LF NETWORKING

LFN Developer & Testing Forum

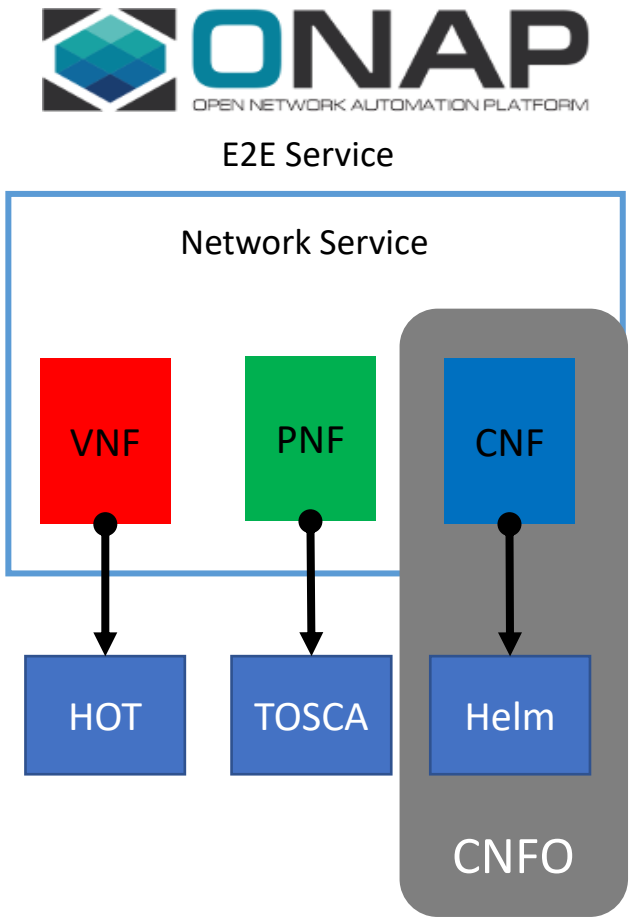
CNF Orchestration Scenarios

Jakarta Updates & How To Use Existing Features

Lukasz Rajewski (T-Mobile)
Seshu Kumar (Huawei)

14.06.2022

CNF Orchestration (1)



REQ-341
Guilin

REQ-458
Honolulu

REQ-627
Istanbul

REQ-890
Jakarta

CNF Orchestration (2)

NOKIA

HUAWEI

CNFO

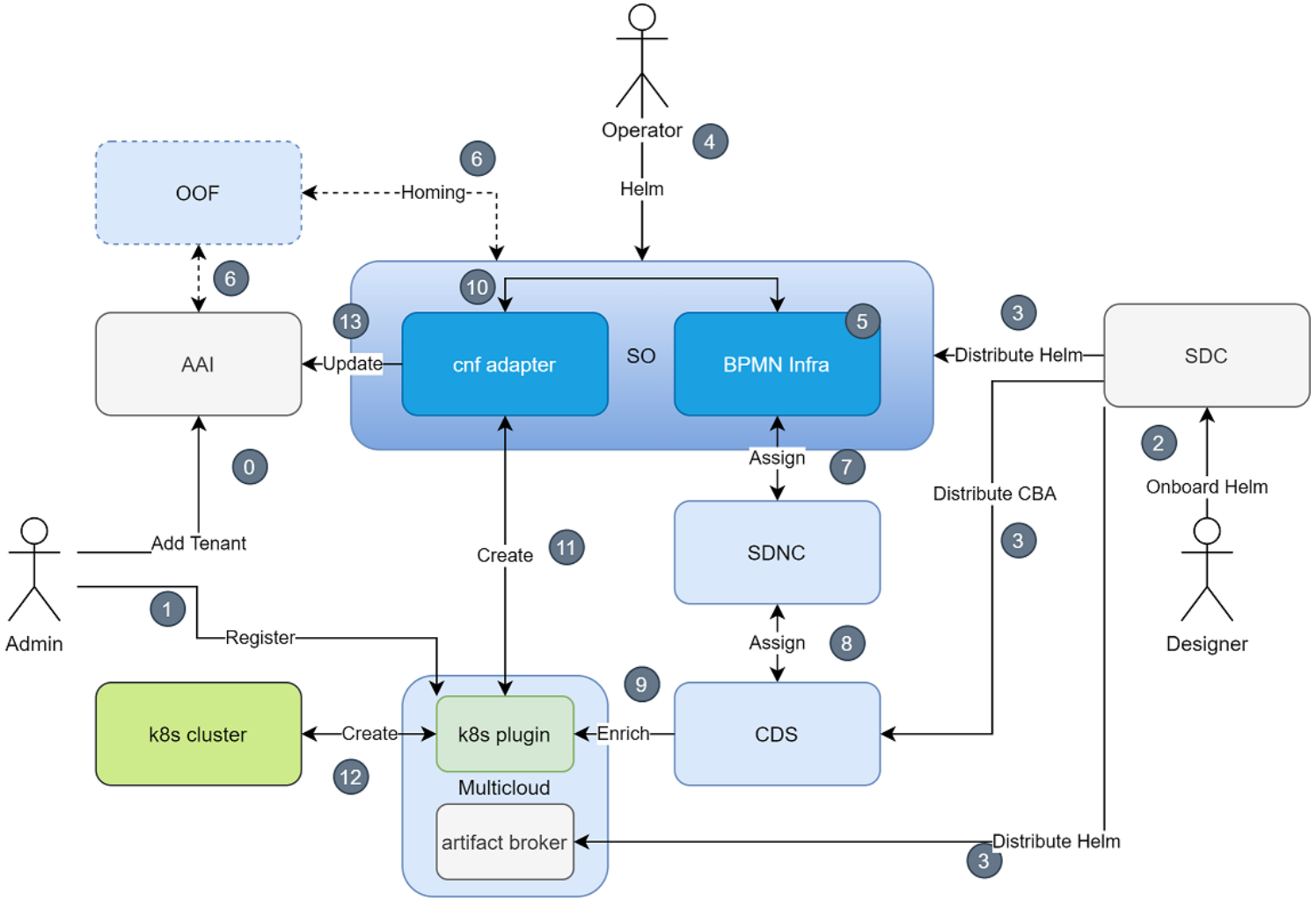
Tech Mahindra

intel

SAMSUNG

AT&T

CNF Orchestration (3)



- **Helm Onboarding** **[SDC]**
- **Day 0/1 Customization** **[CDS/MC]**
- **Instantiation** **[SO/MC]**
- **Day 2 Configuration** **[CDS/MC]**
- **State Synchronization** **[SO/AAI/MC]**
- **Healthcheck** **[SO/CDS/MC]**
- **Upgrade** **[SO/CDS/MC]**
- **Multi-cluster deployment** **[SO/MC]**
- **VNF/PNF Integration** **[ALL]**
- **E2E Service Automation** **[SDK]**

CNFO Jakarta Changes

CNFO Jakarta Changes

- Refined AAI Synchronization
  - K8S-Resource object created after CNF deployment
  - Changes in K8s Cluster Updated in AAI automatically
  - **Synchronization into AAI objects created by K8s Operators**
- **K8S Resource Change Notifications**
- **Helm Upgrade Endpoint**
- Easy K8s Resource modifications through CDS
  - Create/Update/Delete/Rollback
- Apache CNF OOM Gating Test

K8sPlugin - Helm Upgrade (1)

- Instance Upgrade Endpoint
  - http://multicloud-k8s:9015/v1/instance/<id>/upgrade
  - New resources created, existing upgraded, old removed
  - Upgrade, Migration, Reconfiguration
- Mimic of the implementation of helm hooks for upgrade operation
  - Support of pre/post upgrade hooks
  - Hooks can be created in order
  - Next one must wait for previous one to finish before continue.
  - Support hook upgrade policy

{upgrade}

K8sPlugin - Helm Upgrade (2)

POST http://multicloud-k8s:9015/v1/instance/<id>/upgrade

```
{
  "rb-name": "22a02dd7-6c87-48c4-ada7-f592e6c83f73",
  "rb-version": "430b035b-e13e-4636-8600-7404ba2f9d33",
  "profile-name": "profile-node-port",
  "cloud-region": "region-1",
  "labels": {
      "custom-label-1": "value"
  },
  "override-values": {
      "version": "10.0.0"
  }
}
```

