

How to Enable SMO through Nephio, a Perspective

Seshu Kumar Mudiganti, Principal Technologist (WindRiver)

Ravi Ravindran, Architect (F5)

(ORAN-SC Workshop/ONE Summit, April 29, 2024)

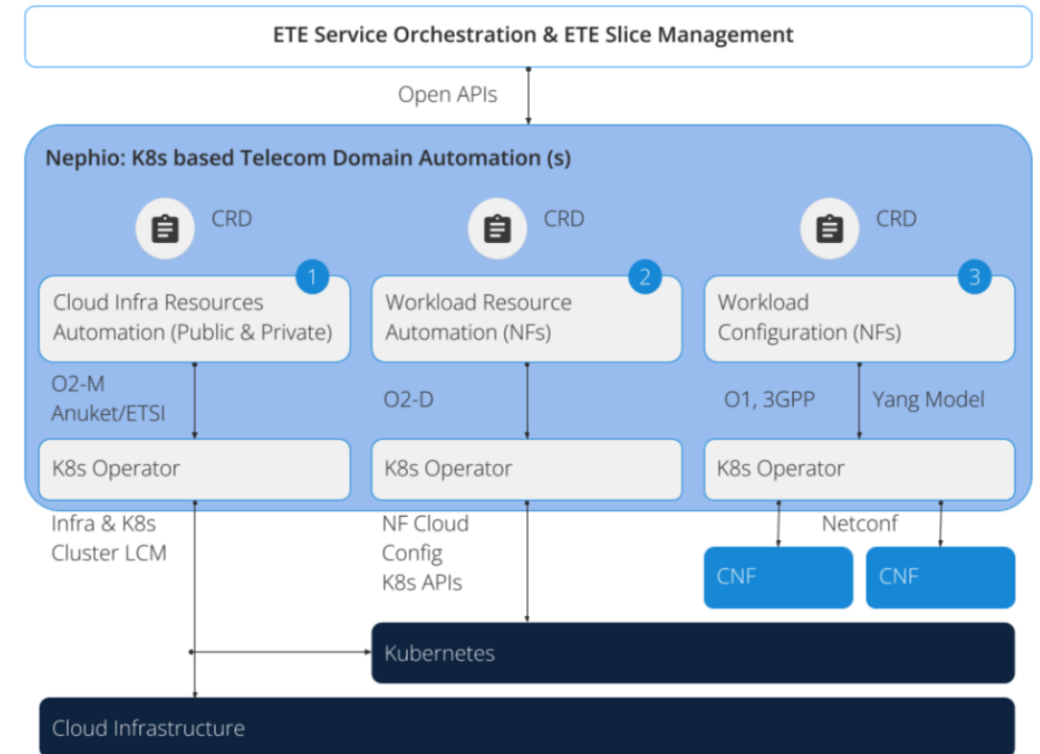
* Slides also reference material presented during R2 Developer Summit and Sig-1 Arch/WG-2

Agenda

- Nephio Problem Scope
- Nephio R2 Architecture
- Nephio R2 Features
- O-RAN/Nephio Integration Architecture
- Realizing O-RAN Use Cases over Nephio
 - NF Orchestration using SMO
 - Create O-Cloud K8S Cluster
 - Deploy Near-RT RIC without SMO
 - Reconfigure O-RAN NF

Nephio Problem Scope

- **Developing a Centralized Domain specific Orchestration K8S based Platform**
 - Domain – Telco (5G/6G etc.) , Enterprise, Transport
 - Telco – vRAN/O-RAN, 5GC use case
 - Enterprise scenarios are still under discussion
- **Many 1000x of Cloud Sites with Infrastructure**
 - Multiple Operator Personas, Cloud/NF Vendors
 - Compute, Storage, Networking Resources
 - How to manage the Infrastructure/Clusters at scale ?
 - SW Management of these Clusters
 - How to manage NF deployments at Scale ?
 - E.g 100 NF*100 parameters/NF * 10K Clusters/NF – the goal is to automate this deployment
 - Workload configuration to enable Services
- **Design Principles**
 - Intent Driven Automation based on K8S
 - Design to Scale
 - Configuration as Data
 - Accommodate Standards such as O-RAN, 3GPP etc.



