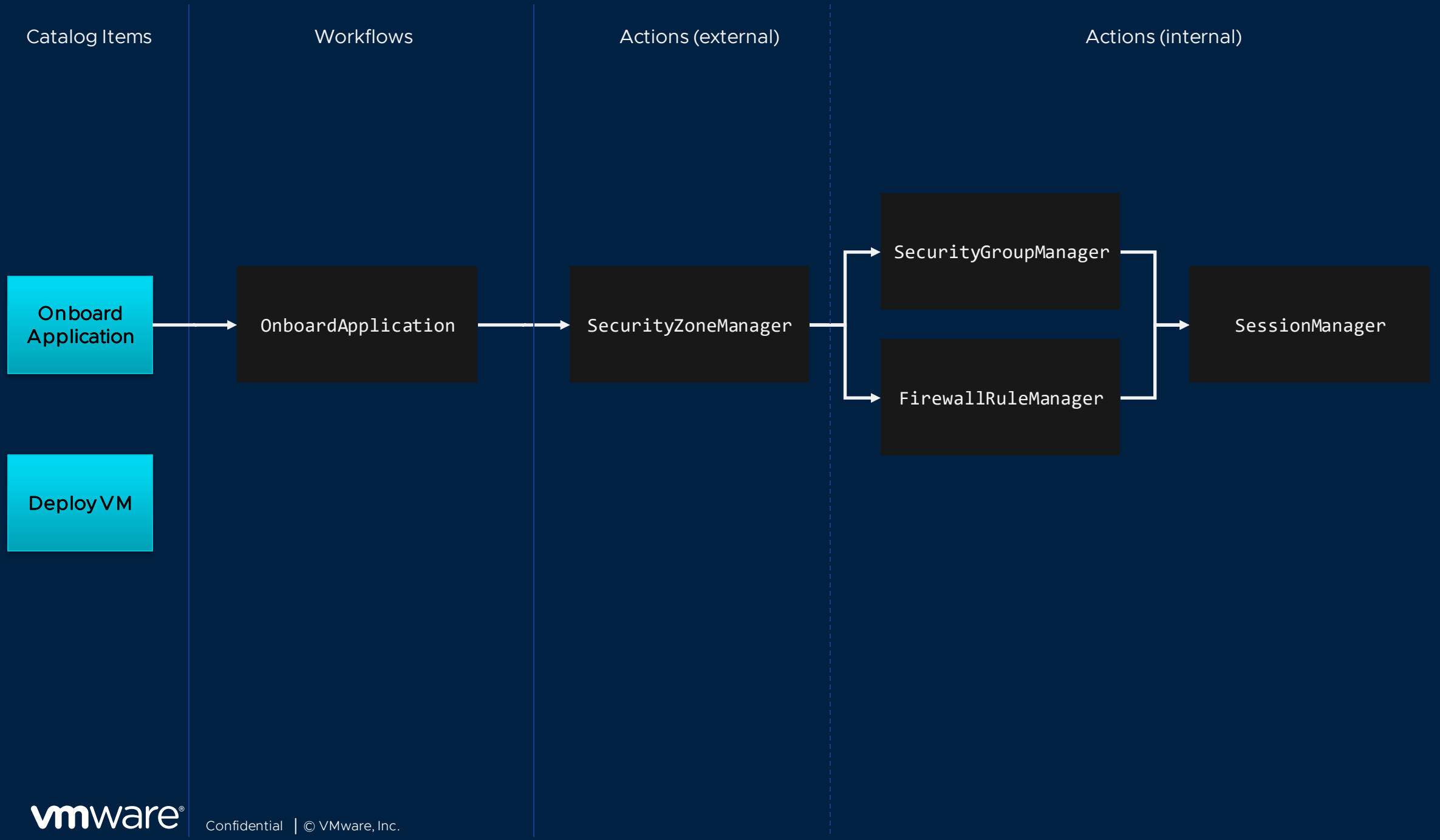
```
FirewallRuleManager(){
  createFirewallRule(sgA, sgB, allowed, traffic){
    // API call..
  }
}
```

