LF NETWORKING

LFN Developer & Testing Forum

ONAP for xNF Based 5G Service Orchestration

Łukasz Rajewski, Michał Chabiera (Orange)
Abderaouf Khichane, Ilhem Fajjari (Orange)
Seshu Kumar (Huawei)

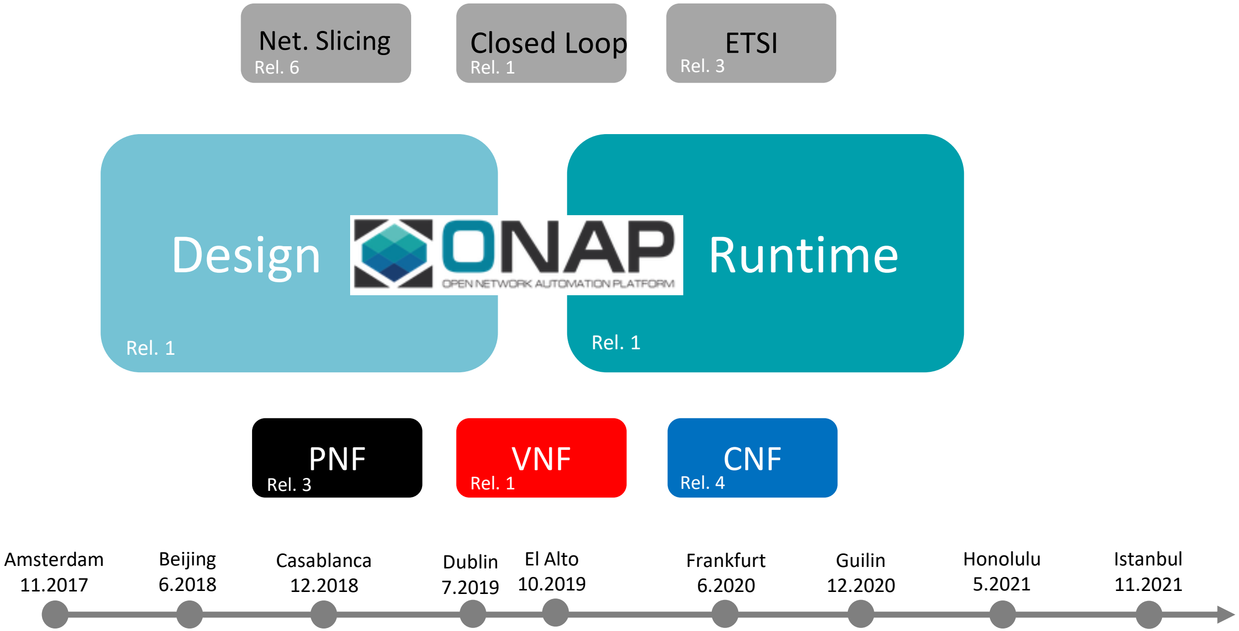
13.01.2022

LF NETWORKING

LFN Developer & Testing Forum

ONAP for xNF Orchestration

ONAP for xNF Orchestration



Net. Slicing
Rel. 6

Closed Loop
Rel. 1

ETSI
Rel. 3

Design
Rel. 1

Runtime
Rel. 1

PNF
Rel. 3

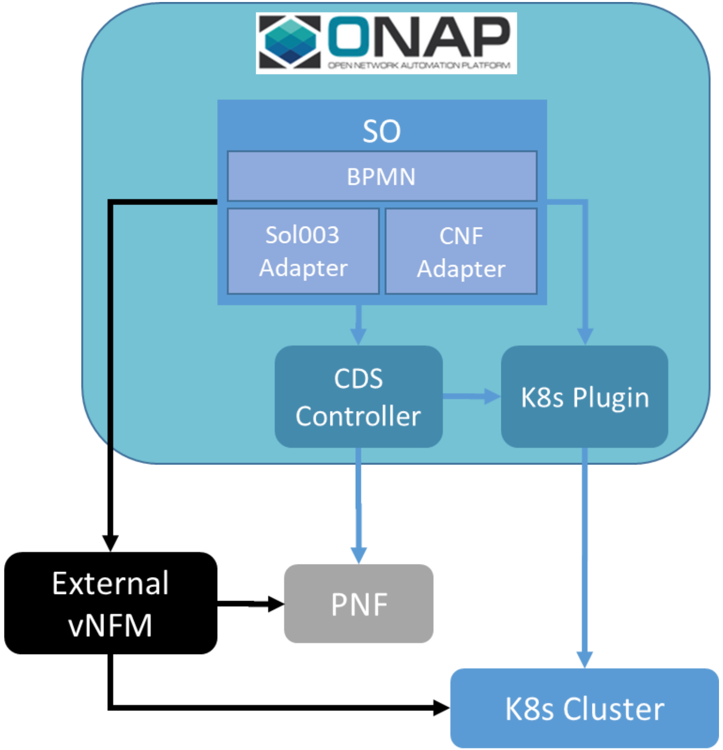
VNF
Rel. 1

CNF
Rel. 4

Amsterdam 11.2017
Beijing 6.2018
Casablanca 12.2018
Dublin 7.2019
El Alto 10.2019
Frankfurt 6.2020
Guilin 12.2020
Honolulu 5.2021
Istanbul 11.2021

ONAP Orchestration Scenarios: Native vs ETSI

- SO takes decision of the resource orchestration based on the input model
- Currently we manage the flows of the ETSI and the Native Helm as separate entities and call the flow on need basis.
- The idea is to merge them in future, based on the ETSI spec for the CNFs
- The current Demo is using the native orchestration flow for the VNF, PNF and CNF



LF NETWORKING

LFN Developer & Testing Forum

Towards5GS

Rationale

Several open-source projects are dealing with the implementation of 5G core network

magma free5GC Open5GS 5G

× None of them is providing tools to perform on Kubernetes platform

Towards5GS provides Helm charts for deploying and testing a containerized 5G service on top of Kubernetes

kubernetes HELM

5G-related software

free5GC

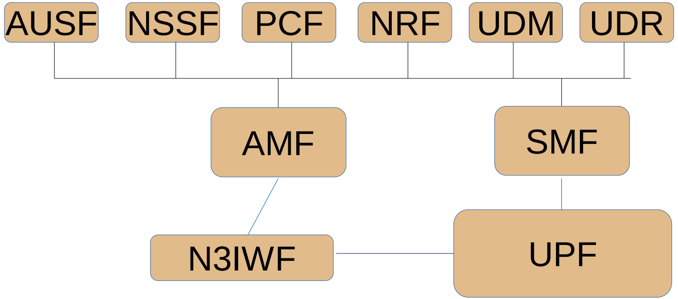
- Open source 5G core network based on 3GPP R15
- The ultimate goal of this project is to implement the 5G core network (5GC) defined in 3GPP Release 15 (R15) and beyond
- Supported versions are v3.0.4 and v3.0.5