Nephio R1 & R2 Releases

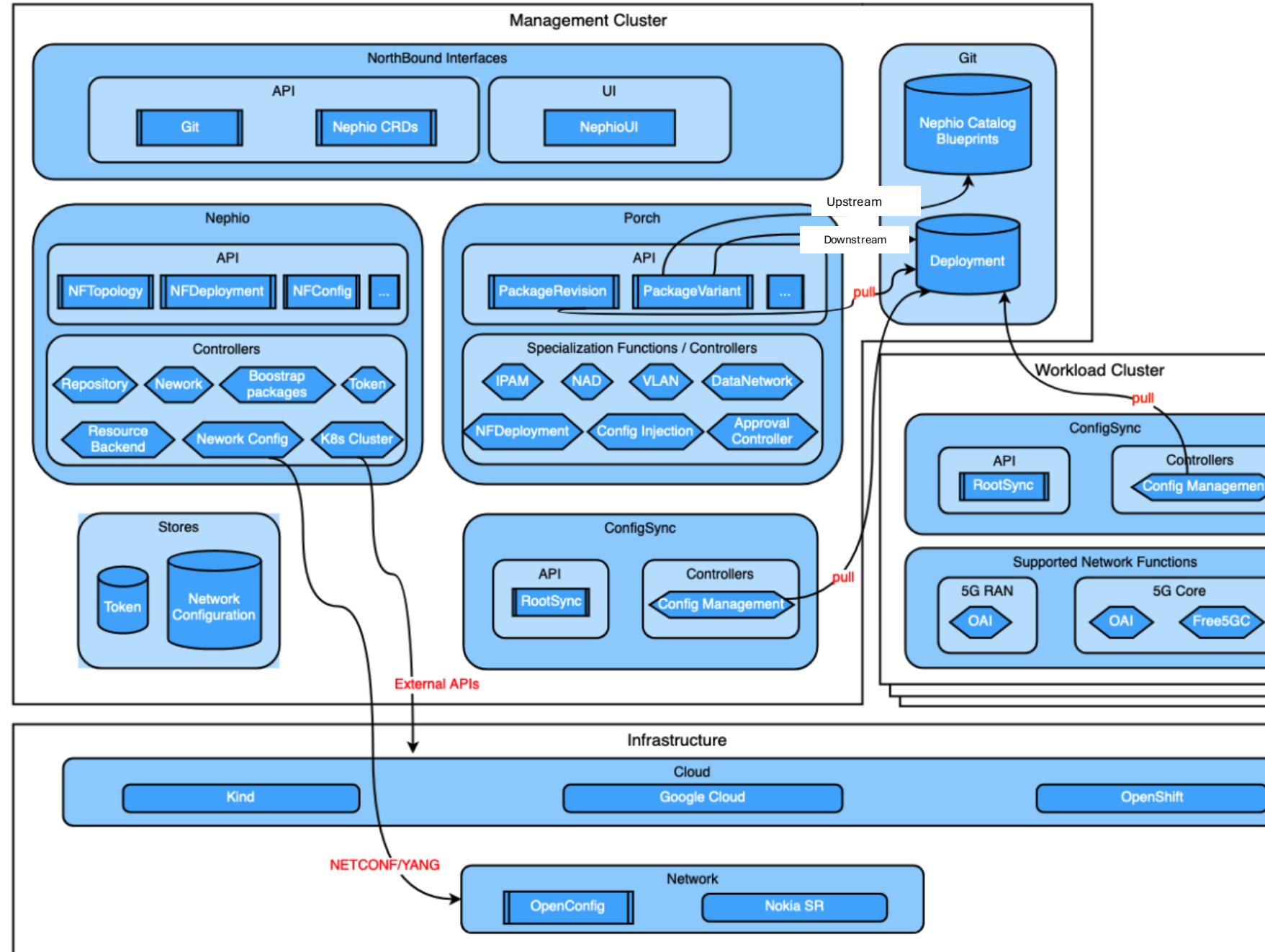
- **R1 release last November/23**
 - Demonstrated Nephio design principles on Package Specialization, Fanout for Workload cluster and 5GC NF deployments using free5gc
- **R2 Release, March/24, generalized the platform for multi-vendor NF and Cloud Provider Orchestration**
 - Multi-Vendor NF Orchestration (OAI & Free5gc)
 - Multi-Vendor Workload Cluster Orchestration (Kind, GCP & Redhat)
 - End-to-end 5G Orchestration (vRAN + 5GC)
 - Topology Controller (Service API for SO)
 - New Documentation (<https://docs.nephio.org/docs/>)
- **R3 Release to realize O-RAN use cases and many others**
 - Nephio R3 Developer Summit (D&TF , May 2nd)

Nephio R2 Architecture

- The functional components are the similar to R1 – PORCH, Nephio KRM functions/Controllers, Nephio CRs, ConfigSync
- R2 generalized the Nephio CRs to be Multi-vendor for 5G Workloads and generic extensions to handle Vendor specific configuration parameters for the NFs.
- There is separation for PORCH from Nephio, as they are expected to evolve separately in Nephio
- Nephio as such is application agnostic, in the sense, that the control plane machinery can be used be used for 5G Core, vRAN provisioning etc.

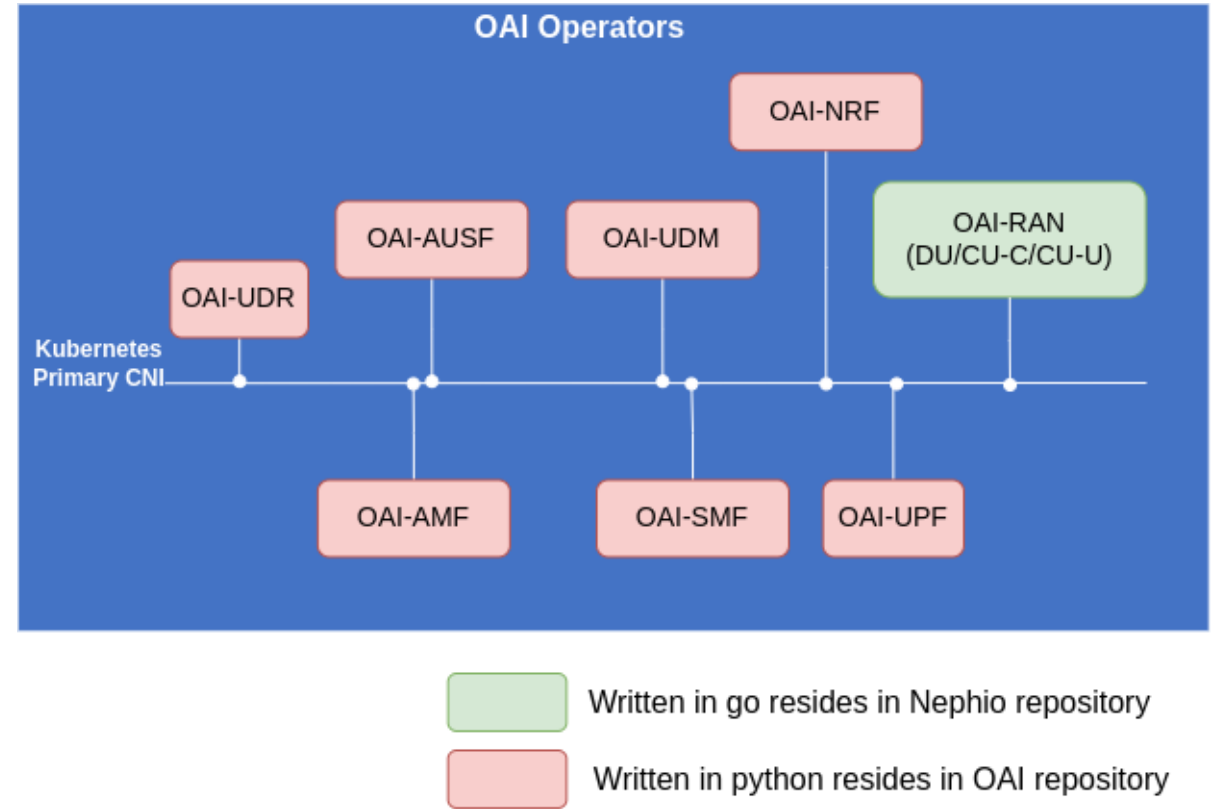
<https://docs.nephio.org/docs/architecture/> (Architectural Views)