- rb-name + rb-version
  - Helm package upgrade
- cloud-region
  - Migration
- profile-name
  - Namespace change
  - Overrides change
  - Update K8s cluster version
  - Change of extra types for AAI synch

K8sPlugin – Additional API Changes

- Status API – Ready Flag (All Resources in Desired State)
- Config API
  - Config Template from the main RB Definition
  - Create/Update/Delete/Rollback Config
- Profile API
  - Extra K8s-Resource Types for Status API
- Status Notification API
  - Create/Delete/List Status Subscription
  - Notification about Create/Delete/Update of k8s-resource
  - Notification has new Status
  - Used By CNF-Adapter to update k8s-resources in AAI

AAI model: k8s resource object

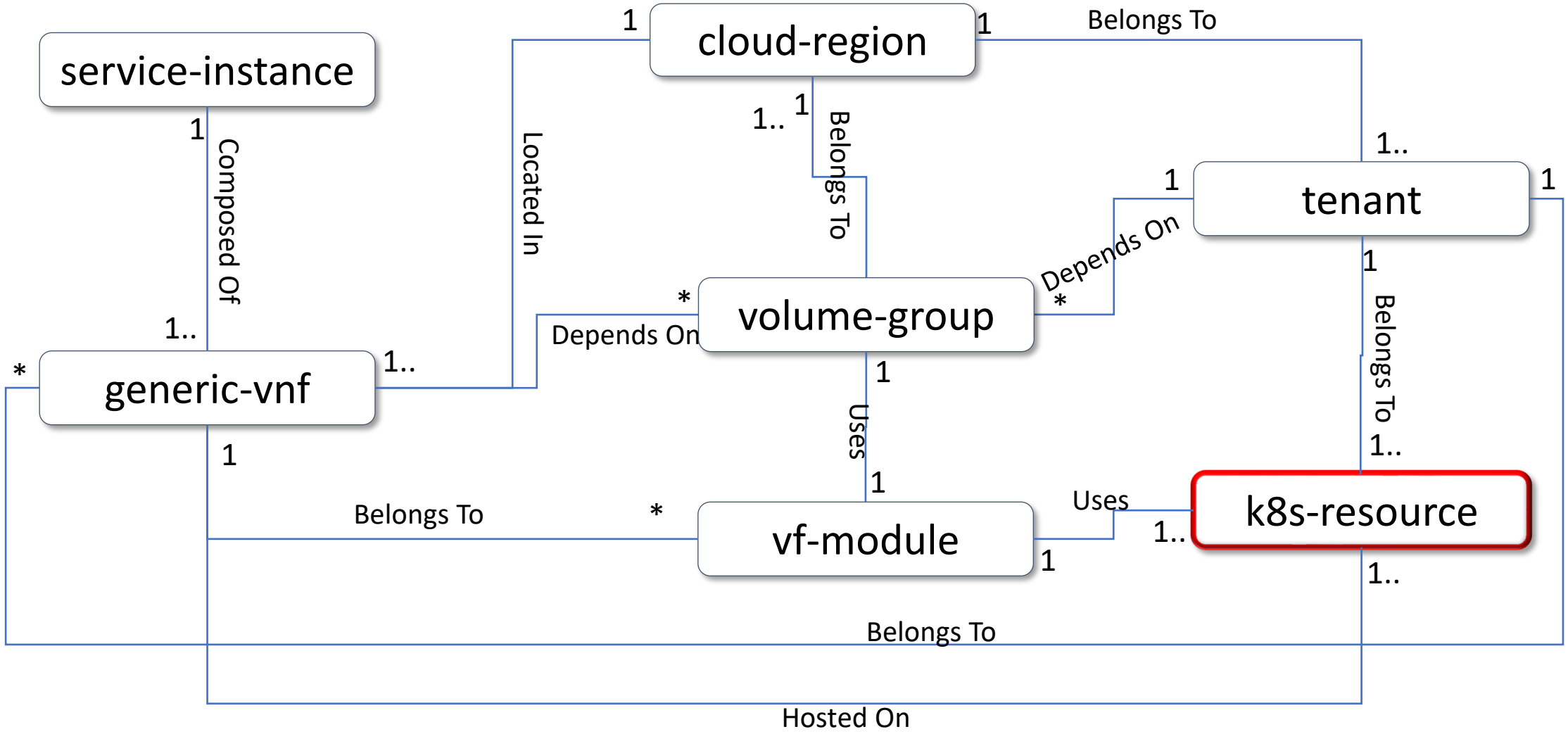
Attribute	Type	Mandatory
id	UUID	Yes (PK)
name	String	Yes
group	String	Yes
version	String	Yes
kind	String	Yes
labels	List of strings	No
namespace	String	Yes
selflink	URI	Yes

K8s resource is basic AAI entity to model resources created in K8s cluster.

It plays similar role as vserver resource for standard VNFs.

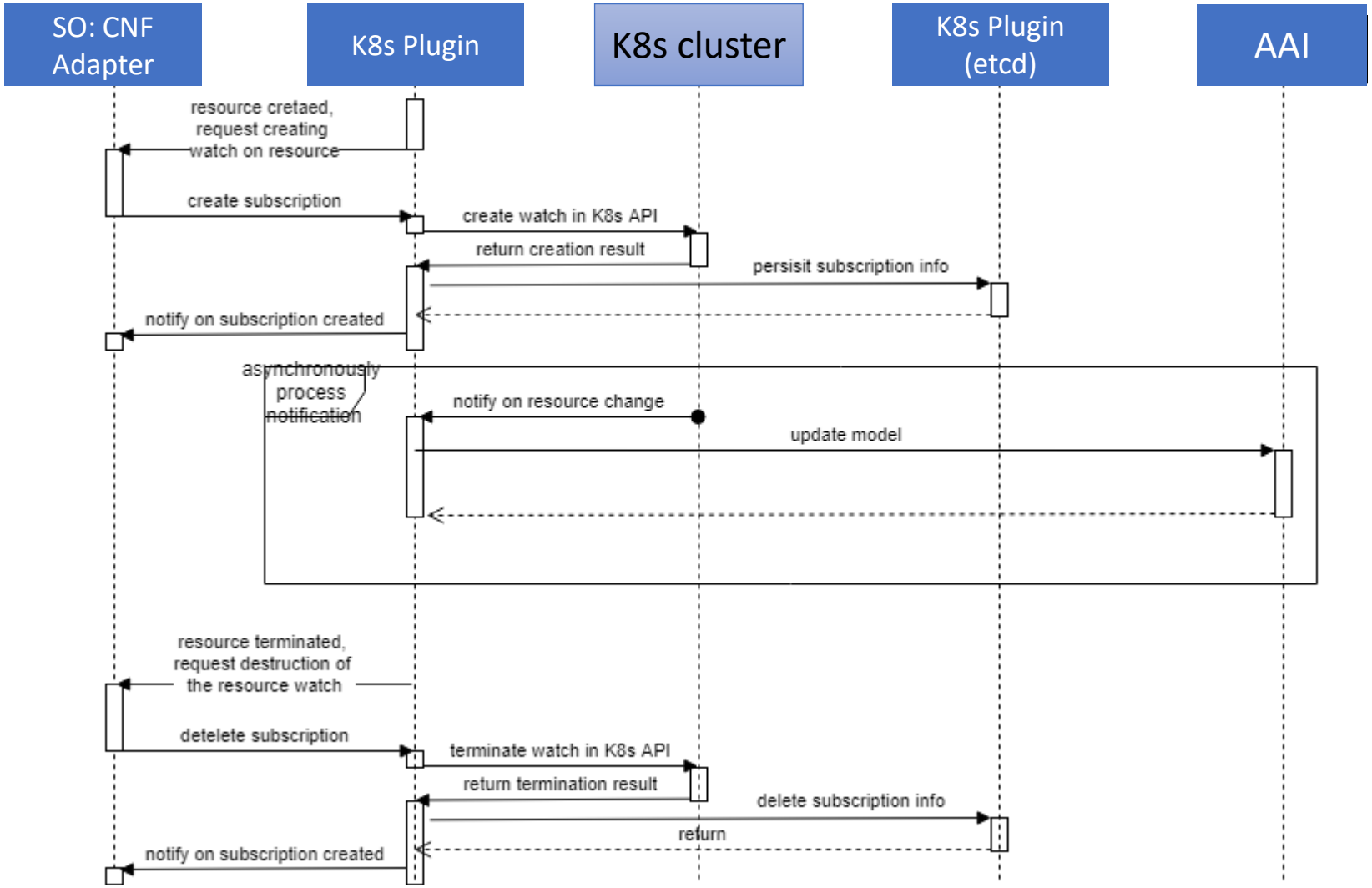
Self-link allows to access full and actual details of the k8s resource