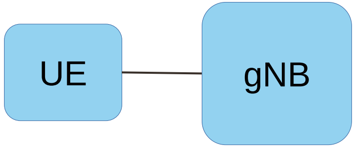
THE APACHE SOFTWARE FOUNDATION 20TH ANNIVERSARY

UERANSIM 5G SOLUTIONS

- Open source 5G User Equipment (UE) and gNodeB (gNB) implementation
- The project can be used for testing 5G Core Network and studying E2E 5G System
- The latest supported version is v3.2.4

GPLv3 Free Software





5G-related software

free5GC

- Open source 5G core network based on 3GPP R15
- The ultimate goal of this project is to implement the 5G core network (5GC) defined in 3GPP Release 15 (R15) and beyond
- Supported versions are v3.0.4 and v3.0.5

THE APACHE SOFTWARE FOUNDATION 20TH ANNIVERSARY

UERANSIM 5G SOLUTIONS

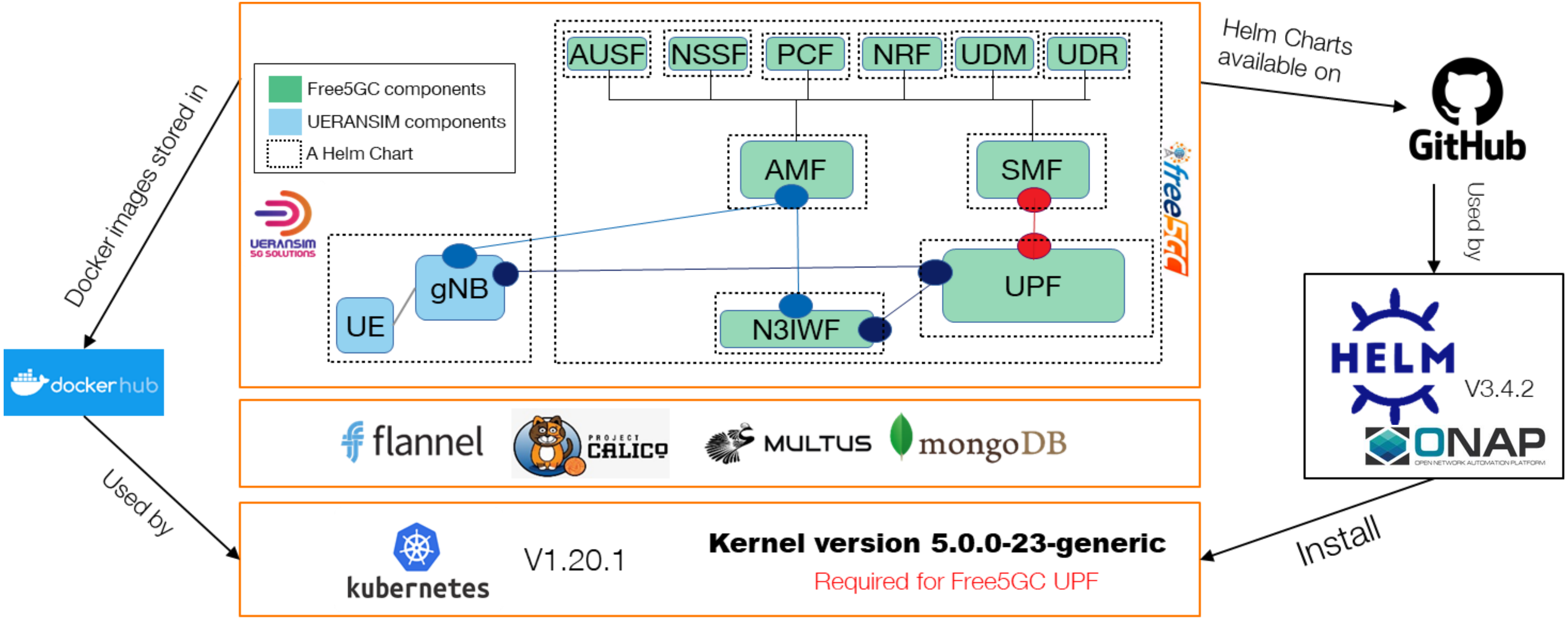
- Open source 5G User Equipment (UE) and gNodeB (gNB) implementation
- The project can be used for testing 5G Core Network and studying E2E 5G System
- The latest supported version is v3.2.4

GPLv3 Free Software

Links:

Free5GC: https://github.com/free5gc/free5gc
UERANSIM: https://github.com/aligungr/UERANSIM

Towards5GS platform



Free5GC components
UERANSIM components
A Helm Chart

AUSF
NSSF
PCF
NRF
UDM
UDR
AMF
SMF
UPF
N3IWF
gNB
UE

UERANSIM 5G SOLUTIONS
free5GC

flannel
PROJECT CALICO
MULTUS
mongoDB

kubernetes
V1.20.1

Kernel version 5.0.0-23-generic
Required for Free5GC UPF

Docker images stored in
docker hub
Used by

Helm Charts available on
GitHub
Used by
HELM
V3.4.2
ONAP
OPEN NETWORK AUTOMATION PLATFORM
Install

Links:

Github Repo: https://github.com/Orange-OpenSource/towards5gs-helm
Dockerhub Organization: https://hub.docker.com/u/towards5gs
Bitnami's MongoDB Helm Chart:
https://github.com/bitnami/charts/tree/master/bitnami/mongodb