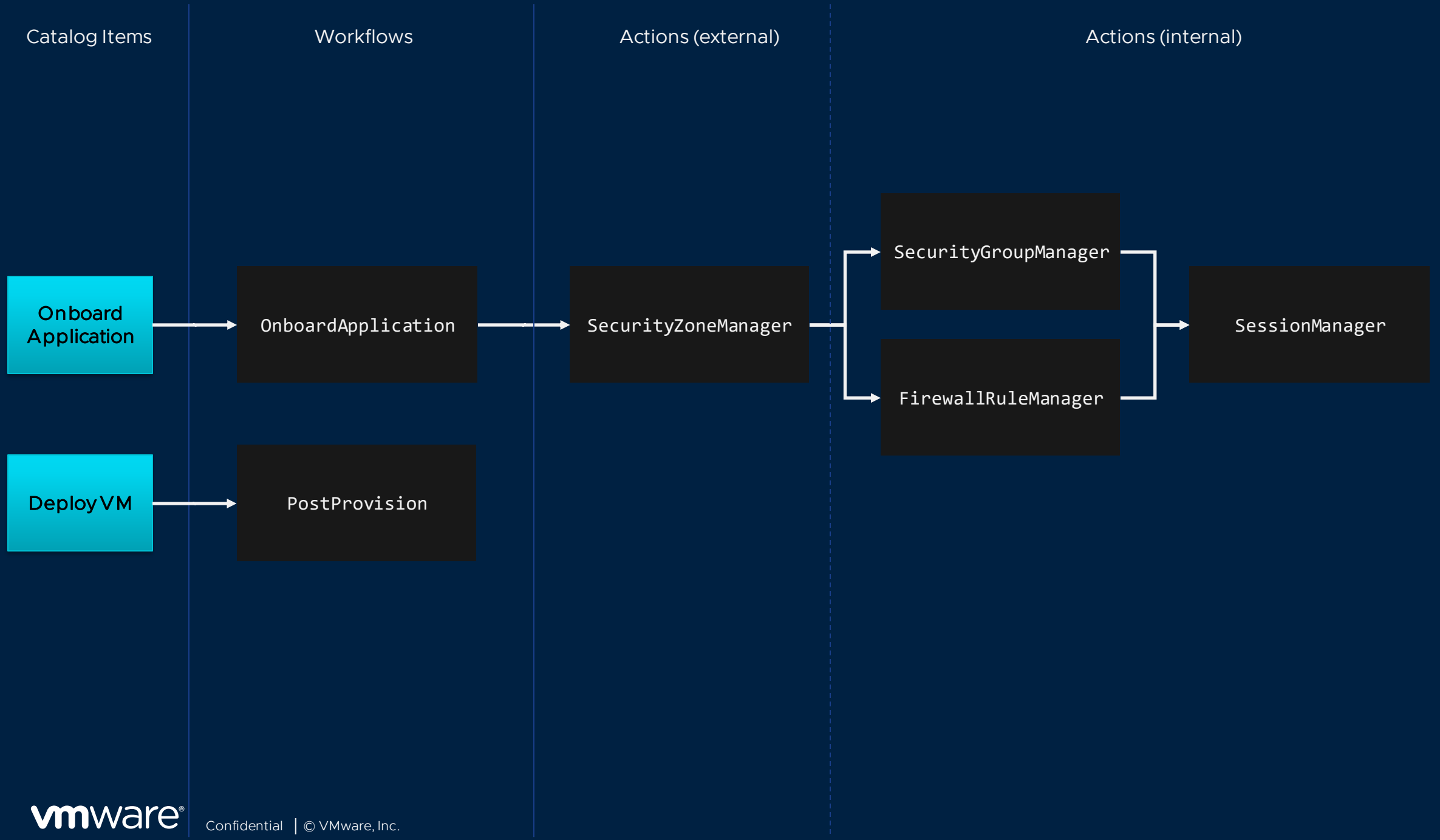
```
SessionManager(){
  getSessionToken(){
    // API call..
  }
}
```