AAI model: relations



https://wiki.onap.org/display/DW/Simplified+K8S+Resource+Model+-+IM

CNF AAI Update - Jakarta



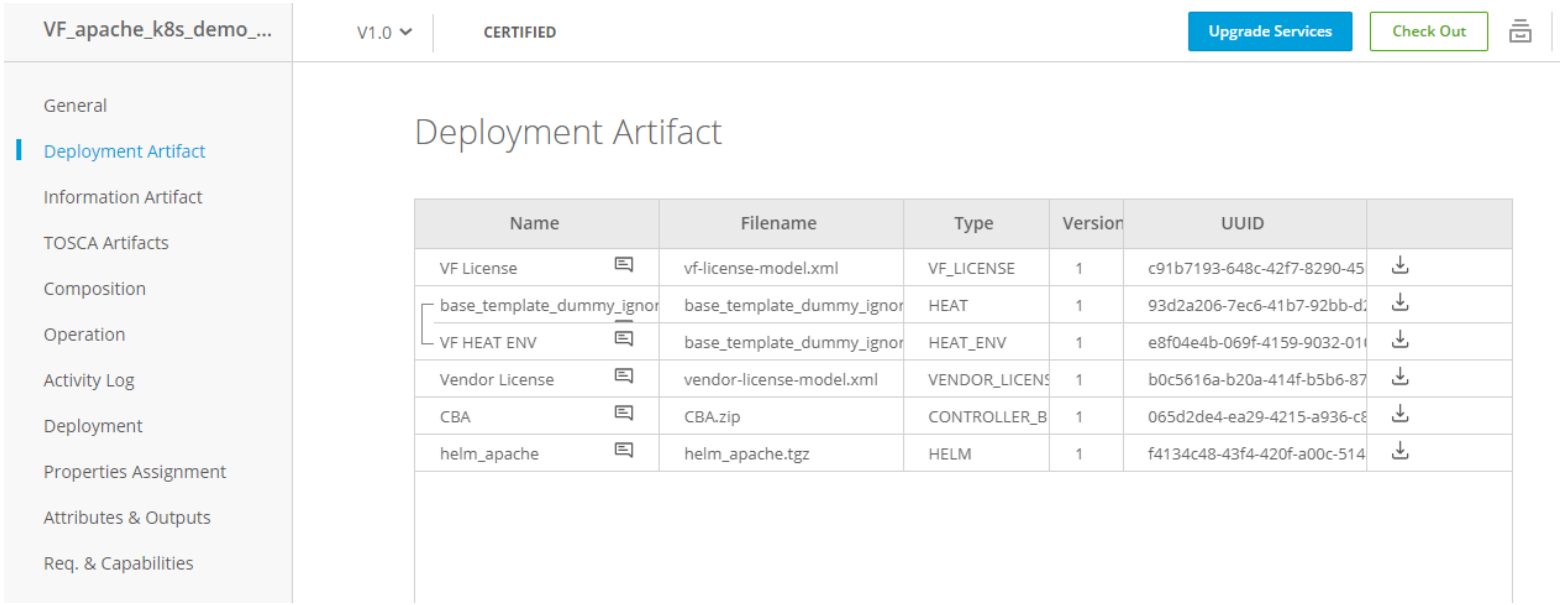
1. CNF Adapter creates status notification subscription
2. K8s Notifies on Resource's change
3. K8sPlugin Sends Subscription Notification
4. CNF Adapter Determines type of change
   - Create new k8s-resource
   - Deletes k8s-resource
   - Update K8s resource version

CNFO Scenarios

CNFO Onboarding

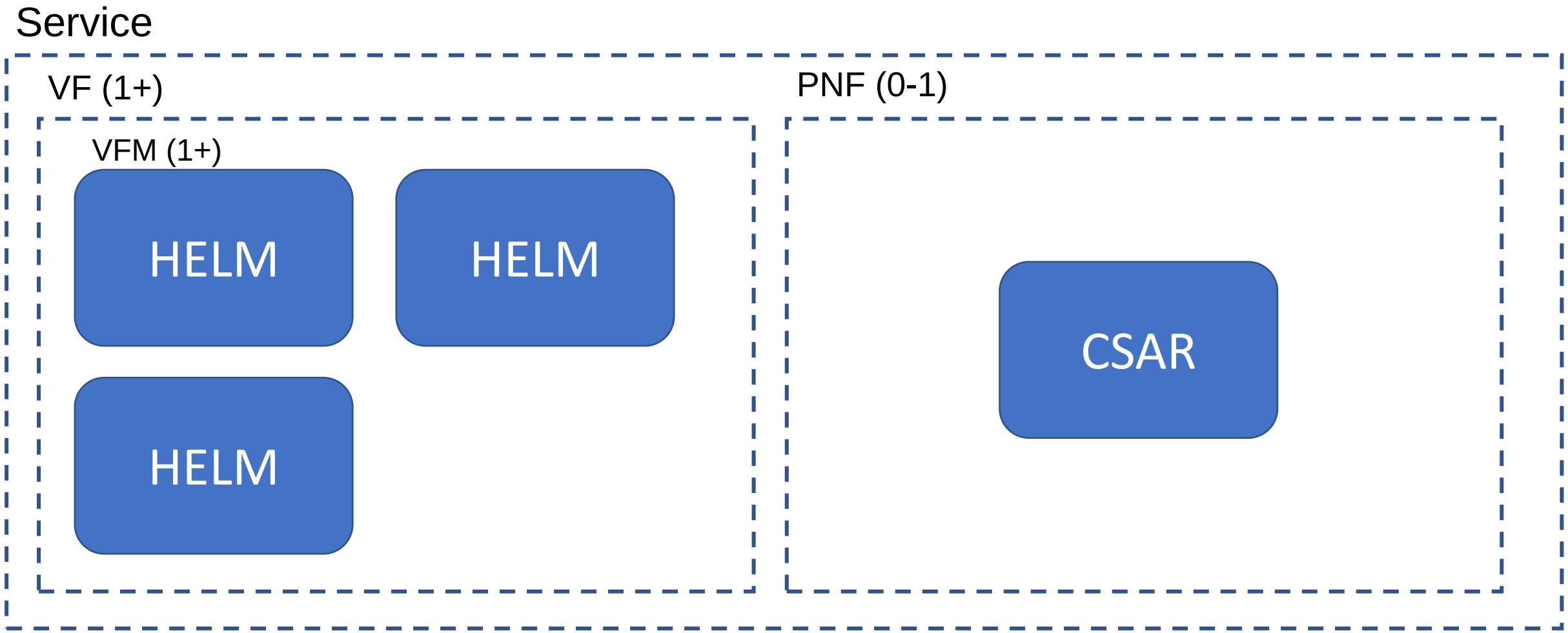
LFN Developer & Testing Forum

```
{
    "name": "simpleCNF",
    "description": "",
    "data": [
        {
            "file": "CBA.zip",
            "type": "CONTROLLER_BLUEPRINT_ARCHIVE"
        },
        {
            "file": "helm_apache.tgz",
            "type": "HELM",
            "isBase": "true"
        }
    ]
}
```

