Introduction to OpenAirInterface Network functions and Operators

Day 1 - October 9

Alexis de Talhouët
Solutions Architect
Red Hat

Sagar Arora
DevOps Engineer
OSA

Joseph Thaliath
Architect
Samsung

Agenda

- Introduction to OpenAirInterface (OAI)
- OAI 5G RAN and Core Network Functions
- R2 OAI Operators
- Beyond R2 for OAI and RAN support

Introduction to OpenAirInterface (OAI)

- OpenAirInterface Software Alliance (OSA)
  - Established in 2014
  - LFN member since 2018
  - French non-profit organization, funded by corporate sponsors

- OpenAirInterface (OAI)
  - Open Source Community
    - Licensed under OAI Public License V1.1
  - Build wireless cellular Radio Access Network (RAN) and Core Network (CN) functions
    - 5G 3GPP release 16 compatible
    - Reference implementation for O-RAN

OAI 5G RAN and Core Network Functions

- Offers open source software implementation of 4G/5G RAN, CN and UE
    - 5G Core: AMF, SMF, UPF, NRF, PCF, NSSF, UDR, UDM, AUSF
    - 5G RAN: DU, CU-CP, CU-UP, Monolithic gNB, Near Real-Time RIC (FlexRIC)

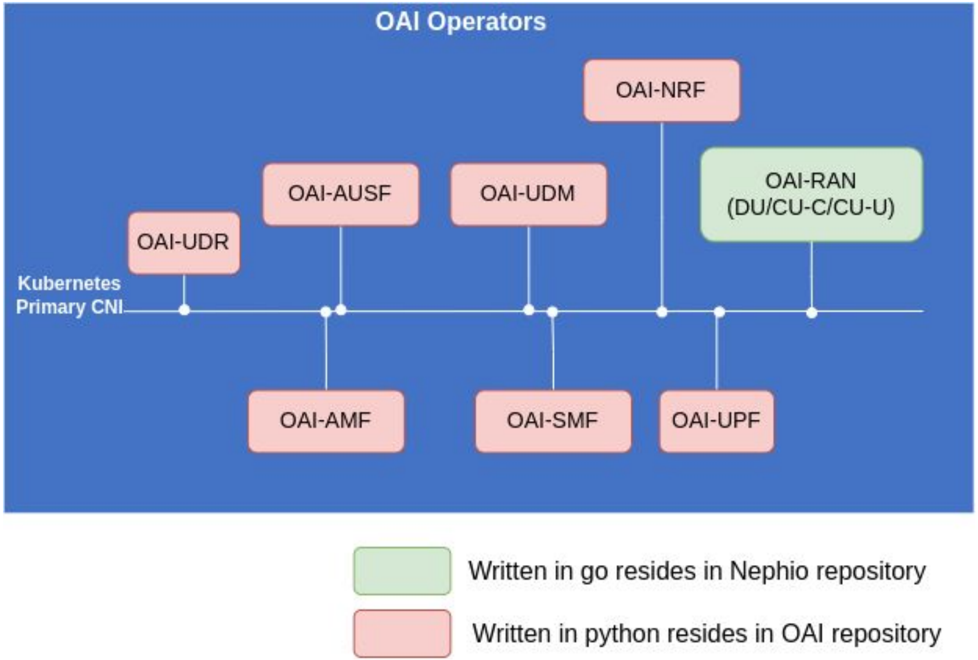
- Supports
    - Operating System: RHEL, CoreOS, Ubuntu
    - CPU Architecture: x86, aarch64
    - Container platform: Vanilla Kubernetes and Openshift

- Containerized network functions images hosted on docker hub

R2 OAI Operators

- OAI-RAN operators code bootstrapped by using the samsung helm to operator code generator sdk.
  - Enhancements to enable Nephio approach to perform IP allocations, interfaces handling and dependency handling.
  - Currently one controller to deploy and undeploy OAI CU-CP, CU-UP, DU
  - Hosted in Nephio repository under apache 2 license