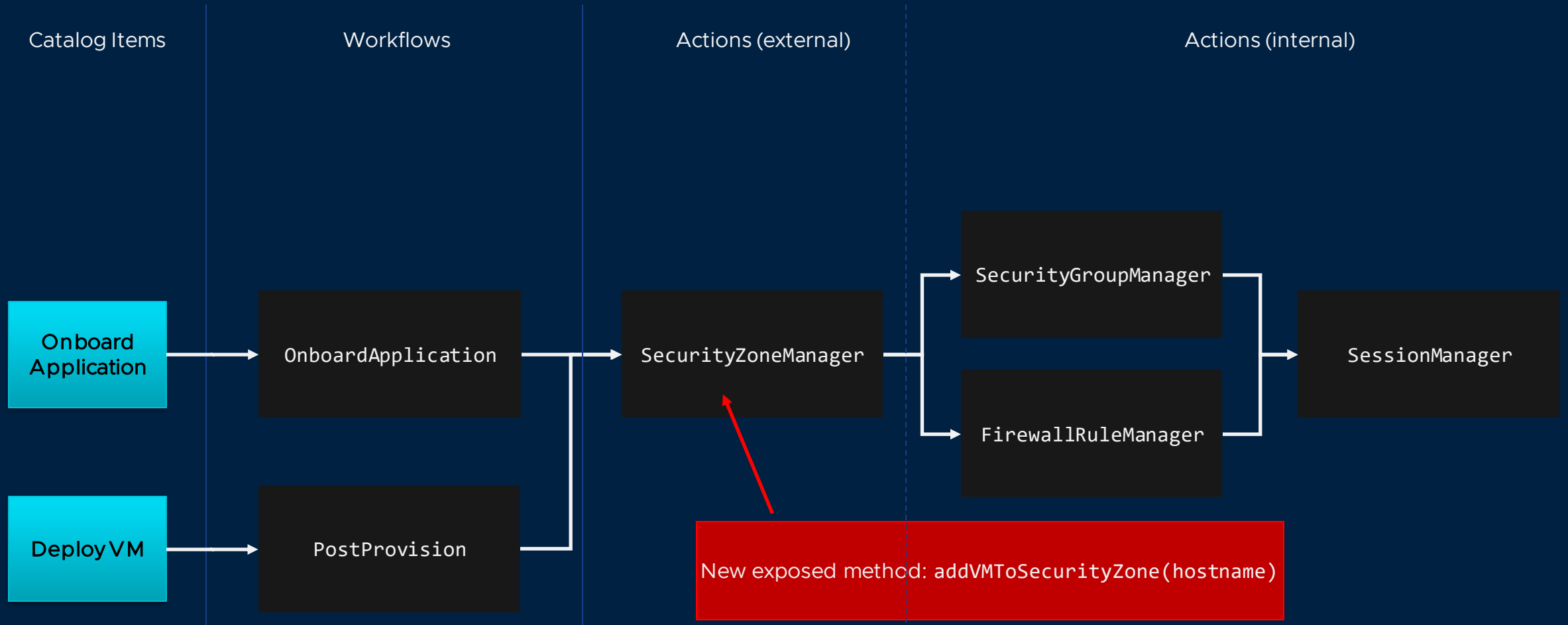
Logical Diagram

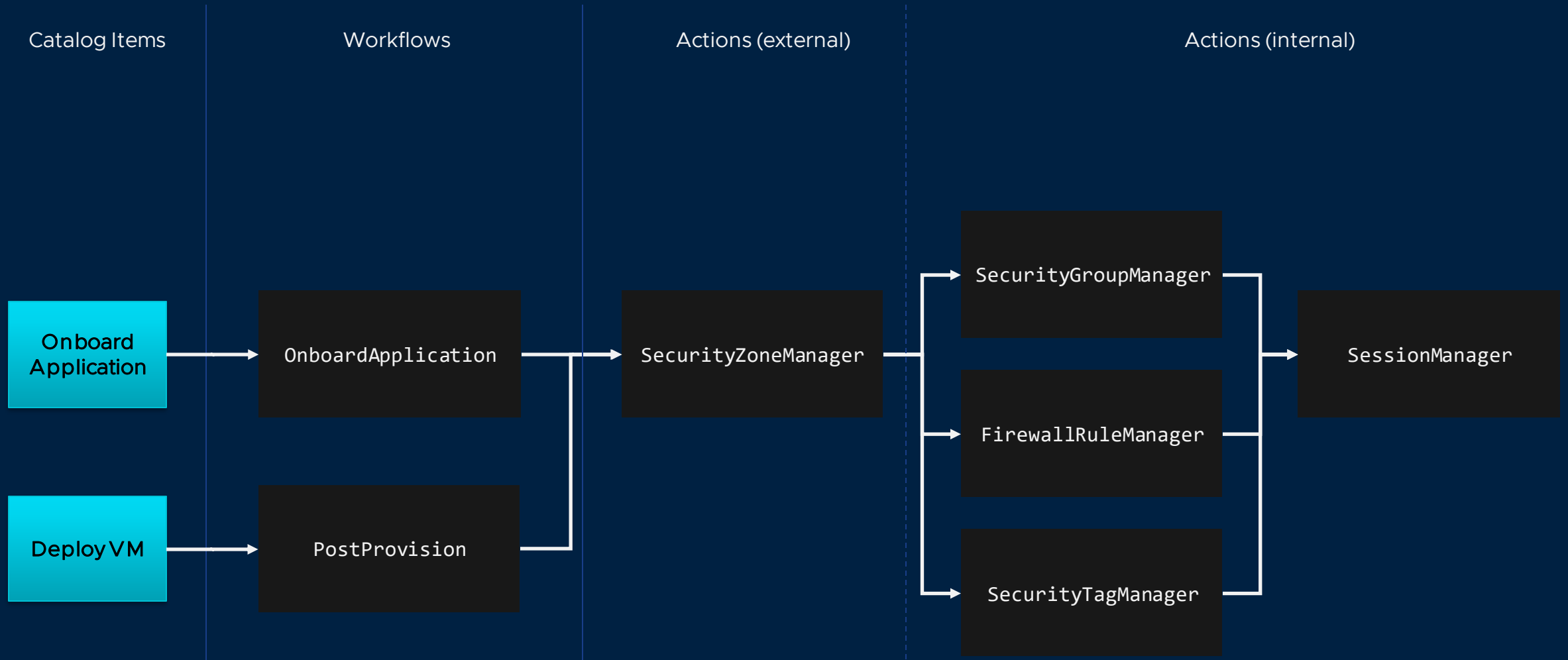

Onboard Application
OnboardApplication
SecurityZoneManager
SecurityGroupManager
FirewallRuleManager
SessionManager