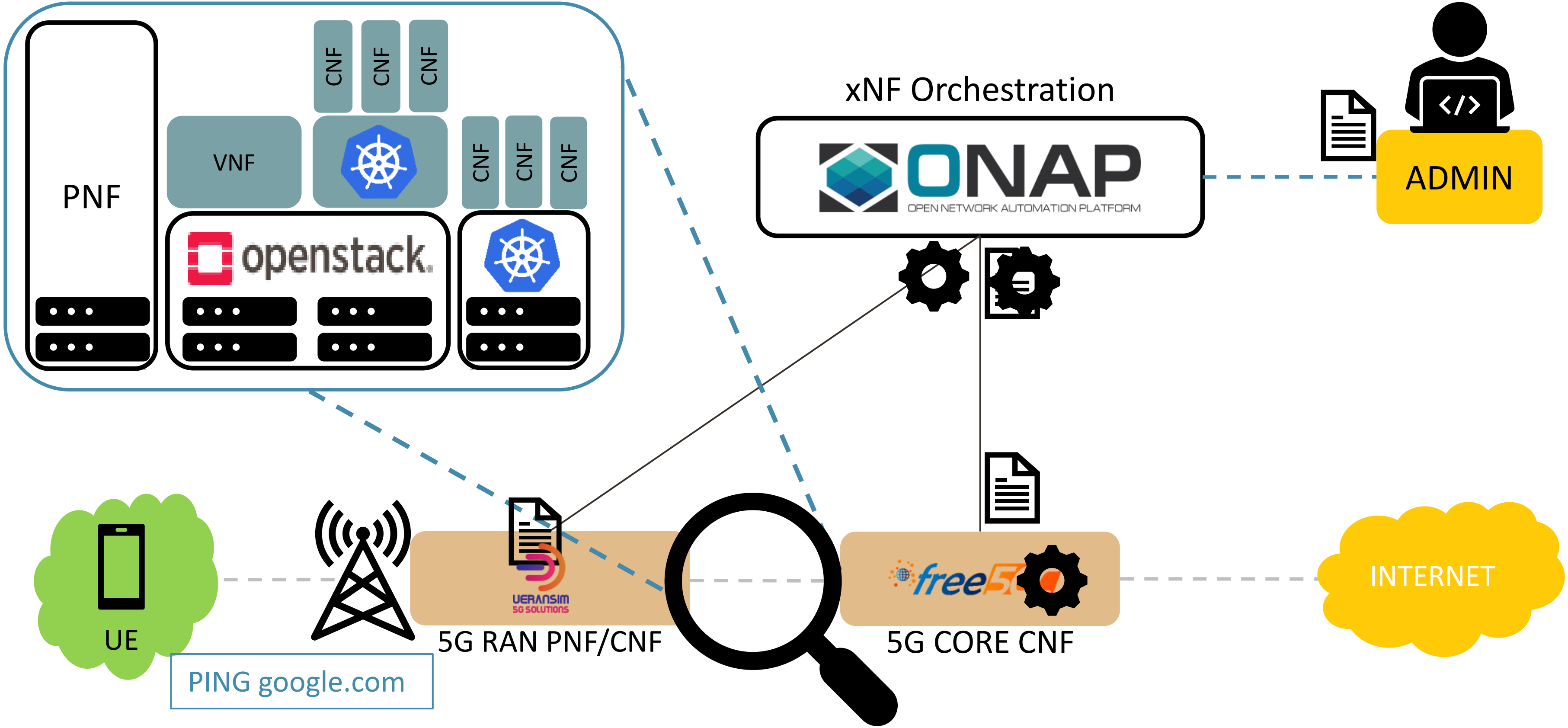
LF NETWORKING

LFN Developer & Testing Forum

Orchestration of Free5GC using ONAP

ONAP for xNF Based 5G Service Orchestration

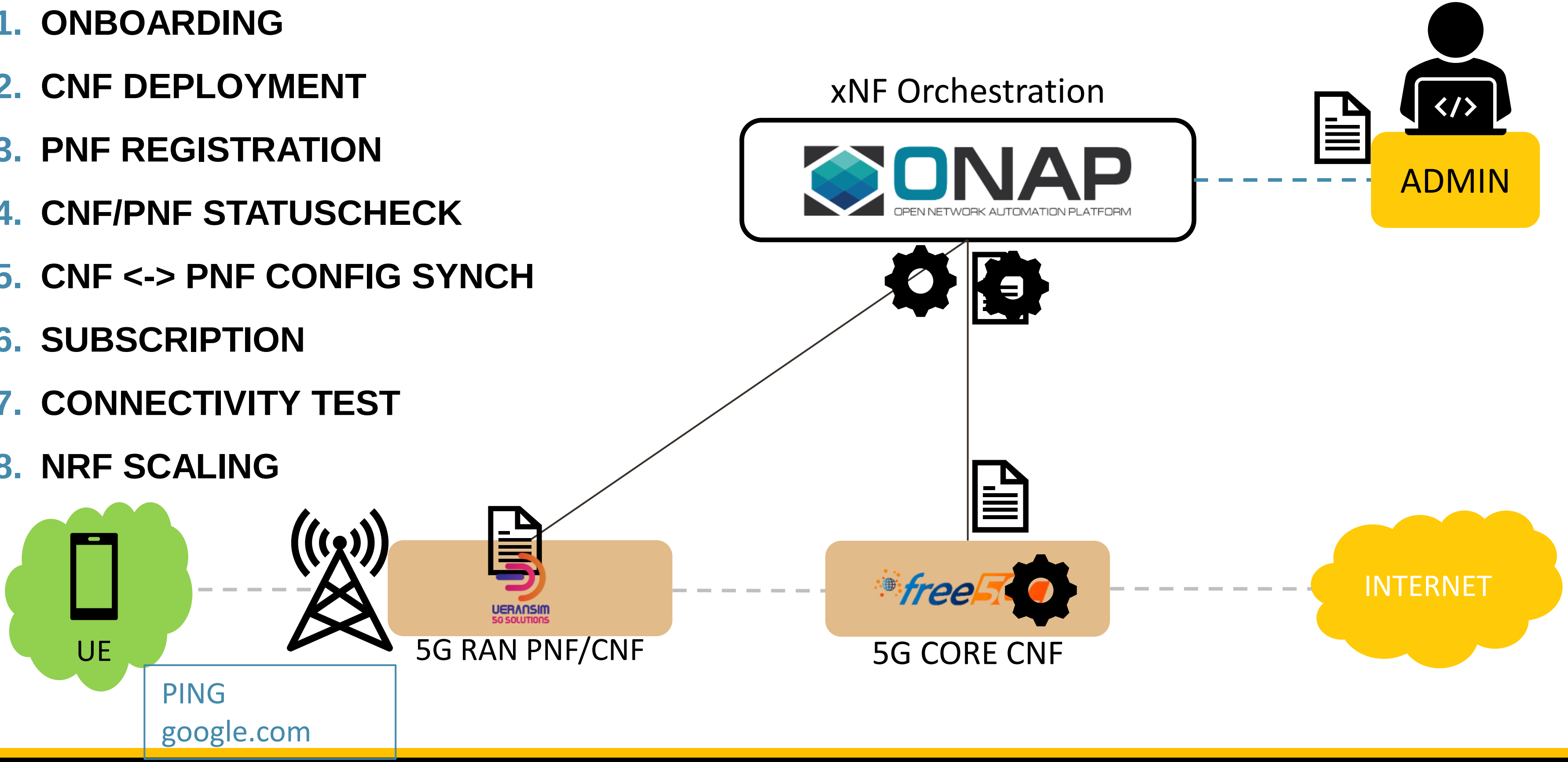
LF NETWORKING
LFN Developer & Testing Forum