- Core network function operators are written in python using kopf framework
  - Hosted in OAI github repository under 3-Clause BSD license



Beyond R2

These two points depends on Nephio (how it provides the infra)

- ➔ Including Radio Units (RUs) as infrastructure resources (Either split 8 RUs or 7.2 RUs) [Optional]
- ➔ Exposing DPDK NICs, DPUs, Hardware Accelerators for DU

These are OAI points

- ➔ Improve the RAN intent to have more 3GPP/O-RAN Oriented parameters
- ➔ Testing DU with hardware RUs
- ➔ Configuration management using Cloud-native NETCONF for RAN NF
- ➔ Developing operators for PCF and NWDAF

RAN use case: O-RAN and OpenAirInterface

Day 2 - October 10

Alexis de Talhouët
Solutions Architect
Red Hat

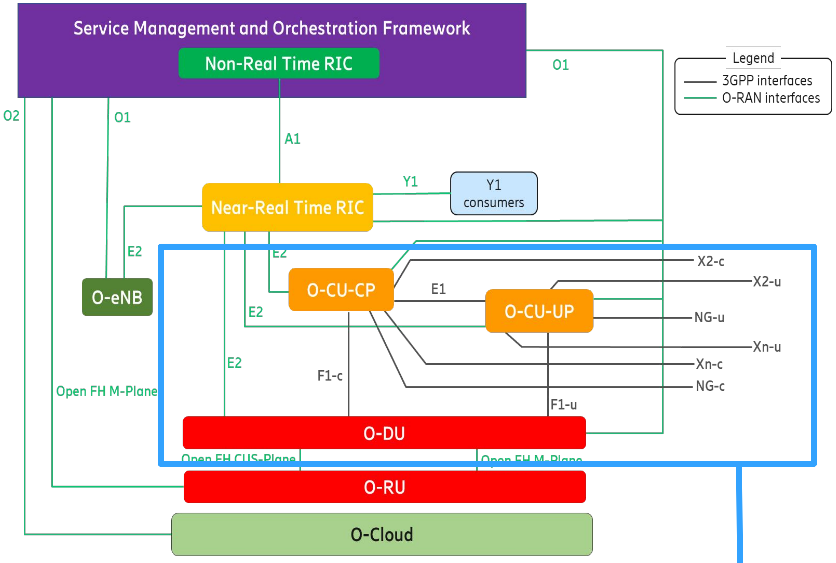
Sagar Arora
DevOps Engineer
OSA

Joseph Thaliath
Architect
Samsung

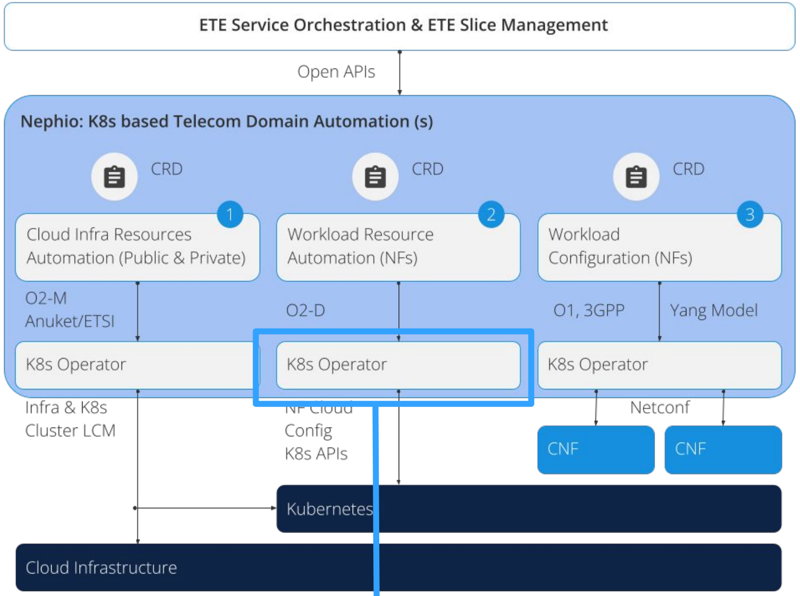
Agenda

- R2 Architecture Mapping
- R2 Deployment Blueprint
- OAI Operators (RAN + Core)
- Proposed RAN Custom Resources (CRs)