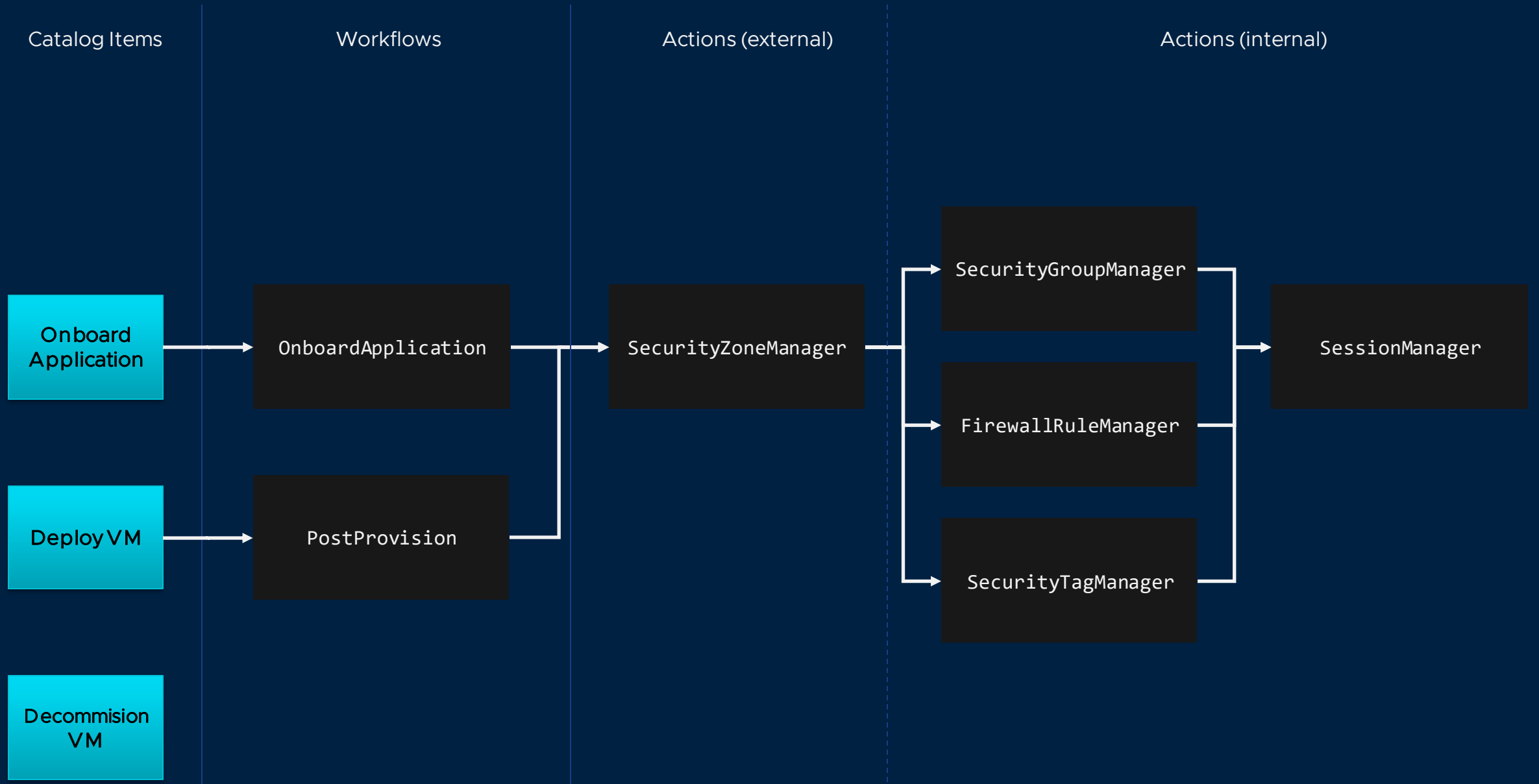

Catalog Items
Workflows
Actions (external)
Actions (internal)
Onboard Application
OnboardApplication
SecurityZoneManager
SecurityGroupManager
FirewallRuleManager
SessionManager