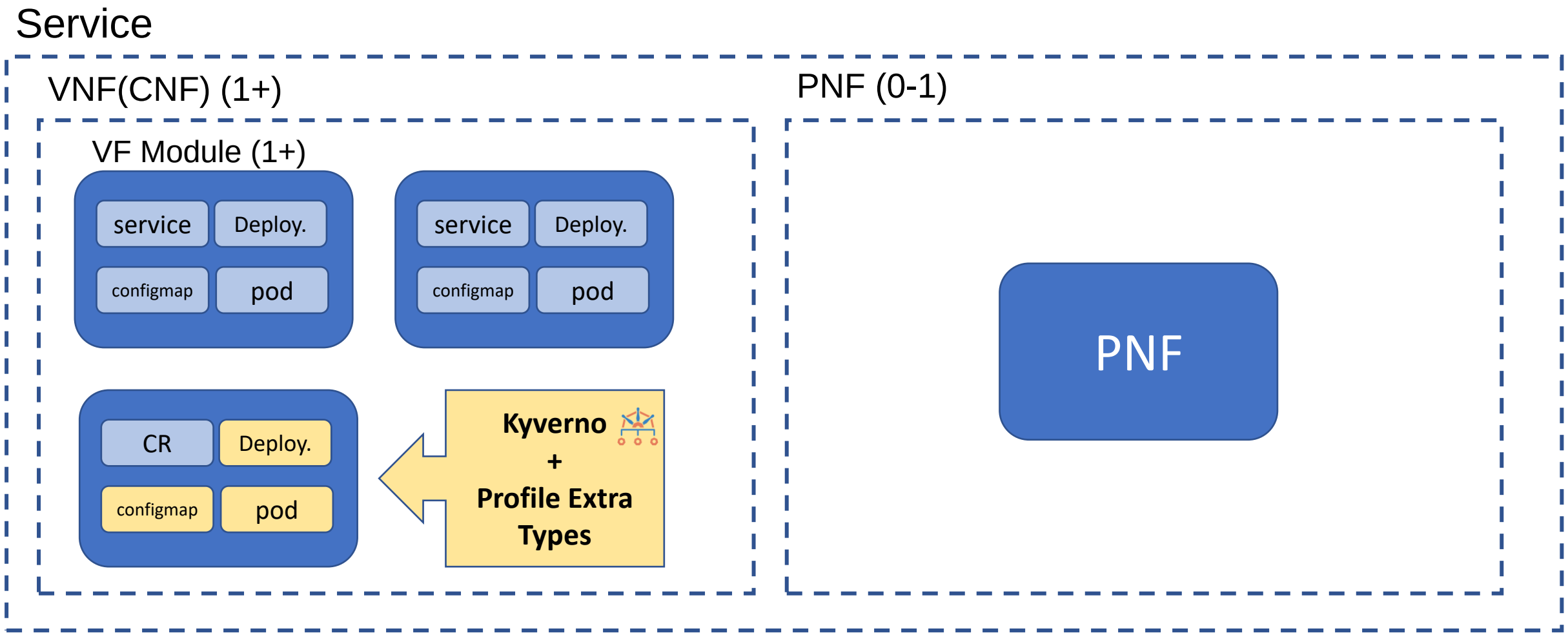


- Standard Simple VSP Package (ZIP)
- CBA is crucial and mandatory for CNFO
- In the future may be replaced with ASD

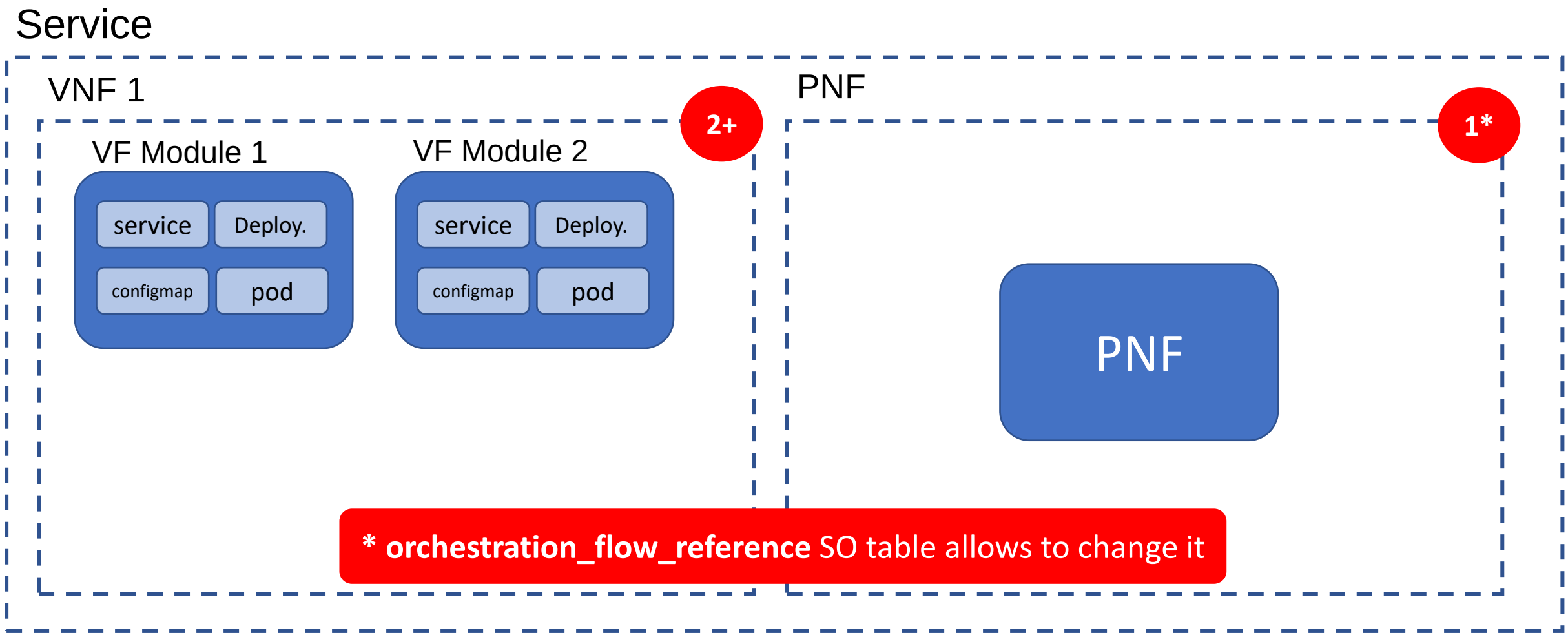
ONAP modeling concept (SDC)



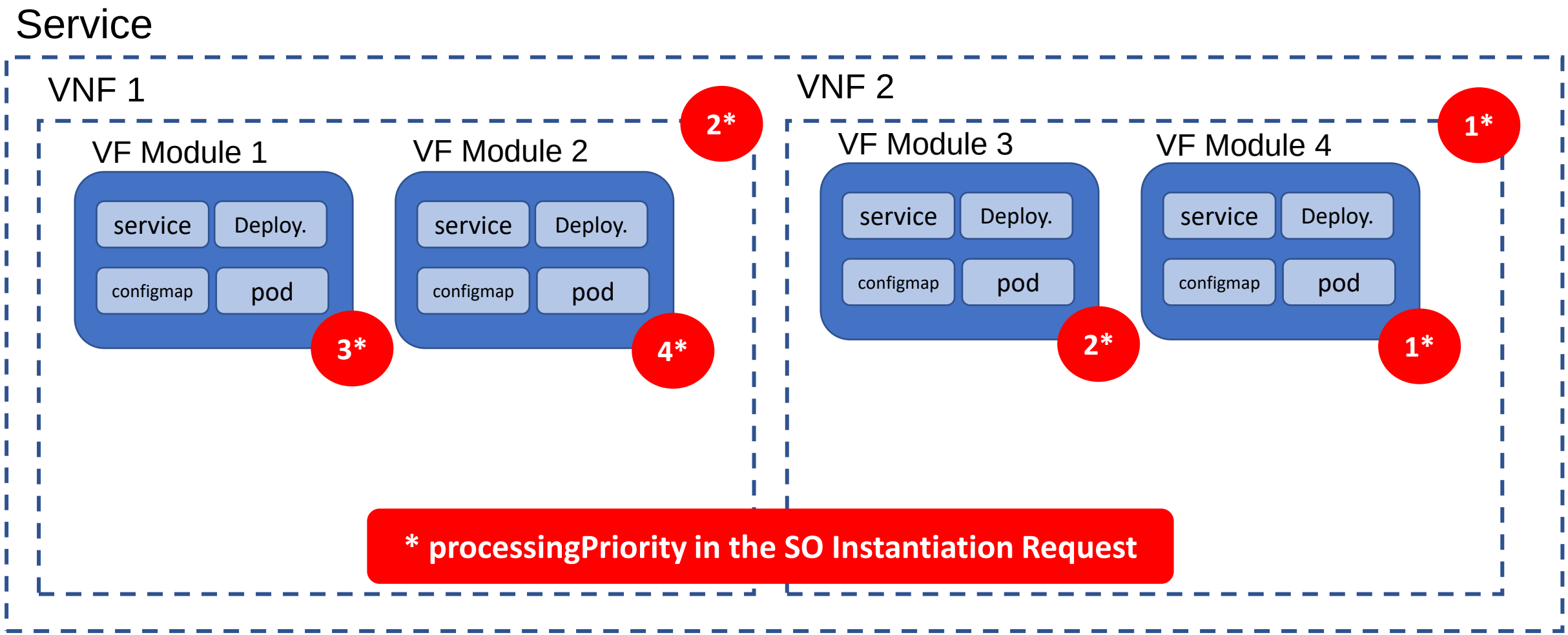
ONAP modeling concept (AAI)