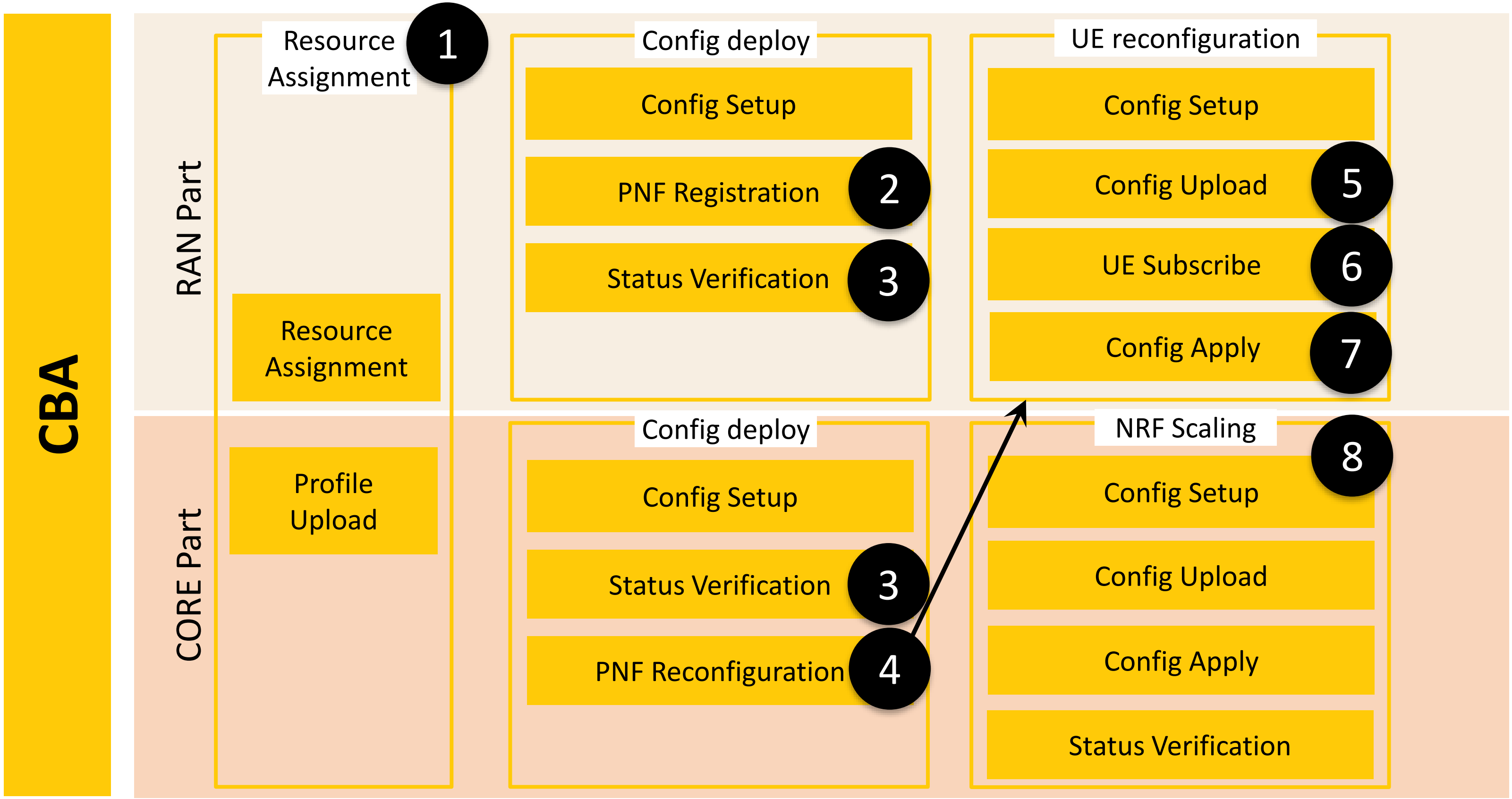
ONAP for xNF Based 5G Service Orchestration

LF NETWORKING
LFN Developer & Testing Forum

1. ONBOARDING
2. CNF DEPLOYMENT
3. PNF REGISTRATION
4. CNF/PNF STATUSCHECK
5. CNF <-> PNF CONFIG SYNCH
6. SUBSCRIPTION
7. CONNECTIVITY TEST
8. NRF SCALING



Joint CBA for RAN and Core Part NF



CBA

RAN Part

Resource Assignment 1
- Resource Assignment

Config deploy
- Config Setup
- PNF Registration 2
- Status Verification 3

UE reconfiguration
- Config Setup
- Config Upload 5
- UE Subscribe 6
- Config Apply 7

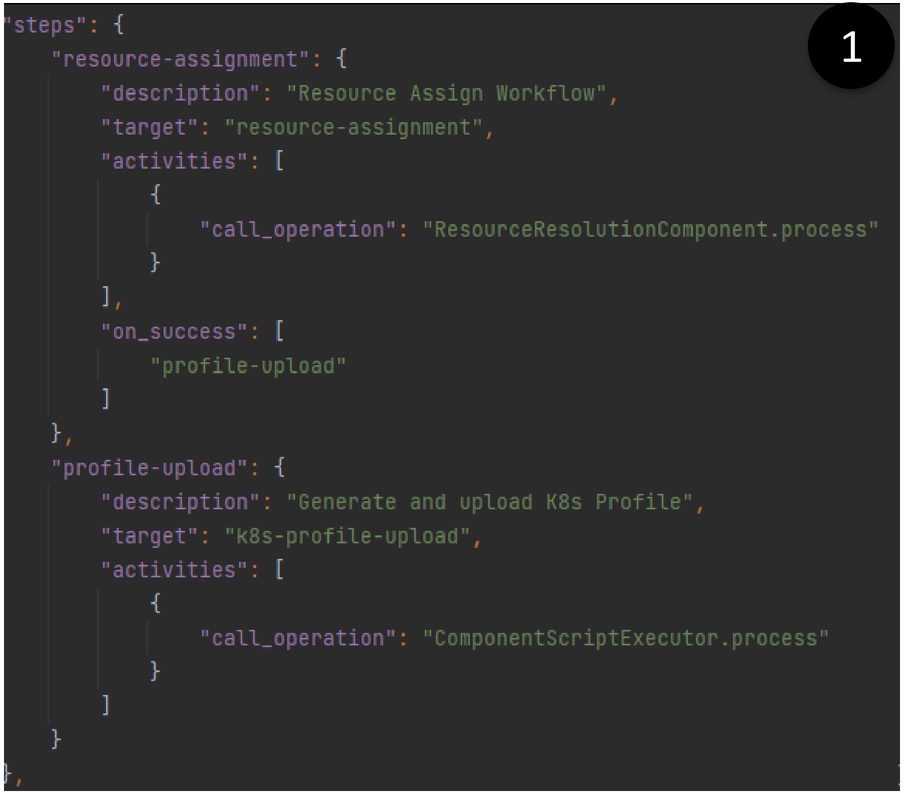
CORE Part

- Profile Upload

Config deploy
- Config Setup
- Status Verification 3
- PNF Reconfiguration 4

NRF Scaling 8
- Config Setup
- Config Upload
- Config Apply
- Status Verification