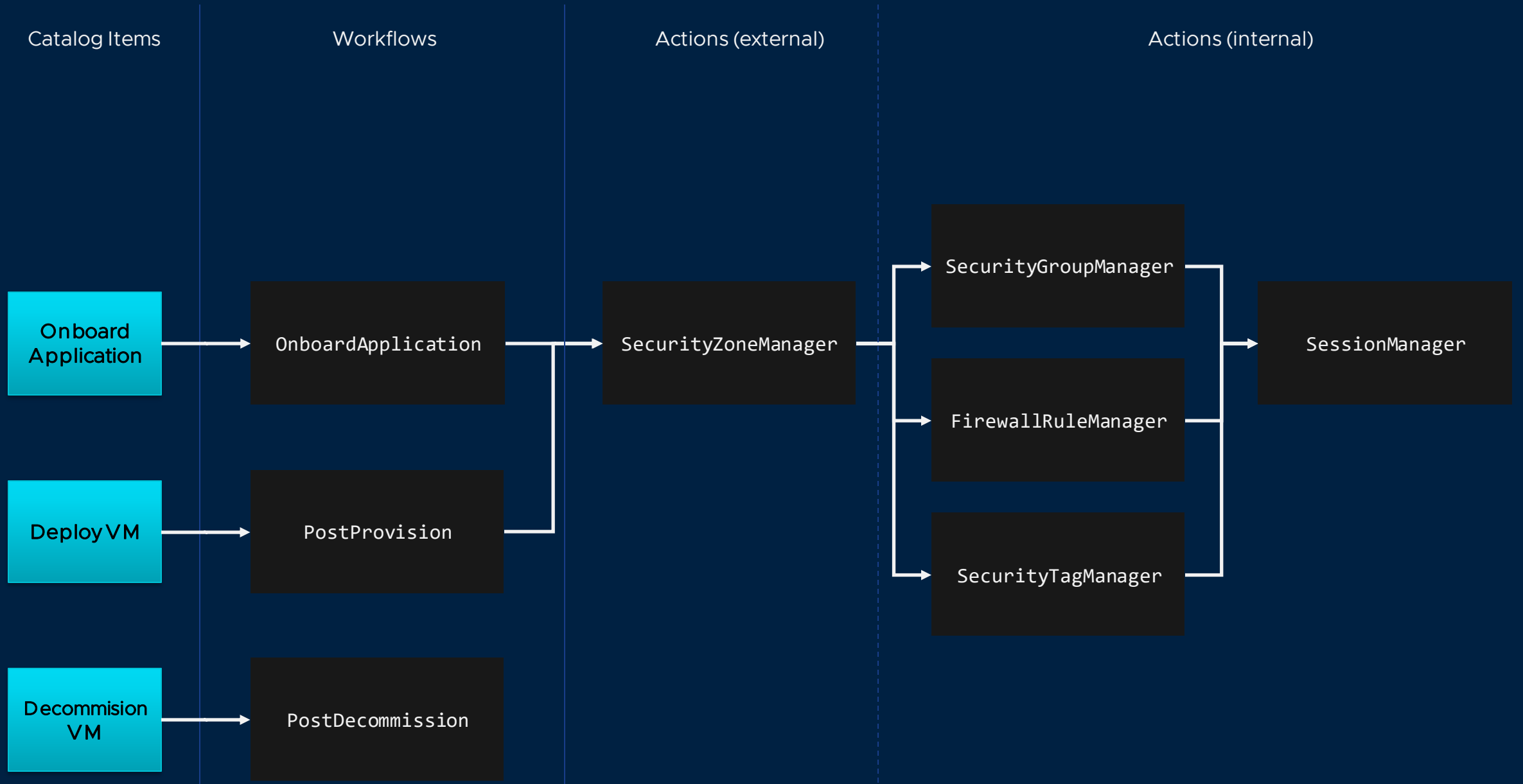

Catalog Items
Workflows
Actions (external)
Actions (internal)
Onboard Application
Deploy VM
OnboardApplication
SecurityZoneManager
SecurityGroupManager
FirewallRuleManager
SessionManager