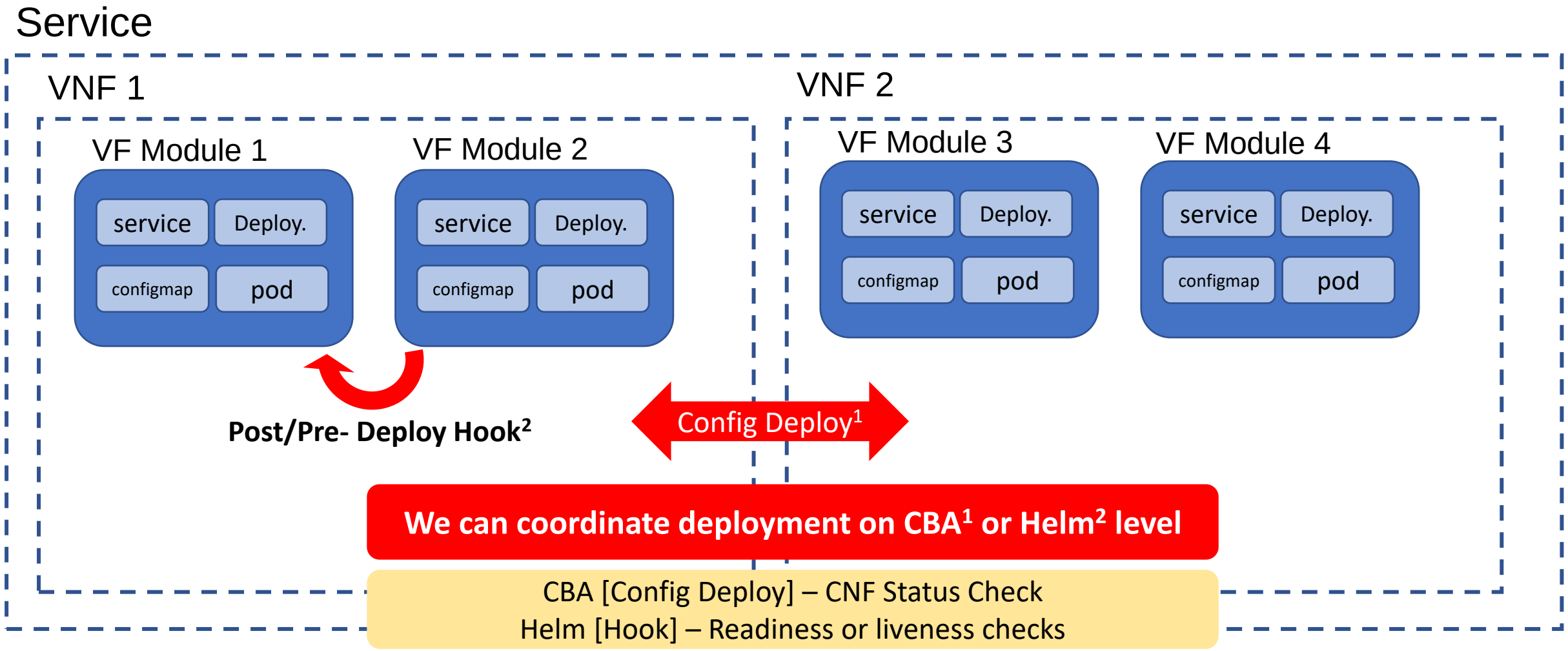
CNF/PNF Coordination (1)



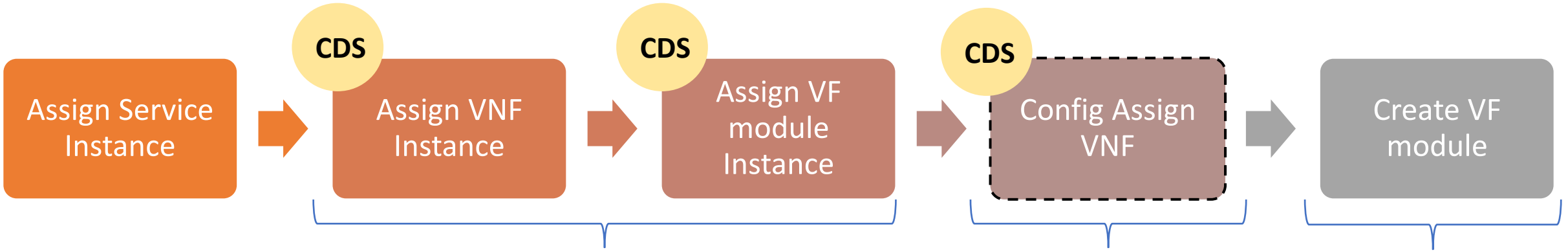
CNF/PNF Coordination (2)

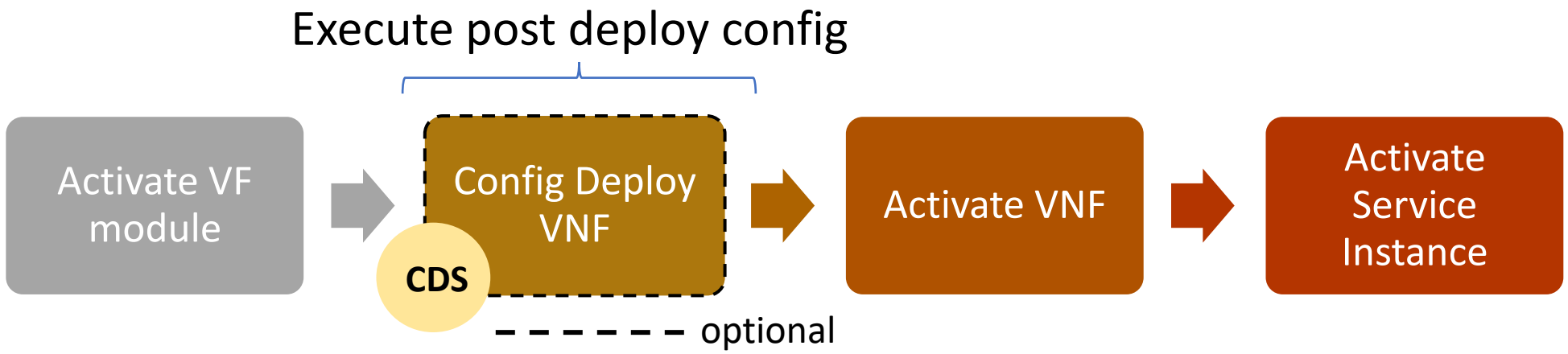


CNF/PNF Coordination (3)