Nephio R2 Architecture



Operator Based NF Deployment

- **Operators are Application Specific controllers in the Workload Clusters that receives Nephtio's "NF Deployment" CR and generates K8S Deployment Artifacts**
- **OAI-RAN operators code bootstrapped by using Helm to Operator code generator SDK.**
 - Enhancements to enable Nephtio approach to perform IP allocations, interfaces handling and dependency handling.
 - Currently one controller to deploy and uninstall OAI CU-CP, CU-UP, DU
- **Core network function operators are written in Python using KOPF framework**



RAN Custom Resources

OAI DU kpt package

- apply-replacements-namespace.yaml
- apply-replacements-owner.yaml
- capacity.yaml
- cm-namespace.yaml
- dependency.yaml**
- interface-flc.yaml**
- interface-flu.yaml
- Kptfile
- network_vpc-ran-internal-flc.yaml**
- network_vpc-ran-internal-flu.yaml
- nfdeployment.yaml
- oai-du-edge.yaml**
- package-context.yaml
- workload-cluster.yaml

```
oai-du-edge.yaml:
apiVersion: ref.nephio.org/v1alpha1
kind: Config
metadata:
  name: oai-du-edge
spec:
  config:
    apiVersion: workload.nephio.org/v1alpha1
    kind: OaiProviderParams
    metadata:
      name: oai-provider-param
      namespace: oai-config
    spec:
      physicalCellId: 0
      cellIdentity : '12345678L'
      Image: <container-image><tag>
```

Example of CRs

network_vpc-ran-internal-flc.yaml:

```
apiVersion: infra.nephio.org/v1alpha1
kind: Network
metadata:
  name: vpc-internal-flc
  annotations:
    config.kubernetes.io/local-config: "true"
spec:
  topology: nephio
  routingTables:
  - name: vpc-internal
    prefixes:
    - prefix: 172:1::/32
    - prefix: 172.1.0.0/16
  interfaces:
  - kind: bridgedomain
```

interface-flc.yaml:

```
apiVersion: req.nephio.org/v1alpha1
kind: Interface
metadata:
  name: flc
  annotations:
    config.kubernetes.io/local-config: "true"
    specializer.nephio.org/owner: workload.nephio.org/v1alpha1.NFDeployment.nf-example
    specializer.nephio.org/namespace: example
spec:
  networkInstance:
    name: vpc-internal-flc
  cniType: macvlan
  attachmentType: vlan
```

dependency.yaml:

```
apiVersion: req.nephio.org/v1alpha1
kind: Dependency
metadata:
  name: oai-du
  annotations:
    config.kubernetes.io/local-config: "true"
    specializer.nephio.org/owner: ref.nephio.org/v1alpha1.Config.oai-3gpp-param
    specializer.nephio.org/namespace: example
spec:
  packageName: oai-3gpp-param
  injectors:
  - apiVersion: ref.nephio.org/v1alpha1
    kind: Config
```

NF Deployment and Config CRDs Flow

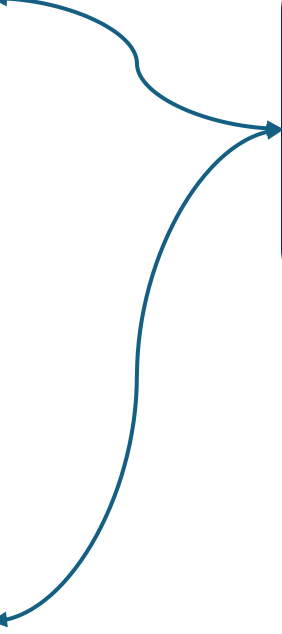
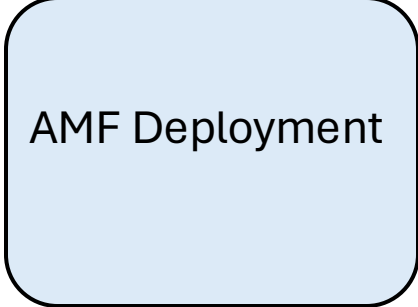
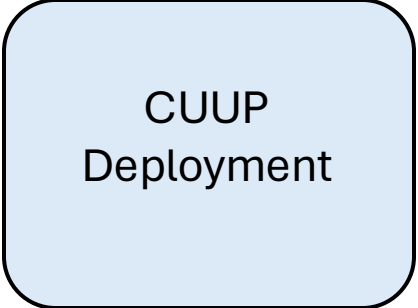
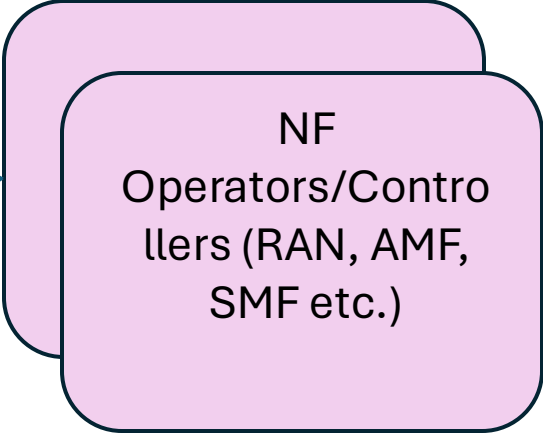
Nephio Mgmt. Cluster