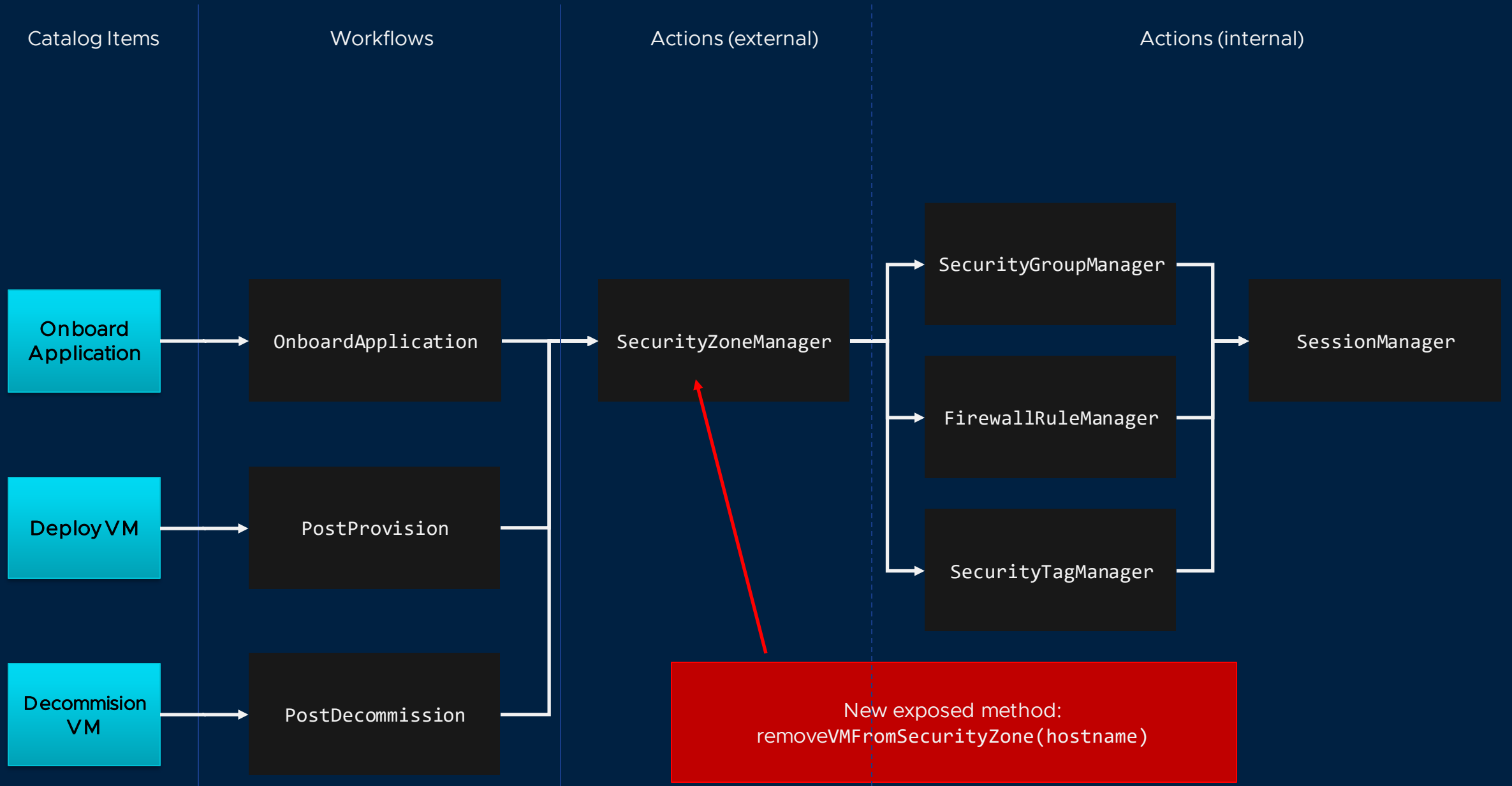

Catalog Items
Workflows
Actions (external)
Actions (internal)
Onboard Application
Deploy VM
OnboardApplication
PostProvision
SecurityZoneManager
SecurityGroupManager
FirewallRuleManager
SessionManager