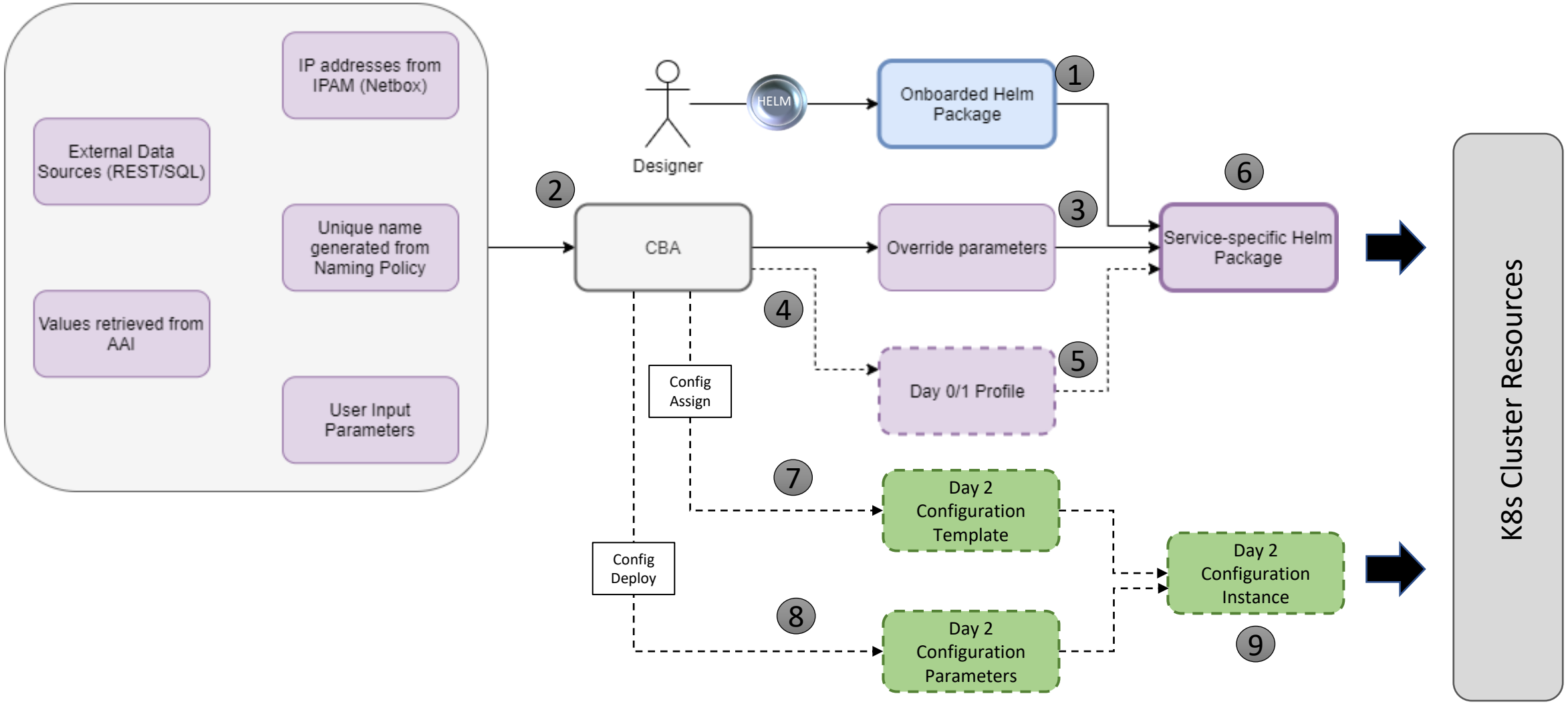


CNF Instantiation (macro mode)





Helm Package Day 0/1 + Day2



CNF Day 0 – Helm Enrichment


```
"resource-assignment": {
    "steps": {
        "resource-assignment": {
            "description": "Resource Assign Workflow",
            "target": "resource-assignment",
            "activities": [
                {
                    "call_operation": "ResourceResolutionComponent.process"
                }
            ],
            "on_success": [
                "profile-upload"
            ]
        },
        "profile-upload": {
            "description": "Generate and upload K8s Profile",
            "target": "k8s-profile-upload",
            "activities": [
                {
                    "call_operation": "K8sProfileUploadComponent.process"
                }
            ]
        }
    },
```

- CNF instance based
- Modifies Helm package from VSP
- K8s Profile Creation & Upload
  - Native mechanisms in CDS
  - Customizable by CBA
- Modification of Helm values
- Customization of labels
- Selection of k8s namespace
- Modification of Helm templates
- Provisioning of new Helm templates
- New k8s-resource types to

CNF Day 2 – Config Preparation


```
"config-assign": {
    "steps": {
        "config-setup": {
            "description": "Gather necessary input for config template upload",
            "target": "config-setup-process",
            "activities": [
                {
                    "call_operation": "ResourceResolutionComponent.process"
                }
            ],
            "on_success": [
                "config-template"
            ]
        },
        "config-template": {
            "description": "Generate and upload K8s config template",
            "target": "k8s-config-template",
            "activities": [
                {
                    "call_operation": "K8sConfigTemplateComponent.process"
                }
            ]
        }
    },
```

- CNF instance based
- Config Template (CFT)
  - Helm package
  - Build or modified by CDS
  - We can use VSP Helm as a template
- CFT preparation may be a part of Config-Assign in CDS
- Native mechanisms in CDS
  - Customizable by CBA
- Config Setup merges data
  - CBA
  - AAI i.e. vf-modules info
  - MDSAL – i.e. resolved Day 0
  - K8s – i.e. k8s resource status info
  - Kotlin, Python, REST
  - Complex JSON

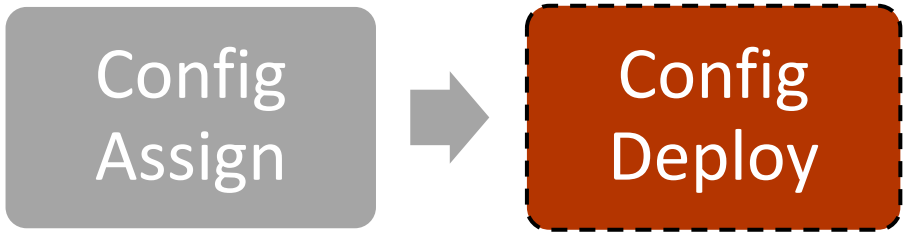
CNF Day 2 – Config Creation

LFN Developer & Testing Forum

```
"config-deploy": {
    "steps": {
        "config-setup": {
            "description": "Gather necessary input for config init and status verification",
            "target": "config-setup-process",
            "activities": [
                {
                    "call_operation": "ResourceResolutionComponent.process"
                }
            ],
            "on_success": [
                "config-apply"
            ],
            "on_failure": [
                "handle_error"
            ]
        },
        "config-apply": {
            "description": "Activate K8s config template",
            "target": "k8s-config-apply",
            "activities": [
                {
                    "call_operation": "K8sConfigTemplateComponent.process"
                }
            ],
            "on_success": [
                "status-verification-script"
            ]
        },
```

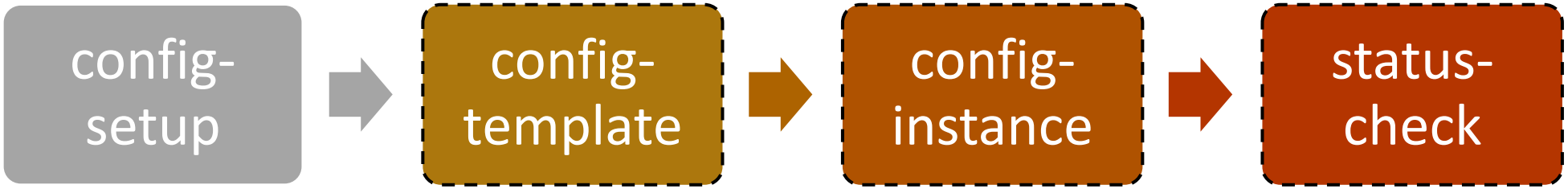
- CNF instance based
- Config Instance (CFI)
  - Instantiates CFT
  - Provides overrides for CFT
- CFI creation is part of Config-Deploy in CDS
  - Creates new k8s resources
  - Modifies k8s resources of existing CNF instance
- Native mechanisms in CDS
  - Customizable by CBA
- In vFW CNF Use Case followed by simple Status Check
  - Checks Pod Status until „Running”
  - Fails after 30 retries

VNF-Macro-Modify Workflow



- **SO Workflow To Trigger Any Day-2 Operation with CDS**
- ControllerExecutionBB is utilized in each SO workflow's step
- **config-assign** and **config-deploy** CBA workflows utilized
- Nº1 step - determine the desired CDS workflow and its inputs
- Nº2 step - execute **desired CBA workflow** (Kotlin script)
  - BluePrintWorkflowExecutionService.executeBluePrintWorkflow()
- **One SO workflow to execute any CBA workflow for Day-2 operation**

CNF Scaling On-Demand



Apache CNF Use Case

- **config-setup –** takes details of CNF instances from AAI and MDSAL
- **config-template –** creates config template from VSP Helm charts
- **config-instance –** enables configuration
  - We just specify- new overrides ReplicaCount of deployment in this case
  - ReplicaCount can come from the input
  - K8sPlugin performs „kubectl apply” for all k8s resources from the Config Helm package
- **status-check –** verifies status of pods till the Running state