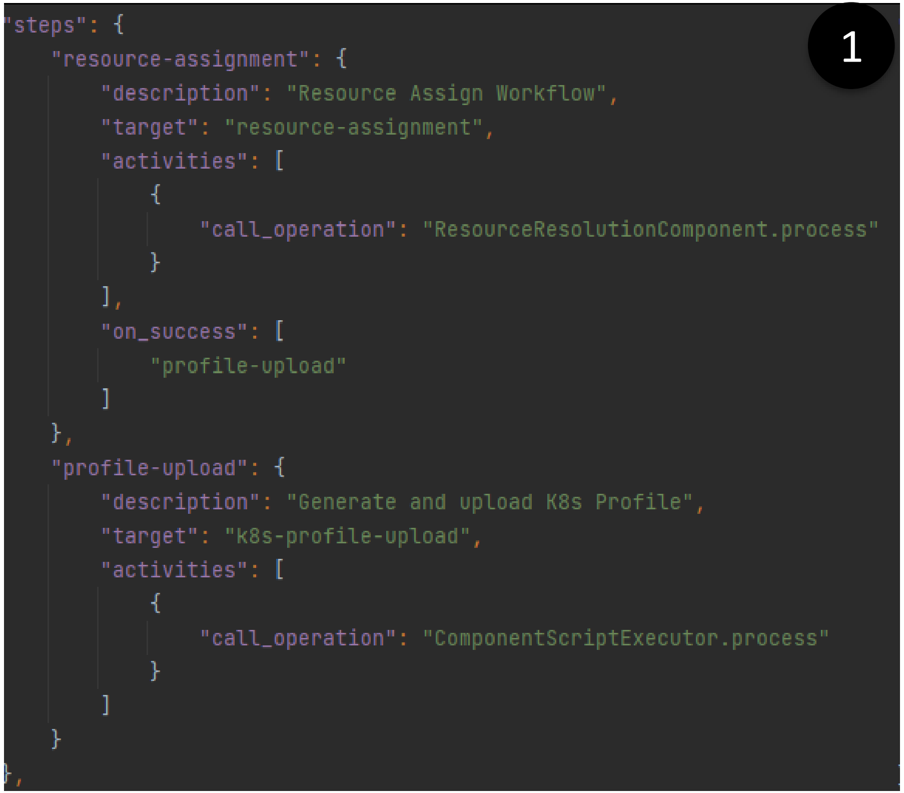
Resource Assignment (CNF/PNF)



```
"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": "ComponentScriptExecutor.process"
            }
        ]
    }
},
```

- ❑ **Resource Assignment**:
    - ❑ First of the ways to enrich Helm package
    - ❑ Resolves overrides for Helm instantiation
    - ❑ It is supplemented by profiling
    - ❑ We use it to gather inputs and prepare for profiling
    - ❑ Result is stored in MDSAL and can be easily used during Day2 operations
- ❑ **ResourceResolutionComponent** used

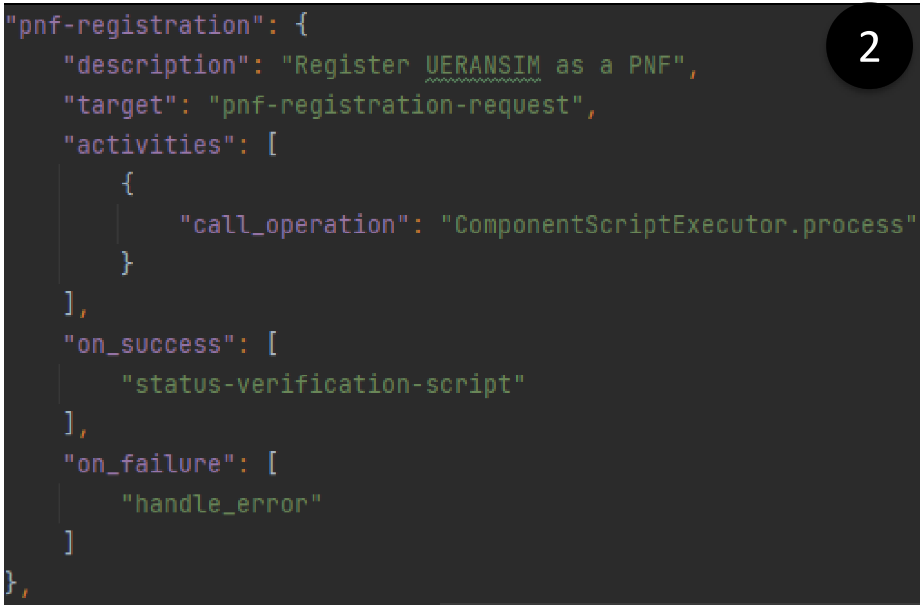
Resource Assignment (CNF/PNF)



```
"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": "ComponentScriptExecutor.process"
            }
        ]
    }
},
```

- **Profiling** mechanism allows to also parametrize complex overrides values
    - values.yaml file is taken from the profile
    - original helm chart **is not modified**
    - There are two types of profiles
        - **static** – predefined in CBA
        - **dynamic** – generated and templated during instantiation
    - CBA may have many profils with predefined overrides.
- **K8sProfileUploadComponent** is used

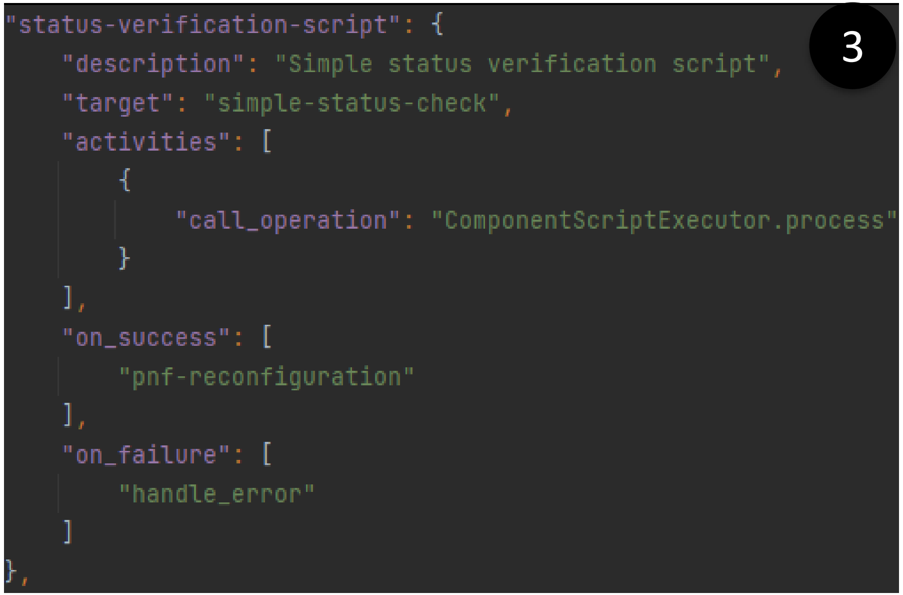
Config Deploy: PNF Registration



```
"pnf-registration": {
    "description": "Register UERANSIM as a PNF",
    "target": "pnf-registration-request",
    "activities": [
        {
            "call_operation": "ComponentScriptExecutor.process"
        }
    ],
    "on_success": [
        "status-verification-script"
    ],
    "on_failure": [
        "handle_error"
    ]
},
```