Nephio Workload Cluster

*Nephio NF
CRs*

```
apiVersion: workload.nephio.org/v1alpha1
kind: NFDeployment
metadata: # kpt-merge: example/cuup-example
  name: cuup-edge
  namespace: oai-ran-cuup
  annotations:
    internal.kpt.dev/upstream-identifier: 'wor
spec:
  provider: cuup.openairinterface.org
  parametersRefs:
  - apiVersion: ref.nephio.org/v1alpha1
    kind: Config
    name: cuup-edge-cucp-regional
```

```
apiVersion: workload.nephio.org/v1alpha1
kind: NFConfig
metadata: # kpt-merge: example/cuup-nf-config
  name: cuup-nf-config
  namespace: oai-ran-cuup
  annotations:
    internal.kpt.dev/upstream-identifier: 'wor
spec:
  configRefs:
  - apiVersion: workload.nephio.org/v1alpha1
    kind: RANConfig
    metadata:
      name: ranconfig
      namespace: oai-ran-cuup
    spec:
      cellIdentity: 12345678L
      physicalCellID: 0
      tac: 1
```



NFDeployment CR

- **Package:**
 - Extendable (vendor, parameters)
- **Specialization**
 - Can be transparent
- **Operator/Controller**
 - Provider:
 - Specific namespace
 - To be validated
 - Validate existence before deploying NF
 - Validate/Process parameters for deployment
- **Namespace need to match**

```
apiVersion: workload.nephio.org/v1alpha1
kind: NFDeployment
metadata: # kpt-merge: example/cucp-example
  name: cucp-regional
  namespace: oai-ran-cucp
  annotations:
    internal.kpt.dev/upstream-identifier: 'workload.nephio.org|NFDeployment'
spec:
  provider: cucp.openairinterface.org
  parametersRefs:
  - apiVersion: workload.nephio.org/v1alpha1
    kind: NFConfig
    name: cucp-nf-config
  - name: cucp-regional-amf-core
    apiVersion: ref.nephio.org/v1alpha1
    kind: Config
  - name: cucp-regional-amf-core
    apiVersion: ref.nephio.org/v1alpha1
    kind: Config
  - name: cucp-regional-amf-core
    apiVersion: ref.nephio.org/v1alpha1
    kind: Config
```

```
networkInstances:
- name: vpc-cu-e1
  interfaces:
  - e1
- name: vpc-cudu-f1
  interfaces:
  - f1c
- name: vpc-ran
  interfaces:
  - n2
interfaces:
- name: e1
  ipv4:
    address: 172.4.2.254/24
    gateway: 172.4.2.1
  vlanID: 5
- name: f1c
  ipv4:
    address: 172.5.2.254/24
    gateway: 172.5.2.1
  vlanID: 6
- name: n2
  ipv4:
    address: 172.2.2.254/24
    gateway: 172.2.2.1
  vlanID: 4
```

Config CR

- Allows to actuate duplicate KRM each referenced by a different resource
- Created during the Package specialization based on the Dependency KRM
- E.g. Regional Cluster
 - AMF and SMF both need the same UPF parameters
 - AMF and SMF could run in a different namespace.

```
apiVersion: ref.nephio.org/v1alpha1
kind: Config
metadata:
  name: smf-regional-upf-edge01
  namespace: free5gc-cp
spec:
  config:
    apiVersion: workload.nephio.org/v1alpha1
    kind: NFDeployment
    metadata:
      name: upf-edge01
      namespace: free5gc-upf
    spec:
      ...
```

Dependency CRD

- **Intent:** defined the dependencies between CU-CP, CU-UP, and AMF for connectivity (similar as SMF one)
- **Realization:** Nephio KRM functions
 - Interfaces
 - NAD
 - NF-Deploy
- All the KRM functions used are generic and not RAN-specific, making them re-usable across use cases.
- The pipeline is supposed to be concatenation of well known KRM functions, vendor would have tested it. But then it can be manipulated by the operator during deployment.