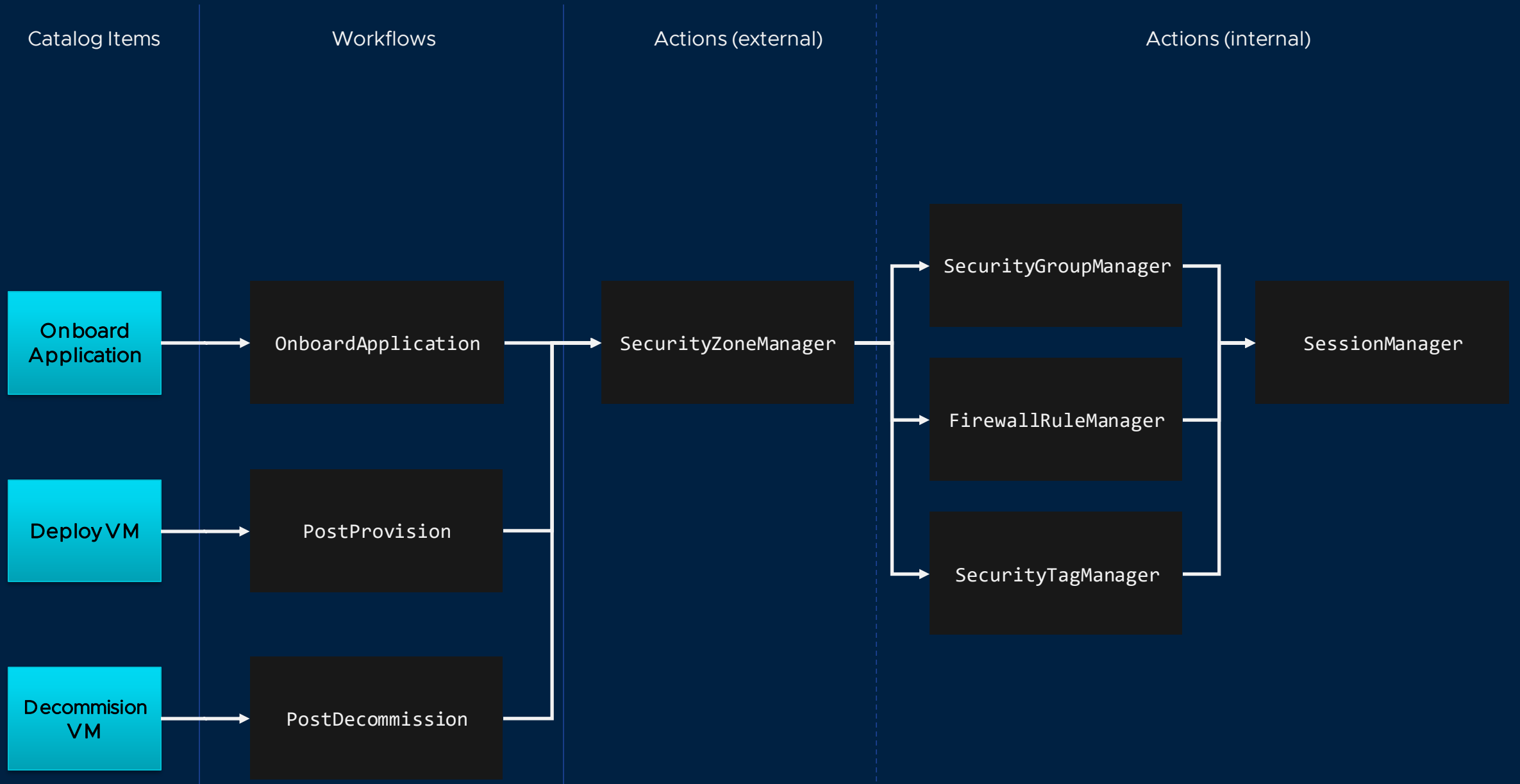

Catalog Items
Workflows
Actions (external)
Actions (internal)
Onboard Application
Deploy VM
OnboardApplication
PostProvision
SecurityZoneManager
SecurityGroupManager
FirewallRuleManager
SessionManager
New exposed method: addVMToSecurityZone(hostname)


Catalog Items
Workflows
Actions (external)
Actions (internal)
Onboard Application
Deploy VM
OnboardApplication
PostProvision
SecurityZoneManager
SecurityGroupManager
FirewallRuleManager
SecurityTagManager
SessionManager