- ❑ Service model is composed of PNF and CNF
- ❑ PNF is simulated by UERANSIM solution
- ❑ In order to register PNF in ONAP **PNF Plug and Play** procedure is used
- ❑ This step sends PNF registration event to PRH component of DCAE
- ❑ CNF Core instantiation waits until PNF Registration finishes

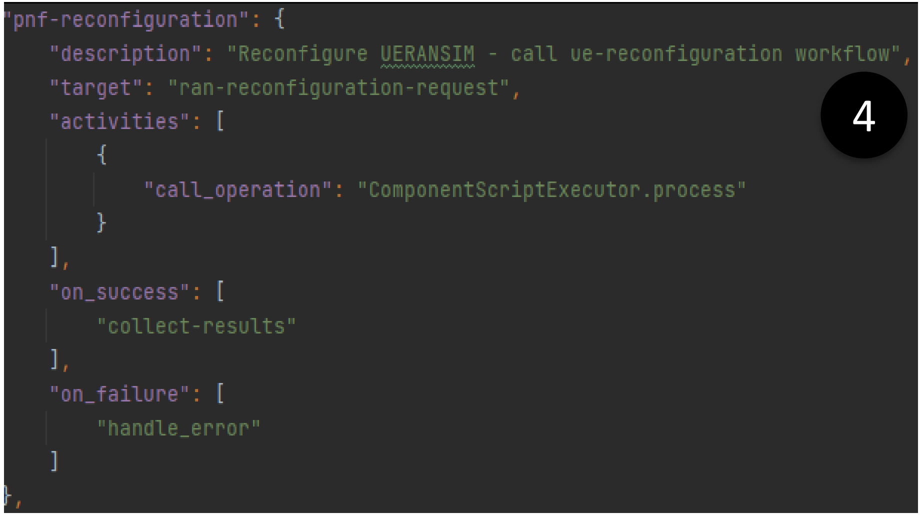
Config Deploy: Status Verification



```
"status-verification-script": {
    "description": "Simple status verification script",
    "target": "simple-status-check",
    "activities": [
        {
            "call_operation": "ComponentScriptExecutor.process"
        }
    ],
    "on_success": [
        "pnf-reconfiguration"
    ],
    "on_failure": [
        "handle_error"
    ]
},
```

- ❑ Procedure verifies if CNF is up and running
- ❑ All k8s resources created must have „Running” state to continue
- ❑ Script calls k8sPlugin Status API
- ❑ Instance status verification checks value of **ready** flag:
    - ❑ **False** means deployment in progress
    - ❑ **True** means deployment is finished
- ❑ **ComponentScriptExecutor** operation used

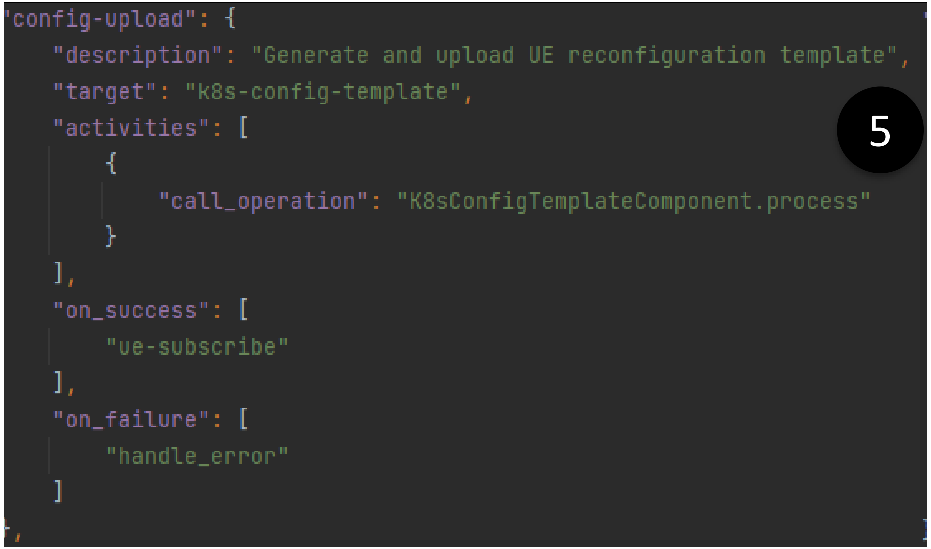
Config Deploy: PNF Reconfiguration



```
"pnf-reconfiguration": {
    "description": "Reconfigure UERANSIM - call ue-reconfiguration workflow",
    "target": "ran-reconfiguration-request",
    "activities": [
        {
            "call_operation": "ComponentScriptExecutor.process"
        }
    ],
    "on_success": [
        "collect-results"
    ],
    "on_failure": [
        "handle_error"
    ]
},
```

- ❑ Aim at configuration of PNF base on the configuration resolved from the CNF
- ❑ Request sent towards UERANSIM component contains parameters required during subscription, eg:
    - ❑ PLMN ID
    - ❑ UE ID
- ❑ **ComponentScriptExecutor** operation used

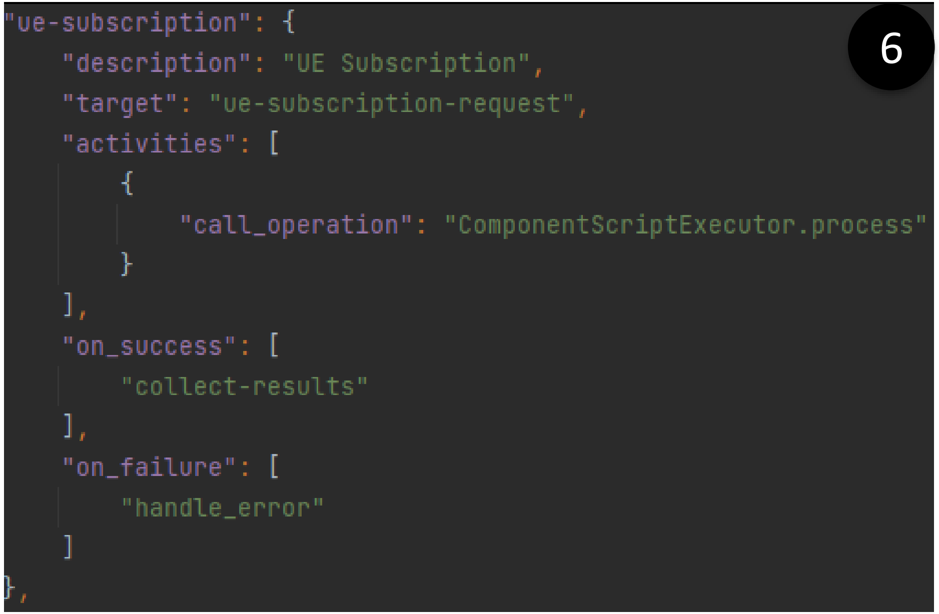
UE Reconfiguration: Config Upload



```
'config-upload": {
    "description": "Generate and upload UE reconfiguration template",
    "target": "k8s-config-template",
    "activities": [
        {
            "call_operation": "K8sConfigTemplateComponent.process"
        }
    ],
    "on_success": [
        "ue-subscribe"
    ],
    "on_failure": [
        "handle_error"
    ]
},
```