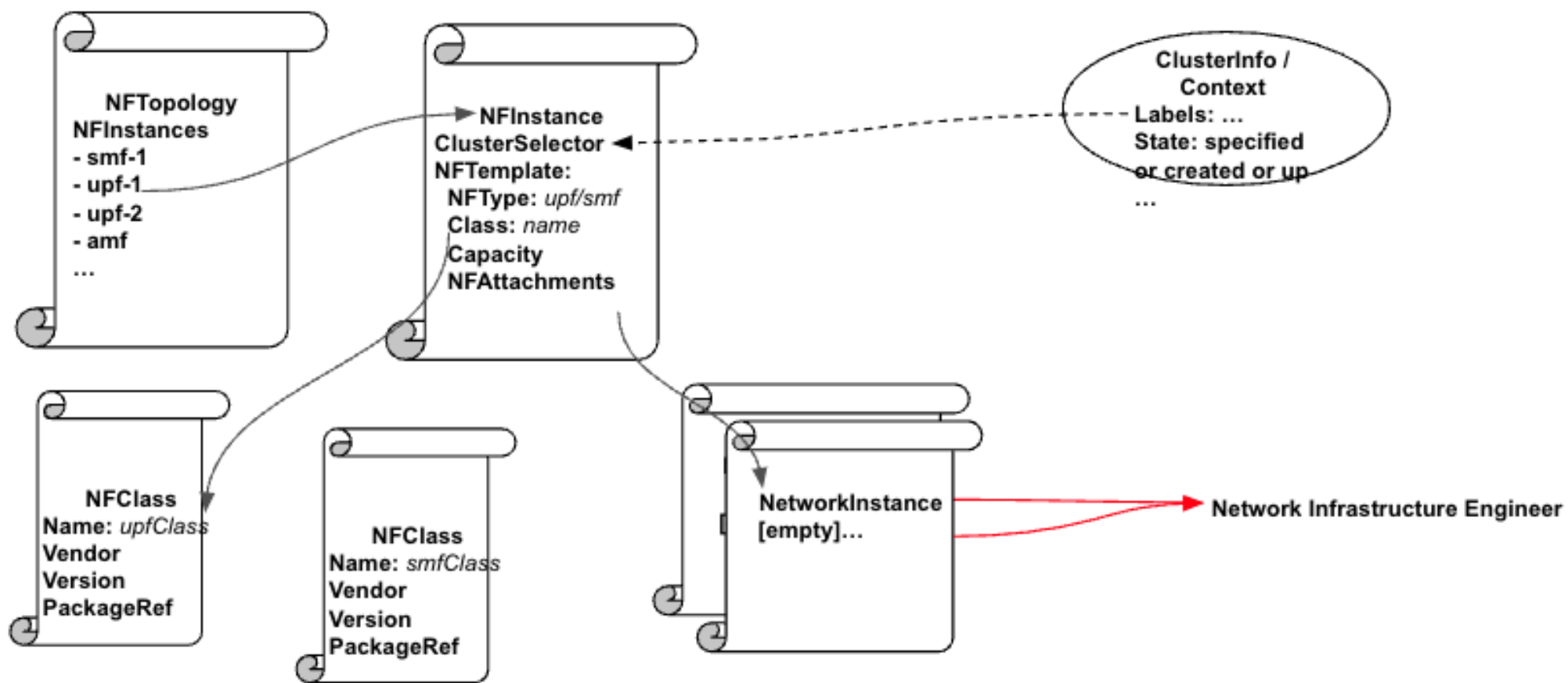
```
apiVersion: req.nephio.org/v1alpha1
kind: Dependency
metadata:
  name: CU-CP
  annotations:
    config.kubernetes.io/local-config: "true"
    specializer.nephio.org/owner:
workload.nephio.org/v1alpha1.RANDeployment.cu-up-example
specializer.nephio.org/namespace: example
spec:
  packageName: oai-amf
  injectors:
  - apiVersion: workload.nephio.org/v1alpha1
    kind: NFDeployment
```

```
pipeline:
  mutators:
  - image: gcr.io/kpt-fn/apply-replacements:v0.1.1
    configPath: apply-replacements-owner.yaml
  - image: gcr.io/kpt-fn/apply-replacements:v0.1.1
    configPath: apply-replacements-namespace.yaml
  - image: gcr.io/kpt-fn/set-namespace:v0.4.1
    configPath: cm-namespace.yaml
  - image: docker.io/nephio/nf-deploy-fn:v1.0.1
  - image: docker.io/nephio/interface-fn:v1.0.1
  - image: docker.io/nephio/nad-fn:v1.0.1
  - image: docker.io/nephio/interface-fn:v1.0.1
  - image: docker.io/nephio/nf-deploy-fn:v1.0.1
```

NF Topology CR (Experimental)

- **Define a topology oriented API (CRDs) in which users can express their NF deployment intent based on NF topology**
 - Maps NF characteristics (templates) to group of clusters (labels)
 - Defines connectivity between NFs
- **Controller to actuate the topology intent**
 - Constructs PackageVariantSet (PVS) for NF templates with capacity inputs
 - Updates deployment statuses via watching PackageRevision
- **Hasn't been integrated with current use cases yet**
- **(future) Tracks and aggregates individual NF deployment statuses via Nephio Observability framework**