R2 Architecture Mapping



Source: O-RAN Alliance (O-RAN.WG1.OAD-R003-v10.00)


ETE Service Orchestration & ETE Slice Management
Open APIs
Nephio: K8s based Telecom Domain Automation (s)
CRD
CRD
CRD
1
Cloud Infra Resources Automation (Public & Private)
2
Workload Resource Automation (NFs)
3
Workload Configuration (NFs)
O2-M Anuket/ETSI
O2-D
O1, 3GPP
Yang Model
K8s Operator
K8s Operator
K8s Operator
Infra & K8s Cluster LCM
NF Cloud Config K8s APIs
Netconf
CNF
CNF
Kubernetes
Cloud Infrastructure

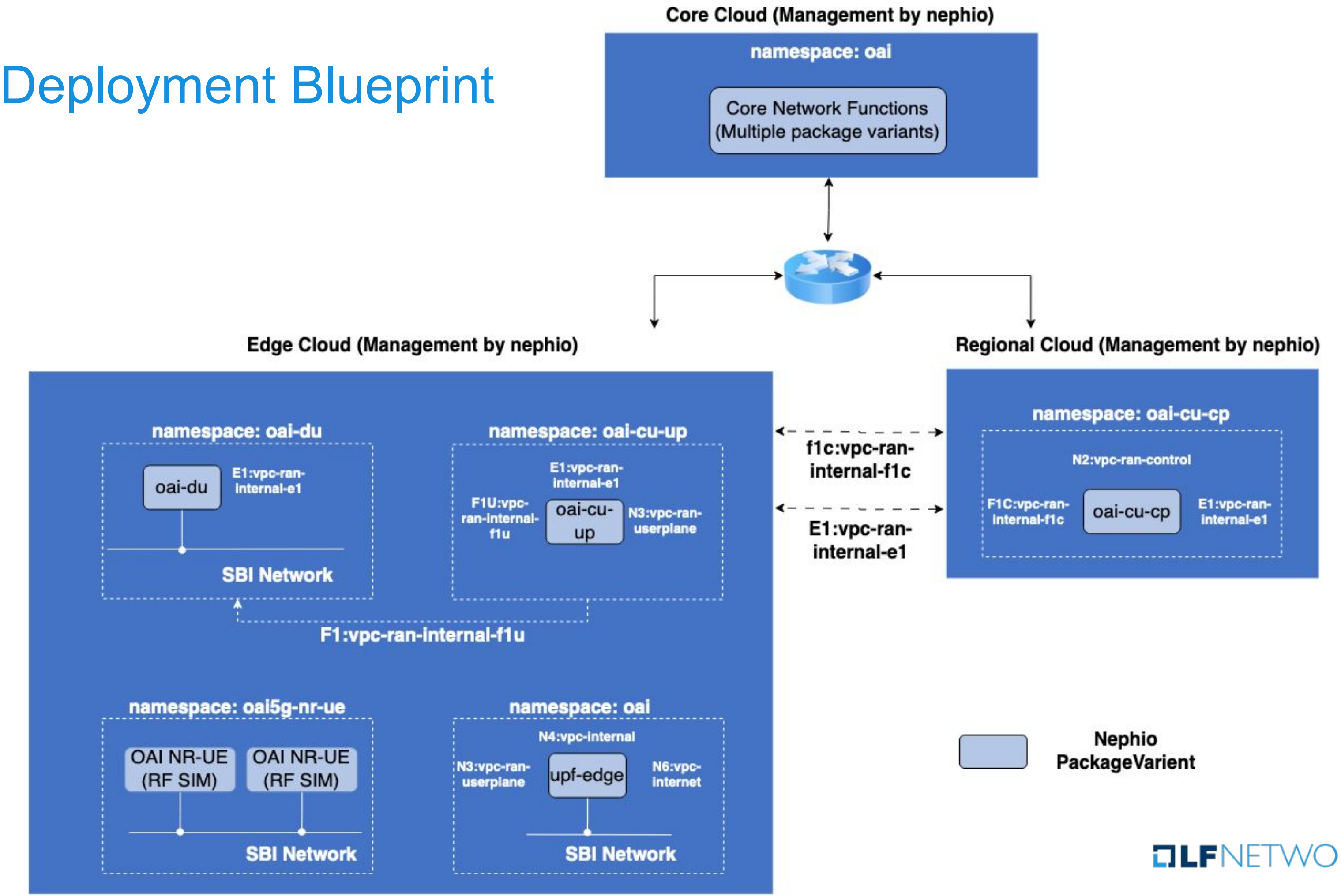
Source: Nephio Website

R2
Scope

CNFs Provided by OAI

Simplified Operators for
deploying OAI NFs
(not O-RAN 02)

R2 Deployment Blueprint



Core Cloud (Management by nephio)

namespace: oai

Core Network Functions (Multiple package variants)

Edge Cloud (Management by nephio)

namespace: oai-du

oai-du

E1:vpc-ran-internal-e1