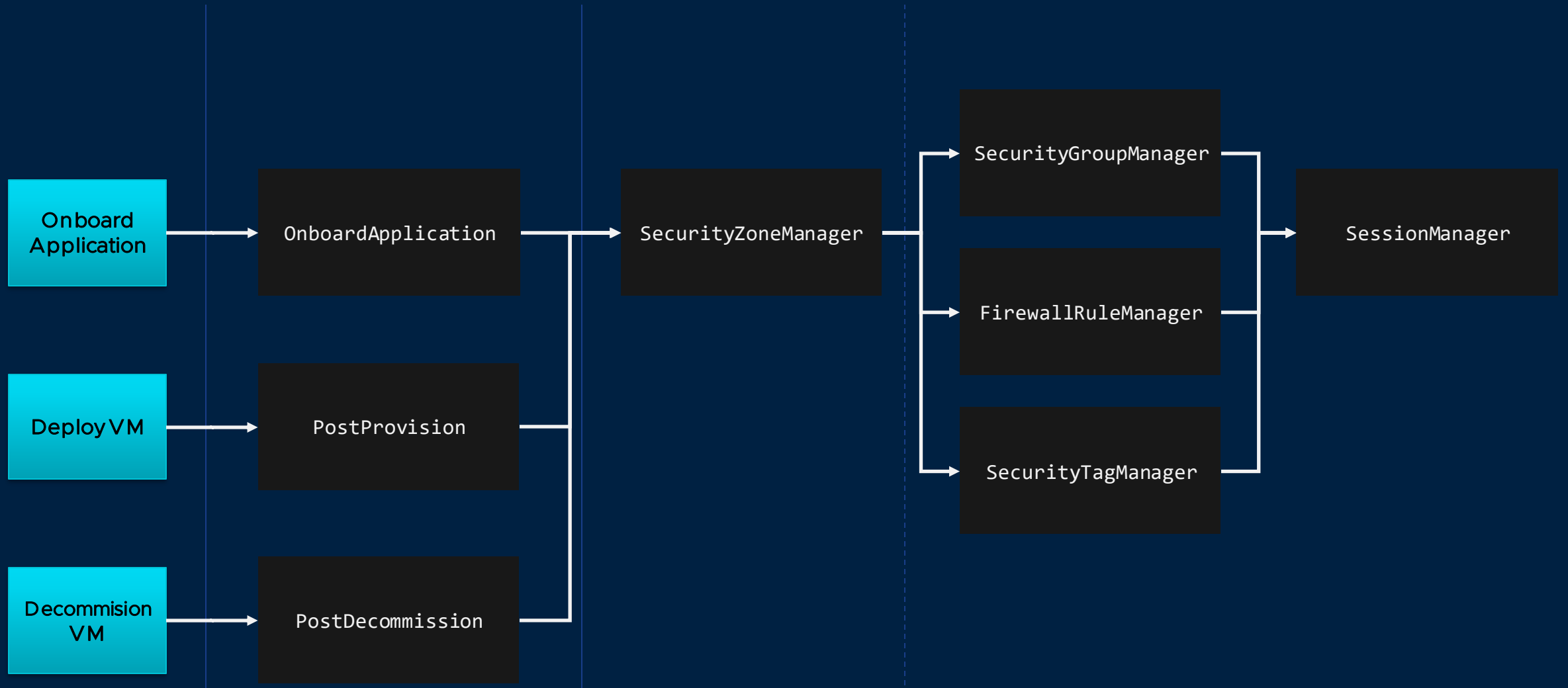

Catalog Items
Workflows
Actions (external)
Actions (internal)
Onboard Application
Deploy VM
Decommision VM
OnboardApplication
PostProvision
SecurityZoneManager
SecurityGroupManager
FirewallRuleManager
SecurityTagManager
SessionManager


Catalog Items
Workflows
Actions (external)
Actions (internal)
Onboard Application
Deploy VM
Decommision VM
OnboardApplication
PostProvision
PostDecommission
SecurityZoneManager
SecurityGroupManager
FirewallRuleManager
SecurityTagManager
SessionManager


Catalog Items
Workflows
Actions (external)
Actions (internal)
Onboard Application
Deploy VM
Decommision VM
OnboardApplication
PostProvision
PostDecommission
SecurityZoneManager
SecurityGroupManager
FirewallRuleManager
SecurityTagManager
SessionManager
New exposed method:
removeVMFromSecurityZone(hostname)


Catalog Items
Workflows
Actions (external)
Actions (internal)
Onboard Application
Deploy VM
Decommision VM
OnboardApplication
PostProvision
PostDecommission
SecurityZoneManager
SecurityGroupManager
FirewallRuleManager
SecurityTagManager
SessionManager

Goal: Improve Quality by looking at maintainability, extensibility and modularity



Thank You

Jordy.van-Leersum@broadcom.com
https://www.linkedin.com/in/jordy-van-leersum

vmware Confidential | © VMware, Inc.