NFTopology CRDs



Nephio-O-RAN Integration

Nephio-O-RAN Architecture

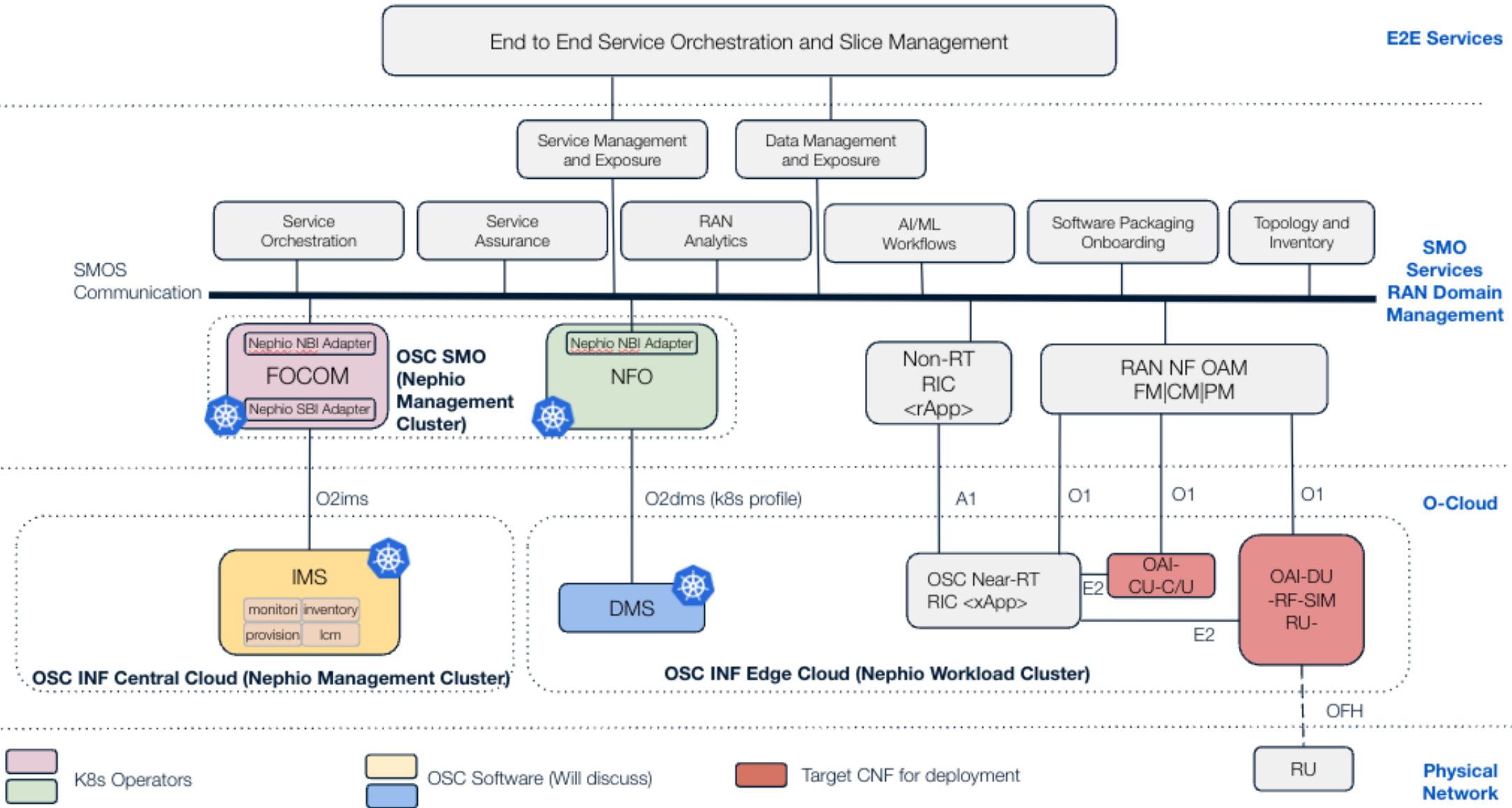
- The primary objective is to realize SMO use cases using Nephio leveraging its automation and capability to handle orchestration at scale
- R3 focusses on realizing O2 use cases, O1 use cases in the future releases
- Preserves the de-coupling of the SMO from the O-Cloud using the O2-ims in Nephio using two layers of Nephio Management
 - This preserves Nephio design and Platform features in the SMO and the O-Cloud as well, e.g.
- The NB of the Nephio/FOCOM and Nephio/NFO are the SMOS APIs that are being discussed in O-RAN

O-RAN Use Cases

- ORAN-SC to Leverage the existing functionality through the integration of Nephio and OAI
 - ETSi to use OAI CU and DU to deploy the workload through Tacker
 - Integration of Nephio with ORAN-SC INF, SMO (K8s and ETSi profiles) and other projects in future.
- NF Orchestration Using SMO
 - Use case 3.1.2 from O-RAN.WG6.O2DMS-INTERFACE-K8S-PROFILE-R003-v04.00
- Create O-Cluster K8S Cluster
 - O-RAN.WG6.ORCH-USE-CASES-R003-v07.00
- Deploy Near-RT RIC without SMO
 - Using Nephio's PV Set and GitOps approach
- Reconfigure O-RAN NF
 - Using an updated NFConfig CR and later using O1

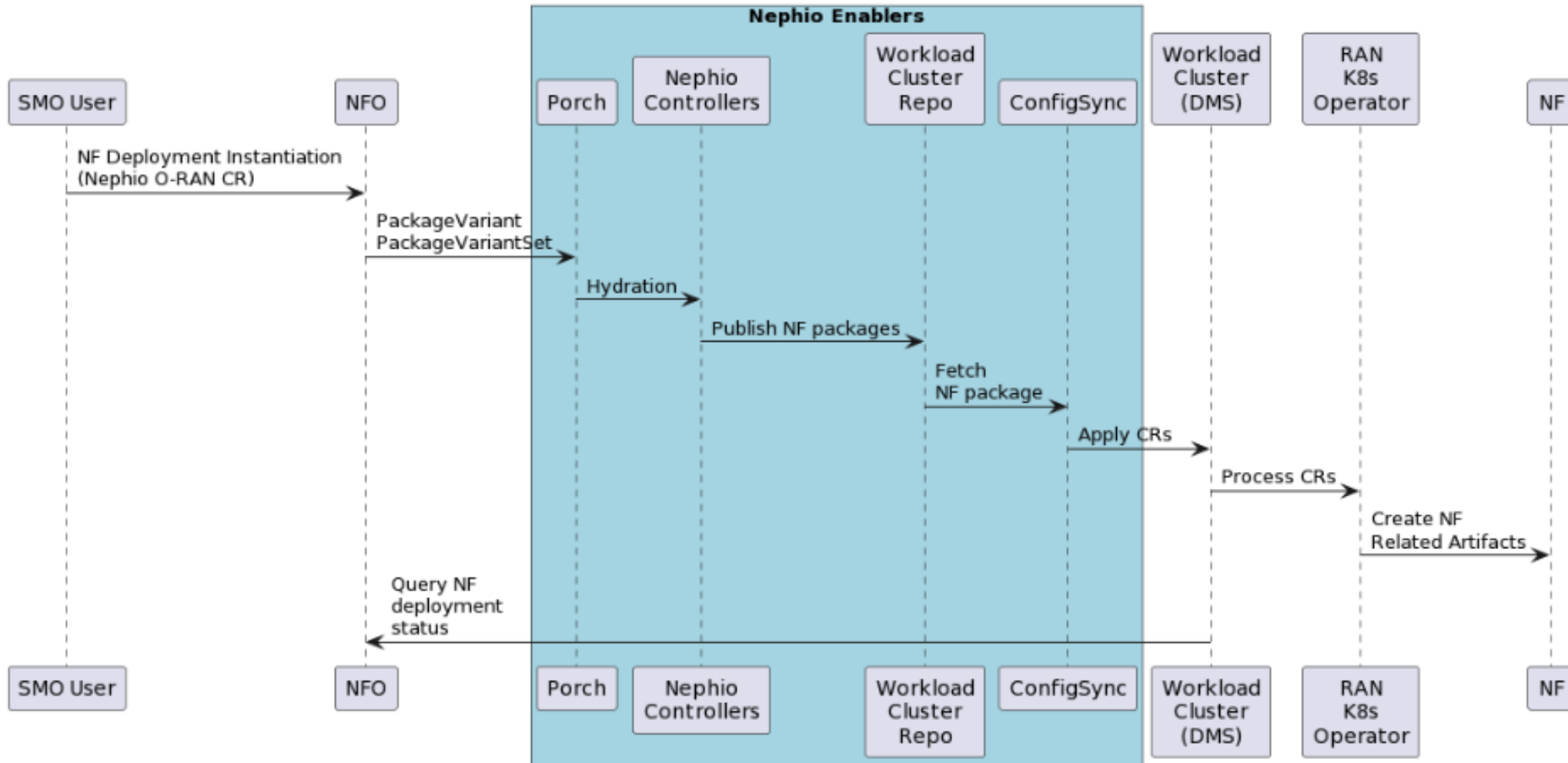
NF Orchestration Using SMO (Architecture)