SBI Network

namespace: oai-cu-up

E1:vpc-ran-internal-e1

F1U:vpc-ran-internal-f1u

oai-cu-up

N3:vpc-ran-userplane

F1:vpc-ran-internal-f1u

namespace: oai5g-nr-ue

OAI NR-UE (RF SIM)

OAI NR-UE (RF SIM)

SBI Network

namespace: oai

N4:vpc-internal

N3:vpc-ran-userplane

upf-edge

N6:vpc-internet

SBI Network

f1c:vpc-ran-internal-f1c

E1:vpc-ran-internal-e1

Regional Cloud (Management by nephio)

namespace: oai-cu-cp

N2:vpc-ran-control

F1C:vpc-ran-internal-f1c

oai-cu-cp

E1:vpc-ran-internal-e1

Nephio PackageVarient

R2 Deployment Blueprint

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.

Example dependency.yaml for cu-cp

```
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: AMFDeployment
```

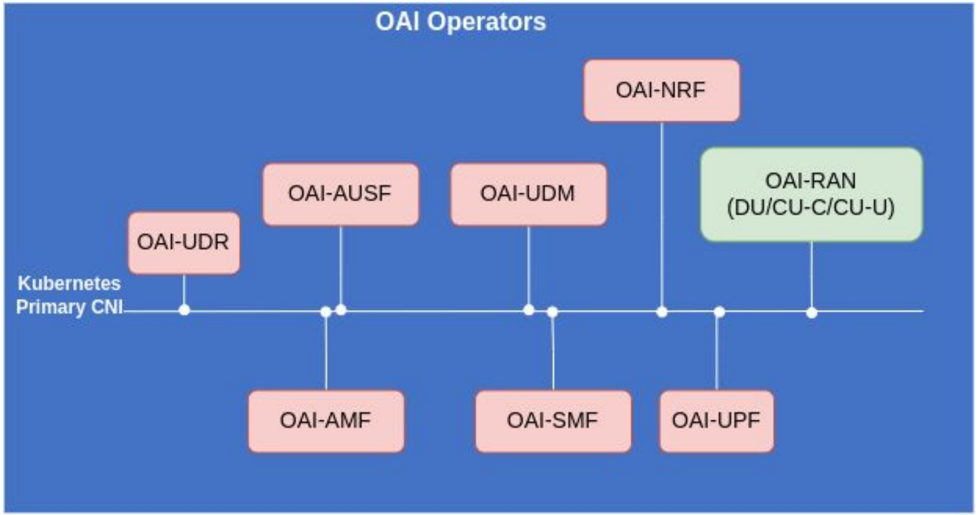
Example KRM functions

```
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
```