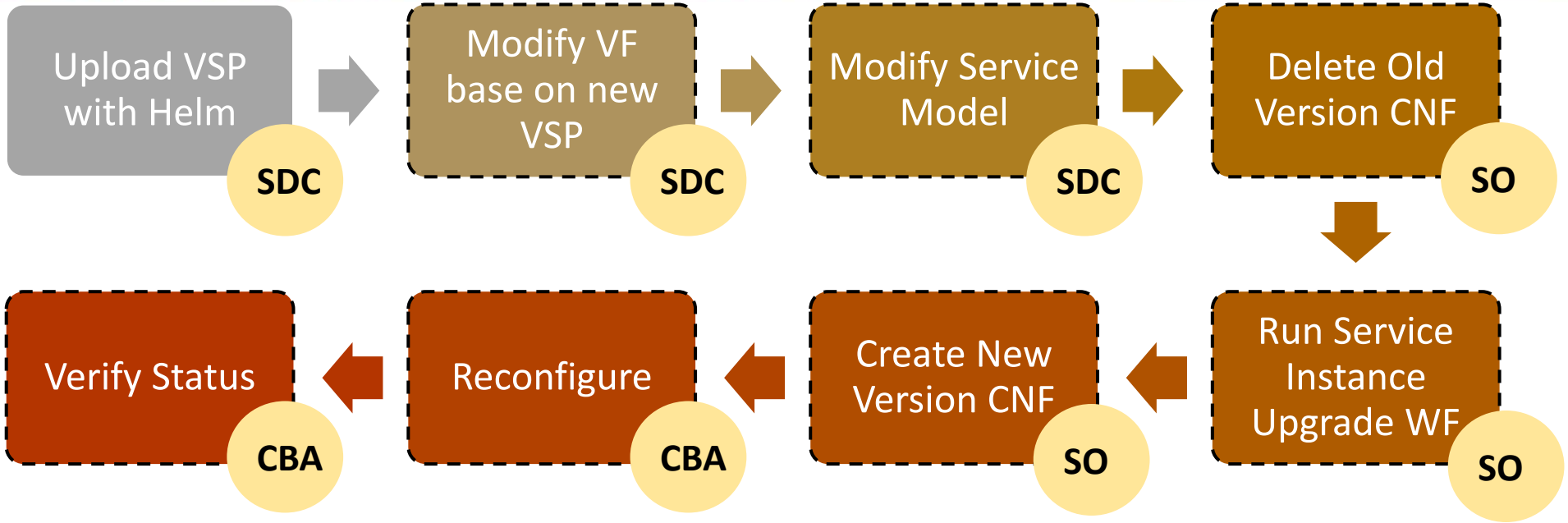
CNF Upgrade Jakarta Options

Build & Replace

Config Apply

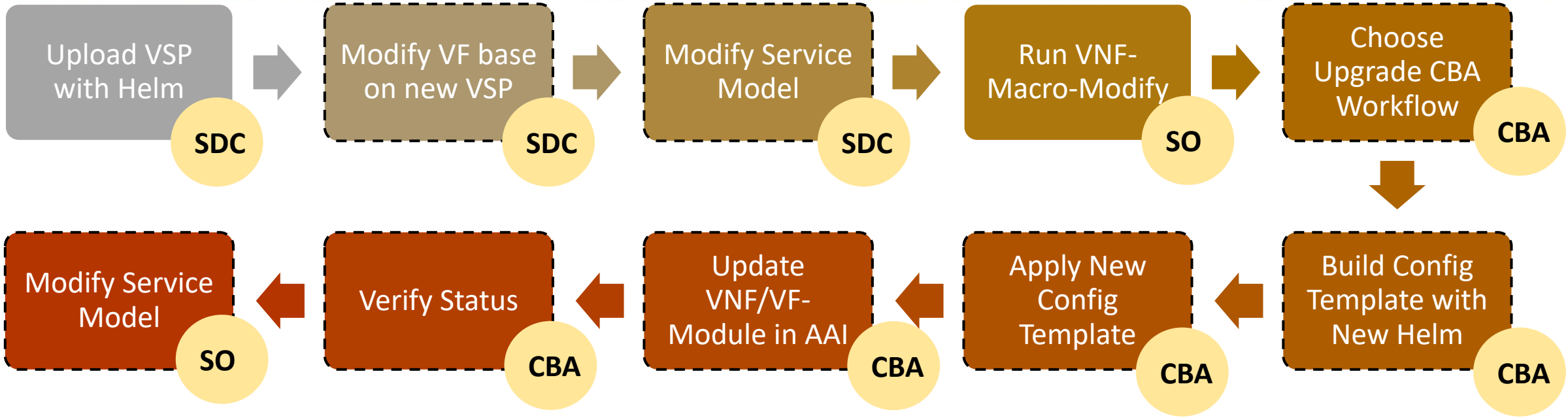
Upgrade in CBA

CNF Upgrade Jakarta (1)



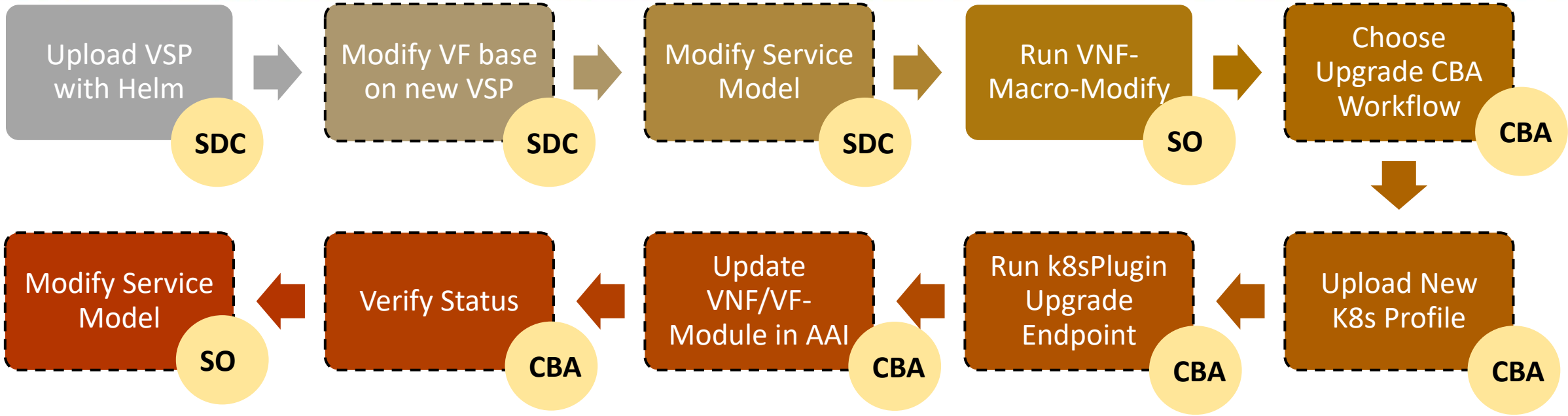
- **Simple to execute**
- **We need to remove old version of the CNF firstly**
- Requires traffic migration/redundancy to not loose the traffic
- All required workflows present in SO (REQ-883)
  - POST serviceInstances/$SERVICE_INSTANCE_ID/upgrade
  - POST serviceInstances/$SERVICE_INSTANCE_ID/vnfs
  - DELETE serviceInstances/$SERVICE_INSTANCE_ID/vnfs/$VNF_ID

CNF Upgrade Jakarta (2)



- **Complex scenario requiring implementation in CBA**
- **No need to stop CNF but**
- We can skip service instance upgrade and AAI Update
  - We loose information about current model in AAI
- We need to implement two new operations in CBA
  - Create Config Template from VFM-Model
  - Update VNF and VF-Module Model info in AAI

CNF Upgrade Jakarta (3)



- **Moderate scenario but also requires implementation in CBA**
- **No need to stop CNF**
- We can skip service instance upgrade and AAI Update
  - We loose information about current model in AAI
- We need to implement three new operations in CBA
  - Create Config Template from VFM-Model
  - Update VNF and VF-Module Model info in AAI