- ❑ **Configuration template** is generated and uploaded to k8sPlugin thanks to utilization of **K8sConfigTemplateComponent**
- ❑ Config Template is a Helm package
    - ❑ Part of the original Helm package
    - ❑ Does not have to be related with original Helm package
    - ❑ Create of Updates k8s resourcs
- ❑ Configuration template contains gnb chart of UERANSIM component with modified values

UE Reconfiguration: UE Subscribe



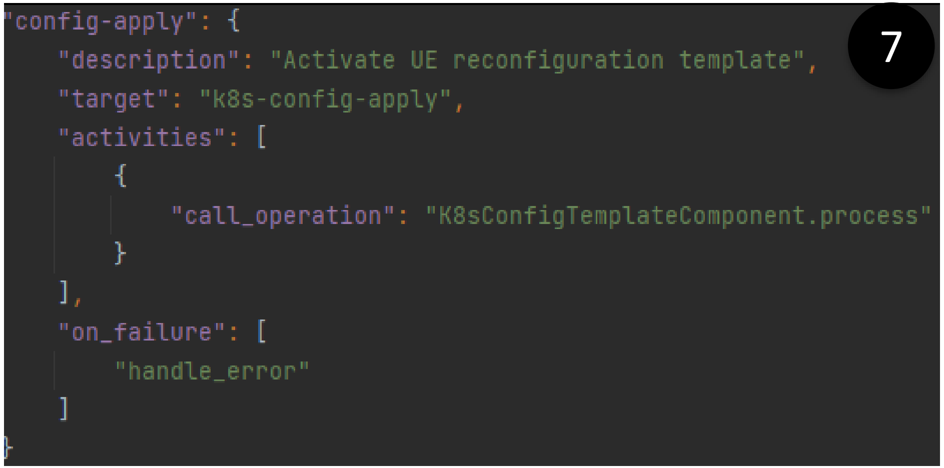
```
"ue-subscription": {
    "description": "UE Subscription",
    "target": "ue-subscription-request",
    "activities": [
        {
            "call_operation": "ComponentScriptExecutor.process"
        }
    ],
    "on_success": [
        "collect-results"
    ],
    "on_failure": [
        "handle_error"
    ]
},
```

❑ Script triggers execution of **ue-subscribe** workflow

❑ Subscription is required to allow for e2e connectivity from the UE to the internet

❑ **ComponentScriptExecutor** operation used

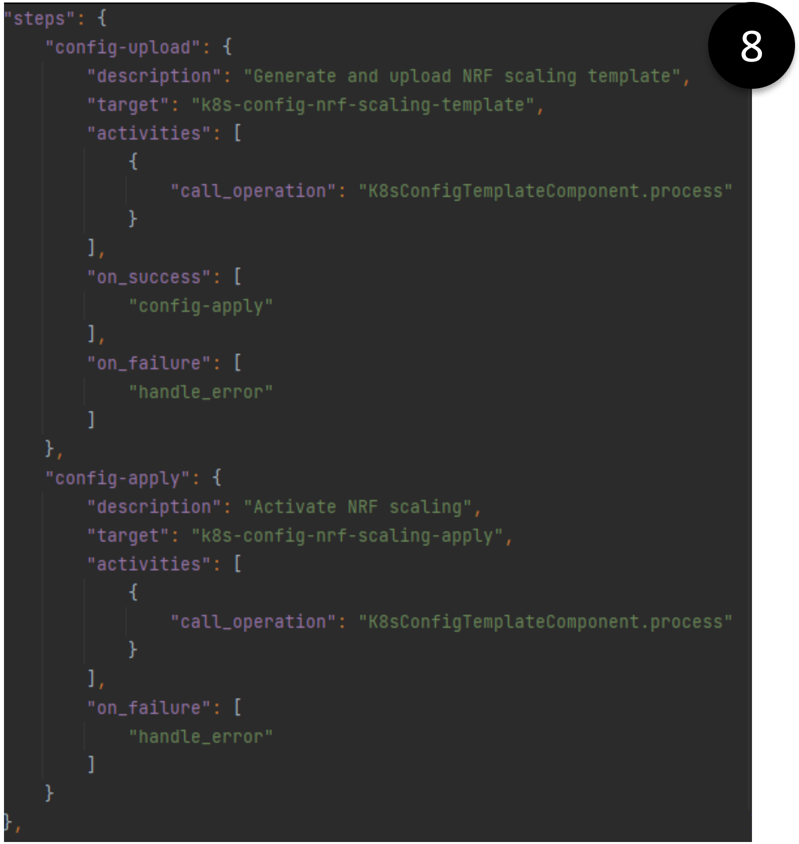
UE Reconfiguration: Config Apply



```
"config-apply": {
    "description": "Activate UE reconfiguration template",
    "target": "k8s-config-apply",
    "activities": [
        {
            "call_operation": "K8sConfigTemplateComponent.process"
        }
    ],
    "on_failure": [
        "handle_error"
    ]
}
```

- ❑ K8sPlugin instantiates the configuration uploaded during config-upload step
- ❑ As a result:
    - ❑ new gnb pod is created with modified parameters
    - ❑ The old instance is deleted
- ❑ **K8sConfigValuesComponent** component is utilized