O-RAN Network Function Deployment Instantiation Using SMO

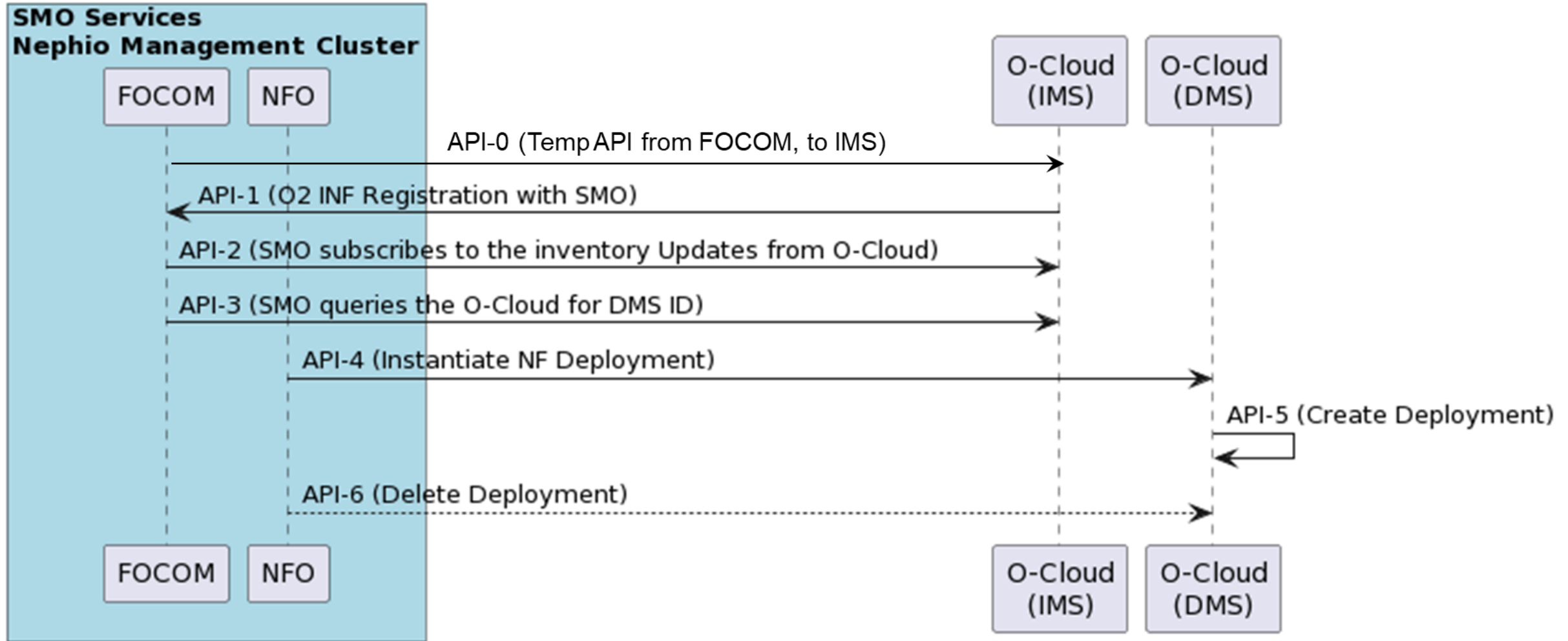


NF Orchestration using SMO

Sequence Flow Diagram (with Nephio Enablers)