OAI Operators

- OAI-RAN operators code bootstrapped by using the helm to operator code generator.
  - Enhancements to enable Nephio approach to perform IP allocations, interfaces handling and dependency handling.
  - Currently one controller to deploy and undeploy OAI CU-CP, CU-UP, DU
  - Hosted in Nephio repository under apache 2 license

- Core network function operators are written in python using kopf framework
  - Hosted in OAI github repository under 3-Clause BSD license



🟩 Written in go resides in Nephio repository

🟥 Written in python resides in OAI repository

RAN Custom Resources

OAI DU kpt package

```
├── apply-replacements-namespace.yaml
├── apply-replacements-owner.yaml
├── capacity.yaml
├── cm-namespace.yaml
├── dependency.yaml
├── interface-f1c.yaml
├── interface-f1u.yaml
├── Kptfile
├── network_vpc-ran-internal-f1c.yaml
├── network_vpc-ran-internal-f1u.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: OaiVendorParams
    metadata:
      name: oai-vendor-param
```

Example of CRs

network_vpc-ran-internal-f1c.yaml:

```
apiVersion: infra.nephio.org/v1alpha1
kind: Network
metadata:
  name: vpc- internal-f1c
  annotations:
    config.kubernetes.io/l
spec:
  topology: nephio
  routingTables:
  - name: vpc-internal
    prefixes:
    - prefix: 172:1::/32
    - prefix: 172.1.0.0/16
    interfaces:
    - kind: bridgedomain
```

interface-f1c.yaml:

```
apiVersion: req.nephio.org/v1alpha1
kind: Interface
metadata:
  name: f1c
  annotations:
    config.kubernetes.io/local-config: "true"
    specializer.nephio.org/owner: workload.nephio.org/v1alpha
    specializer.nephio.org/namespace: example
spec:
  networkInstance:
    name: vpc- internal-f1c
  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
```

RAN Custom Resources

OAI 3gpp param kpt package

```
├── Kptfile
├── oai-3gpp-param.yaml
```

oai-3gpp-param.yaml:

```
apiVersion: ref.nephio.org/v1alpha1
kind: Config
metadata:
  name: oai-3gpp-param
spec:
  config:
    apiVersion: workload.nephio.org/v1alpha1
    kind: 3gppParam
    metadata:
      name: oai-3gpp-param
      namespace: oai-config
    spec:
        plmn:
          mcc: '001'
          mnc: '01'
          mncLength: 2
        tac: '1'
        nssaiList:
        - sst: '1'
          sd: '0xffffff'
```

RAN Custom Resources

Example of CRs

The whole `NFDeployment` is created by

`nfdeploy-fn` function which injects

- **capacity**
- **interfaces**
- **networkInstances**
- **parametersRefs**

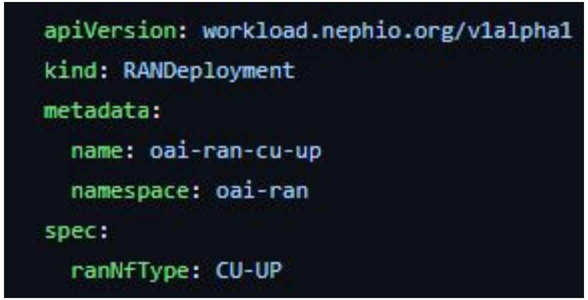
using the embedded `configinject-fn`

Final nfdeployment.yaml:

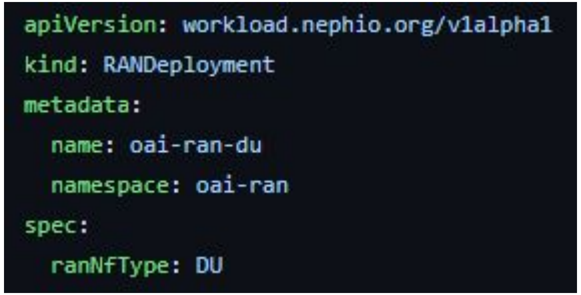
```yaml
apiVersion: workload.nephio.org/v1alpha1
kind: NFDeployment
metadata:
 name: oai-du
 namespace: oai-du
spec:
 provider: du.oai.org
 capacity:
   maxDownlinkThroughput: 100M
   maxUplinkThroughput: 1M
 interfaces:
 - name: f1-du
   ipv4:
     address: 172.21.16.100/24
     gateway: 172.21.16.254
   vlanID: 3
 - name: f1-du
   ipv4:
     address: 172.21.4.100/24
     gateway: 172.21.4.254
   vlanID: 4
 networkInstances:
 - name: vpc-ran
   interfaces:
   - n2
 - name: vpc-e1
   interfaces:
   - e1
 - name: vpc-f1c
   interfaces:
   - f1c
 parametersRefs:
 - name: oai-du-edge
   apiVersion: ref.nephio.org/v1alpha1
   kind: Config
 - name: oai-3gpp-params
   apiVersion: ref.nephio.org/v1alpha1
   kind: Config
```

Initially Proposed RAN Custom Resources

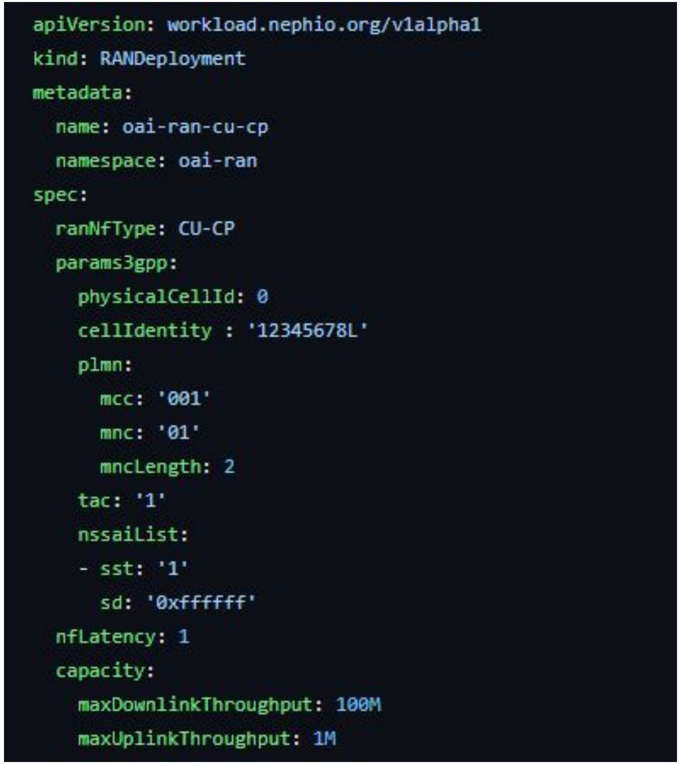
This is up for discussion and not definitive



```
apiVersion: workload.nephio.org/v1alpha1
kind: RANDeployment
metadata:
  name: oai-ran-cu-up
  namespace: oai-ran
spec:
  ranNfType: CU-UP
```

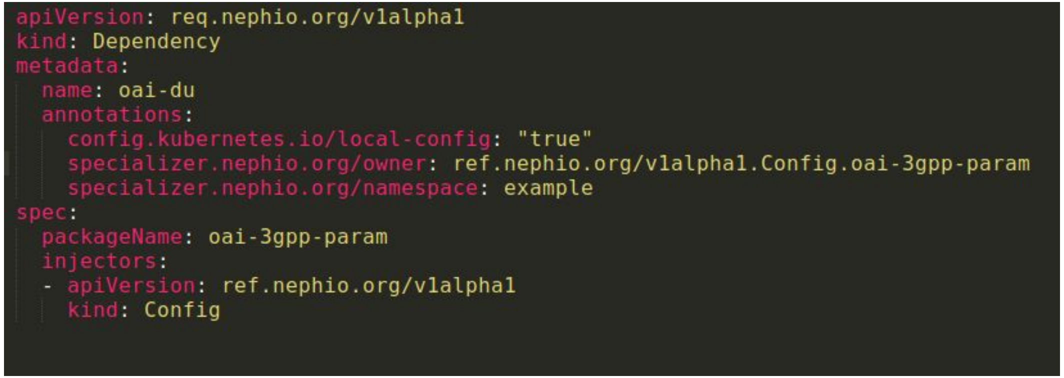


```
apiVersion: workload.nephio.org/v1alpha1
kind: RANDeployment
metadata:
  name: oai-ran-du
  namespace: oai-ran
spec:
  ranNfType: DU
```