CNF Migration (1)

- We can use k8splugin Upgrade endpoint in CBA
- We can modify namespace with by dedicated k8s profile
- K8splugin will upgrade instance
- CNF adapter will update AAI k8s resources
- Tenant relation must be changed in the CBA

```
{
    "rb-name": "${rb_name}",
    "rb-version": "${rb_version}",
    "profile-name": "${profile_name}",
    "cloud-region": "${cloud_region_id}",
    "labels": {
        "custom-label-1": "abcdef"
    },
    "override-values": {
        "image.tag": "latest"
    }
}
```

CNF Migration (2)

Migrate to another namespace

```
"migrate-namespace": {
  "steps": {
    "config-setup": {
      "description": "Gather necessary input",
      "target": "config-setup-process",
      "activities": [
        {
          "call_operation": "ResourceResolutionComponent.process"
        }
      ],
      "on_success": [
        "profile-upload"
      ]
    },
    "profile-upload": {
      "description": "Generate and upload K8s Profile",
      "target": "k8s-profile-upload",
      "activities": [
        {
          "call_operation": "K8sProfileUploadComponent.process"
        }
      ],
      "on_success": [
        "change-namespace"
      ]
    }
```

```
"change-namespace": {
  "description": "Change namespace script",
  "target": "change-namespace",
  "activities": [
    {
      "call_operation": "ComponentScriptExecutor.process"
    }
  ],
  "on_success": [
    "status-verification-script-after"
  ]
},
"status-verification-script-after": {
  "description": "Simple status verification script",
  "target": "simple-status-check",
  "activities": [
    {
      "call_operation": "ComponentScriptExecutor.process"
    }
  ],
  "on_success": [
    "collect-results"
  ]
}
```

CNF Migration (3)

LFN Developer & Testing Forum

Migrate to another cluster

```
"change-cluster": {
  "steps": {
    "config-setup": {
      "description": "Gather necessary input",
      "target": "config-setup-process",
      "activities": [
        {
          "call_operation": "ResourceResolutionComponent.process"
        }
      ],
      "on_success": [
        "update-tenant-in-aai"
      ]
    },
    "update-tenant-in-aai": {
      "description": "Update tenant Info in AAI",
      "target": "resource-assignment",
      "activities": [
        {
          "call_operation": "ResourceResolutionComponent.process"
        }
      ],
      "on_success": [
        "change-cluster"
      ]
    },
```

```
"change-cluster": {
  "description": "Change cluster script",
  "target": "change-cluster",
  "activities": [
    {
      "call_operation": "ComponentScriptExecutor.process"
    }
  ],
  "on_success": [
    "status-verification-script-after"
  ]
},
"status-verification-script-after": {
  "description": "Simple status verification script",
  "target": "simple-status-check",
  "activities": [
    {
      "call_operation": "ComponentScriptExecutor.process"
    }
  ],
  "on_success": [
    "collect-results"
  ]
}
```

AAI vs Prometheus Metrics (1)

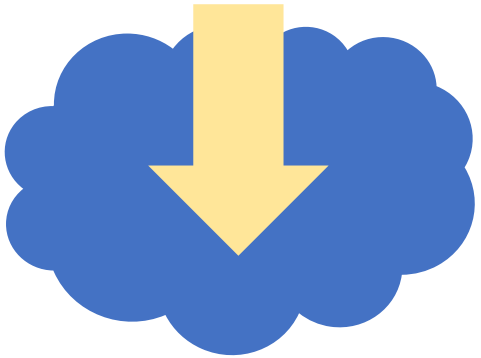
- Received Prometheus metric needs relation with AAI object
  - Correlation of metrics with affected services
  - Root-cause analysis
  - Service PM
- k8s-resource required to find service, generic-vnf or vf-module
- We need to calculated ID base on the metadata of Promrtheus metrics
  - Heat-Stack-ID/InstanceID needs to come in the metric's label
  - K8s Resource name, namespace and GVK needs to come in the metric's labels
  - If we have Prometheus instance per cluster, it is easy to find tenant identifiers in AAI

AAI vs Prometheus Metrics (2)

- k8s-resource ID is calculated from values available to client
- SHA 256 of a concatenation of the following values
  1. K8s RB Instance ID (vf-module/heat-stack-id param or label in resource in K8s)
  2. K8s resource name
  3. K8s namespace
  4. K8s resource kind
  5. K8s resource group
  6. K8s resource version
  7. CloudOwnerName (From AAI, for tenant in which CNF is deployed)
  8. CloudRegionName (From AAI, for tenant in which CNF is deployed)
  9. TenantId (From AAI, for tenant in which CNF is deployed)

(Public) Cloud Deployment

- K8sPlugin stores required kubeconfig file
- K8sPlugin does not support **exec** in kubeconfig file
- Public clouds do not provide kubeconfig files with permanently valid credentials
- We need to maintain in k8splugin a valid kubeconfig file
  - We push new kubconfig file systematically
  - We recommend to add sidecar container to multicloud-k8s deployment
  - Sidecar will update the kubeconfig file base on the cloud specific kubeconfig generation way



Exemplary Use Cases

LF NETWORKING
LFN Developer & Testing Forum

Apache

- ❑ Day 0/1/2 C(N)F
- ❑ Full automation in ONAP
- ❑ Standard K8S Cluster
- ❑ ONAP Istanbul+
- ❑ **The most complete**

DTF January 2022

Smoke Use Case

vFW CNF

- ❑ Day 0/1/2 CNF
- ❑ Full automation in ONAP + Postman
- ❑ Required KUD K8S Cluster
- ❑ ONAP Guilin+
- ❑ Used to validate CNFO
- ❑ **The best documentation**

DTF June 2021 Video

Free5GC

- ❑ Day 0/1/2 CNF + PNF
- ❑ Full automation in ONAP
- ❑ Required Dedicated K8S Cluster
- ❑ ONAP Istanbul+
- ❑ **CNF + PNF + Coordination**

DTF January 2022

Automation for CNF Use Cases

LFN Developer & Testing Forum

https://github.com/onap/demo/tree/master/tutorials/ApacheCNF

- Python ONAP-SDK Based
- Templates Folder
  - Build and Test CBA
  - Build VSP with make
- Automation Folder
  - Step-by-step README
  - Create K8S Region
  - Onboard Service
  - Instantiate Service
  - Delete Service
  - Check Health of CNF

DTF January 2022

Smoke Use Case

```
.
├── README.txt
├── automation
│   ├── Pipfile
│   ├── Pipfile.lock
│   ├── README.md
│   ├── __init__.py
│   ├── artifacts
│   ├── config.py
│   ├── crds
│   ├── create_cloud_regions.py
│   ├── delete.py
│   ├── healthcheck.py
│   ├── instantiate.py
│   ├── k8s_client.py
│   ├── onap_settings.py
│   ├── onboard.py
│   ├── update_cba.py
│   ├── update_connectivity_info.py
│   └── vsp
└── templates
    ├── Makefile
    ├── README.txt
    ├── base_native
    ├── cba
    ├── cba-dd.json
    ├── cba-dev
    ├── cba2dd.py
    ├── helm
    ├── native_cnf_k8s_demo.zip
    ├── package_native
    └── tools
```

K8S & Helm Requirements

- Kubernetes
  - Cluster must support v1.19 API
  - Image repository managed outside ONAP
  - Authentication through the static kubeconfig file
    - No exec command support in kubeconfig file
- Helm
  - Helm 3.5 used
  - No chart repository support – chart dependencies must be embedded as subcharts
  - One Helm Chart instance = only one k8s namespace
  - Helm create/delete/**upgrade** hooks supported
  - **Helm Upgrade Supported**

Jakarta
REQ-890

Future Steps – Kohn++

- CNF Upgrade SO Workflow Finalization (Kohn)
  - Complete CNF Upgrade SO Workflow
  - Helm Upgrade with pre- and post-deploy configuration
  - Migration also possible
- Integration with external k8s orchestrators
- Policy/DCAE support for CNF k8s-resource mappings
- K8S Cluster Homing / Selection
- Utilization of ASD and new information hold there
- Modification in ApacheCNF tutorial to include presented scenarios ☺

Invitation

CNFO – EMCO Integration Demo

3 PM UTC | ONAP 1 | ARIANE 1

LFN Developer & Testing Forum