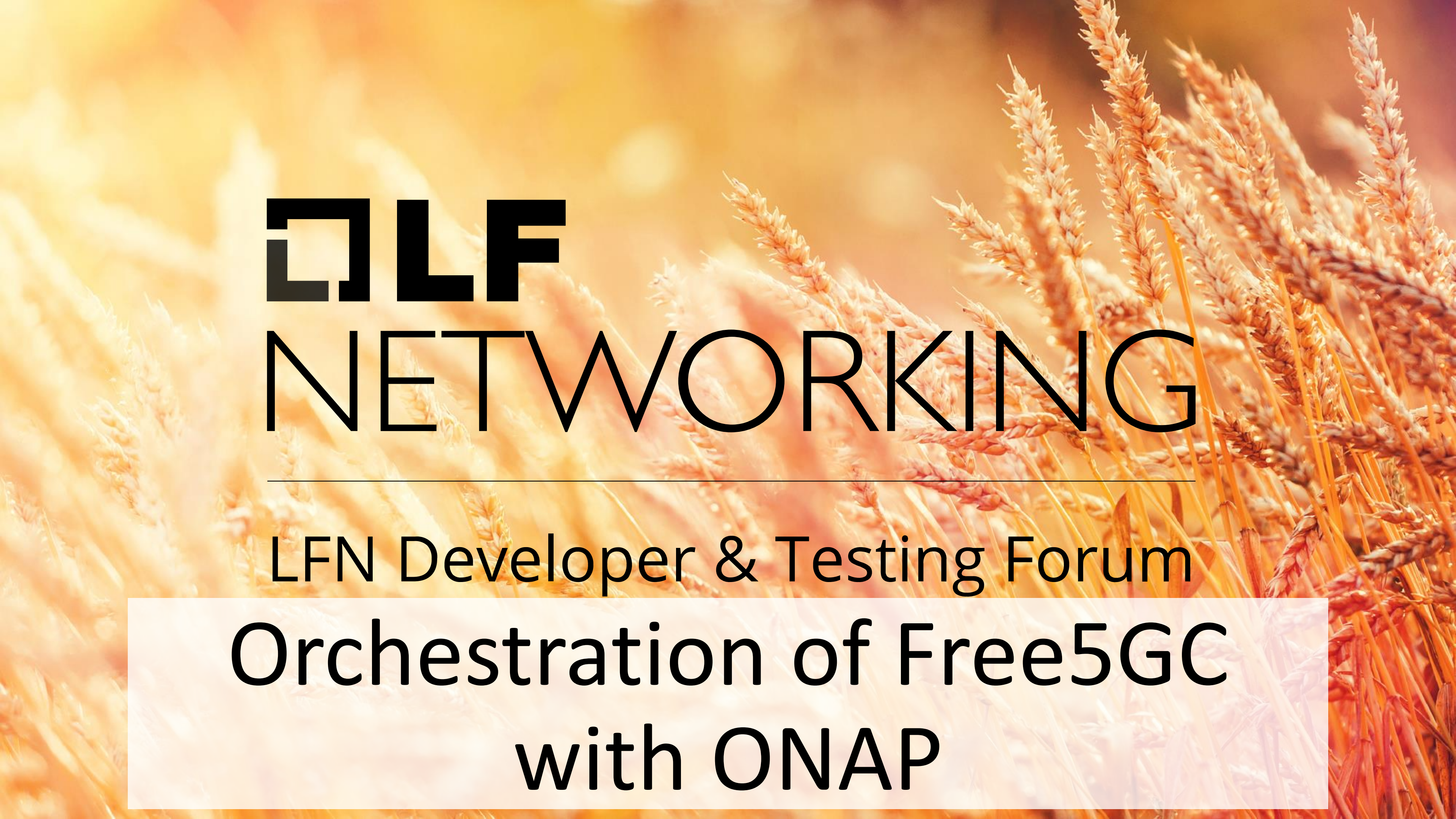
NRF Scaling



```
"steps": {
    "config-upload": {
        "description": "Generate and upload NRF scaling template",
        "target": "k8s-config-nrf-scaling-template",
        "activities": [
            {
                "call_operation": "K8sConfigTemplateComponent.process"
            }
        ],
        "on_success": [
            "config-apply"
        ],
        "on_failure": [
            "handle_error"
        ]
    },
    "config-apply": {
        "description": "Activate NRF scaling",
        "target": "k8s-config-nrf-scaling-apply",
        "activities": [
            {
                "call_operation": "K8sConfigTemplateComponent.process"
            }
        ],
        "on_failure": [
            "handle_error"
        ]
    }
},
```

8

- ❑ We combine config-upload and config-apply step that modifies the NRF deployment
- ❑ **K8sConfigTemplateComponent** is utilized for config-upload procedure
- ❑ **K8sConfigValuesComponent** is utilized for config-apply procedure
- ❑ Configuration template has NRF deployment template with modified number of NRF replicas
- ❑ During the configuration apply, new instance of NRF is created



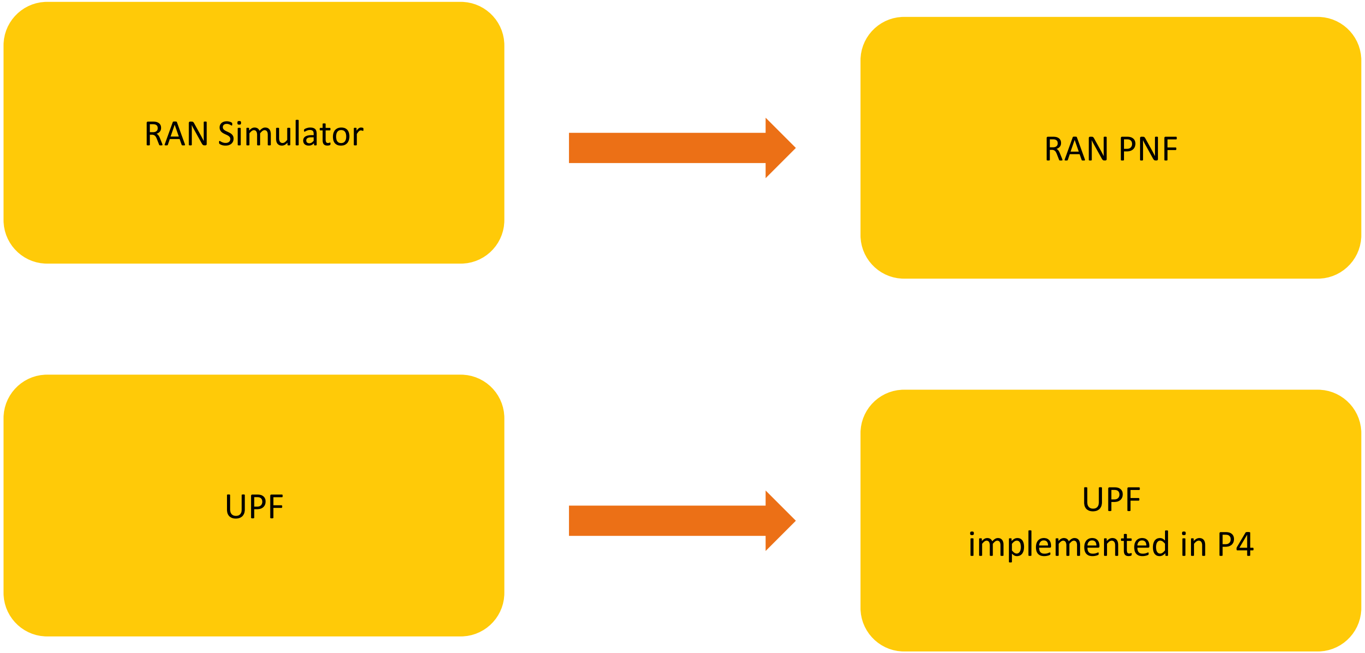
LF NETWORKING

LFN Developer & Testing Forum

Orchestration of Free5GC with ONAP

- VIDEO

Future work & enhancements



RAN Simulator → RAN PNF

UPF → UPF implemented in P4

Thank you

Merci

LF NETWORKING

LFN Developer & Testing Forum

धन्यवाद

Dziękujemy

LF NETWORKING

LFN Developer & Testing Forum