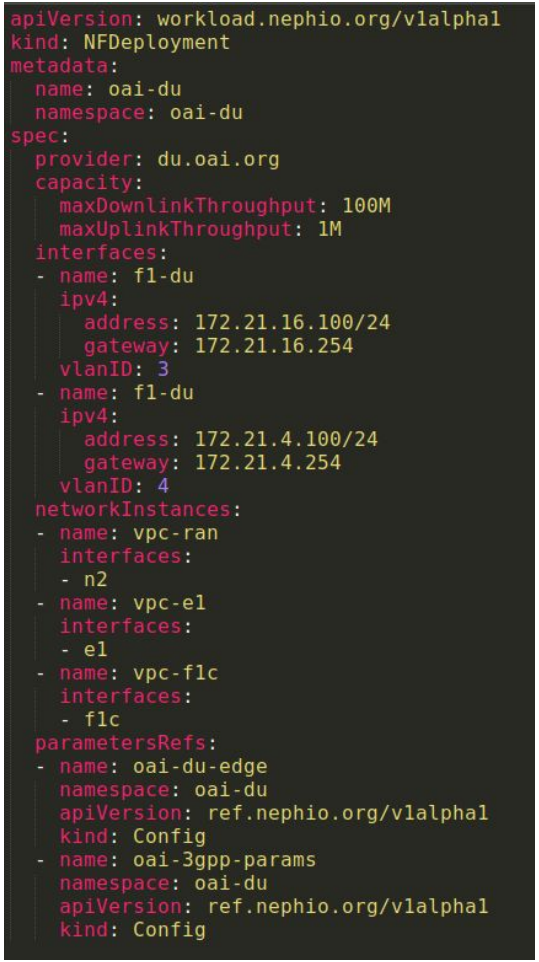


```
apiVersion: workload.nephio.org/v1alpha1
kind: RANDeployment
metadata:
  name: oai-ran-cu-cp
  namespace: oai-ran
spec:
  ranNfType: CU-CP
  params3gpp:
    physicalCellId: 0
    cellIdentity : '12345678L'
    plmn:
      mcc: '001'
      mnc: '01'
      mncLength: 2
    tac: '1'
    nssaiList:
    - sst: '1'
      sd: '0xffffff'
  nfLatency: 1
  capacity:
    maxDownlinkThroughput: 100M
    maxUplinkThroughput: 1M
```

RAN Custom Resources



```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
```



```yaml
apiVersion: workload.nephio.org/v1alpha1
kind: NFDeployment
metadata:
  name: oai-du
  namespace: oai-du
spec:
  provider: du.oai.org
  capacity:
    maxDownlinkThroughput: 100M
    maxUplinkThroughput: 1M
  interfaces:
  - name: f1-du
    ipv4:
      address: 172.21.16.100/24
      gateway: 172.21.16.254
    vlanID: 3
  - name: f1-du
    ipv4:
      address: 172.21.4.100/24
      gateway: 172.21.4.254
    vlanID: 4
  networkInstances:
  - name: vpc-ran
    interfaces:
    - n2
  - name: vpc-e1
    interfaces:
    - e1
  - name: vpc-f1c
    interfaces:
    - f1c
  parametersRefs:
  - name: oai-du-edge
    namespace: oai-du
    apiVersion: ref.nephio.org/v1alpha1
    kind: Config
  - name: oai-3gpp-params
    namespace: oai-du
    apiVersion: ref.nephio.org/v1alpha1
    kind: Config
```