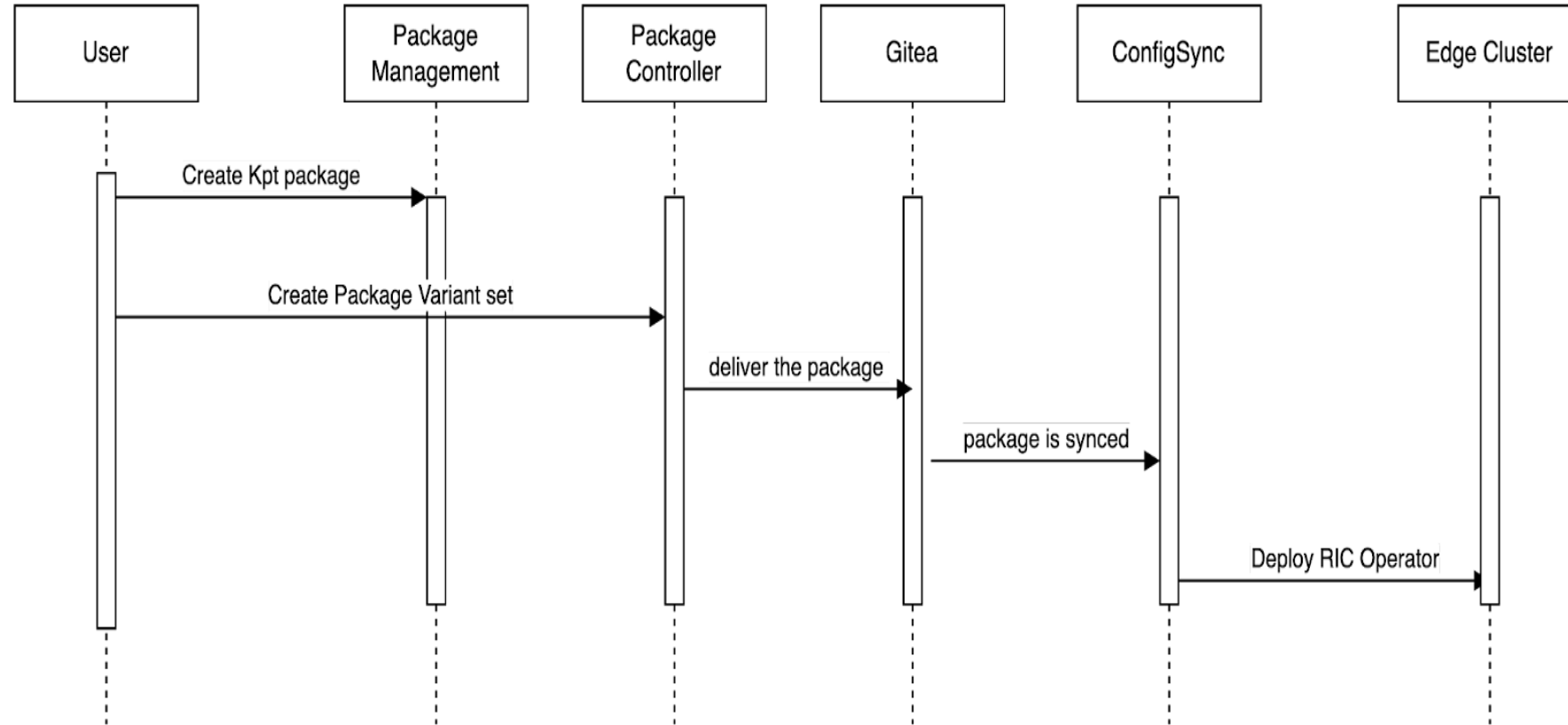
Sequence Flow Diagram for H and R3 release



Note: NBI for FOCOM, NFO, API-0 and API-3 are a stop gap solutions to accommodate the current functionalities and would be replaced with the standards as they evolve.

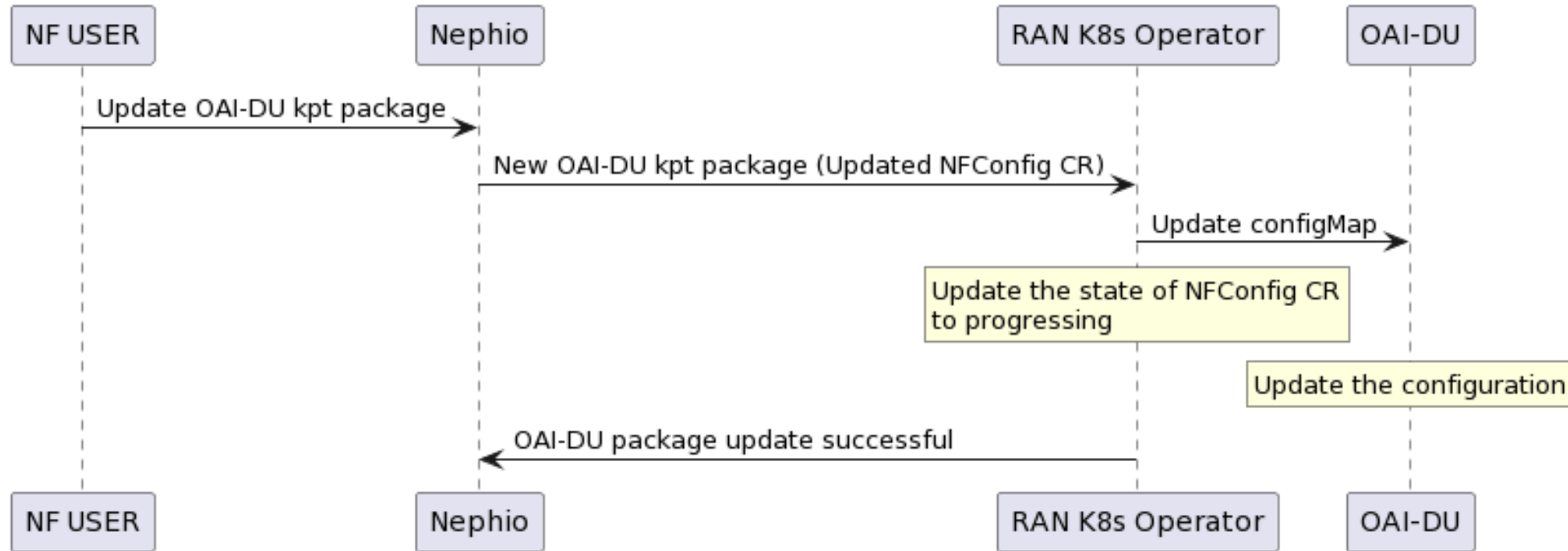
R3 and Beyond Use Cases

Deploy Near-RT RIC without SMO



- Using existing Nephio Capabilities, later to be conformant to O2ims

Reconfigure O-RAN NF



- Using existing Nepthio Capabilities, later to be conformant to O1 specifications

Create O-Cloud K8S Cluster

