UNIÃO EUROPEIA
Fundos Europeus Estruturais e de Investimento

I&D Empresas

PORTUGAL 2020

Metodologia de Avaliação e Especificação de Casos de Uso para Validação

Nº. Entregável: E5.1 **Atividade:** A5

Data: 11/01/2022 **Versão:** v1.0

Nº. Referência Doc.: OREOS_E5.1 **Nível de Disseminação:** Público

Autor(es): Universidade de Coimbra

Palavras-chave: Orquestração, alocação de recursos, monitoria e sistemas analíticos.

Resumo:
O projeto OREOS visa uma orquestração e uma utilização de recursos ótima em cenários com serviços de baixa latência e de elevada resiliência. Este documento vista apresentar a definição da arquitetura da plataforma OREOS, incluindo a identificação dos blocos funcionais da arquitetura, dos serviços a suportar, dos módulos de hardware e software a usar e das respetivas interfaces.

Autoria e controlo de versões

Autores do Documento

Organização	Autores
Universidade de Coimbra	Bruno Sousa; José Luís; Marco Silva; David Abreu; Marília Curado
Altran Portugal SA	Marco Araújo, João Silva, Olusola Agbebiyi, Bruno Mendes, Mojgan Barahman, André Martins
Instituto Pedro Nunes	Bruno Faria; Karima Velasquez

Histórico do 2ocument

Edição	Data	Notas
0.1	2021/09/15	Definição da estrutura do documento
0.2	2021/12/10	Integração de conteúdos
0.3	2021/12/22	Versão integrada
0.4	2021/12/30	Versão revista
1.0	2022/01/11	Versão final

Este projeto é financiado pelo programa Portugal 2020, ao abrigo do sistema de incentivos à investigação e desenvolvimento tecnológico, contrato número POCI-01-0247-FEDER-049029.

Sumário Executivo

O projeto OREOS visa conceber e implementar uma plataforma de orquestração fim-a-fim para provisionamento e gestão de serviços críticos, incluindo comunicações veiculares, redes de distribuição elétrica, e comunicações de emergência de segurança pública, que têm como base as tecnologias de comunicação móvel de quinta geração (5G), em particular, o seu suporte para serviços do tipo *Ultra Reliable and Low Latency Communications* (URLLC).

Os serviços críticos visados pelo projeto OREOS são caracterizados por requisitos de baixa latência e elevada fiabilidade, que beneficiam da combinação otimizada de modelos arquiteturais distintos, tais como computação *Fog*, *Edge* e Cloud. É neste contexto que este documento apresenta a metodologia a ser seguida para avaliar a arquitetura do projeto nos casos de uso de Cidades Inteligentes – *Smart City* e condução autónoma - *Autonomous Driving*. Os casos de uso são detalhados neste documento, em conjunto com a descrição da forma como a arquitetura OREOS é utilizada. O documento apresenta também a instanciação da infraestrutura e software necessários para o desenvolvimento da arquitetura OREOS.

Abstract

OREOS aims to design and implement an end-to-end orchestration platform for provisioning and managing critical services (such as vehicle communications, electrical distribution networks or emergency communications for public safety). The evolution in the 5G networks, in particular the support of services based on Ultra-Reliable and Low Latency Communications (URLLC), enables critical services for vehicle communications and smart cities.

The critical services targeted by the OREOS project are characterized by low latency and high-reliability requirements which benefit from the optimized combination of different architectural models, such as Fog, Edge and Cloud computing. In this context, this document presents the evaluation methodologies to assess the support of the architecture towards two main use cases: Smart City and Autonomous Driving. Such use cases are detailed in this document, as well as the way the OREOS architecture is exercised in each one. The document also provides details for the instantiation of the infrastructure and software that are required to deploy OREOS.

Table of Contents

List of figures

List of tables

Acronyms

3GPP	*3rd Generation Partnership Project*
5G-EIR	*5G Equipment Identity Register*
5GC	*5G Core*
AAL	*Accelerator Abstraction Layer*
AAI	*Active and Available Inventory*
ACI	*Application Centric Infrastructure*
AF	*Application Function*
AI	*Artificial Intelligence*
AMF	*Access and Mobility Management Function*
APPC	*Application Controller*
AUSF	*Authentication Server Function*
BBU	*Baseband Unit*
BCH	*Broadcast Channel*
BH	*Back-Haul*
BS	*Base Station*
BSS	*Business Support System*
CBA	*Controller Blueprint Archive*
CBCF	*Cell Broadcast Centre Function*
CBS	*Configuration Binding Service*
CDAP	*Cask Data Application Platform*
CDS	*Controller Deseign Studio*
CHF	*Charging Function*
CLAMP	*Closed Loop Automation Management Platform*
CMSO	*Change Management Schedule Optimizer*
CN	*Core Network*
CNF	*Cloud Network Function*
CNI	*Container Network Interface*
CPD	*Connection Point Descriptor*
CP	*Control Plane*
CPS	*Configuration & Persistency Service*
CRD	*Custoem Resource Definition*

CSI	*Channel Station Information*
CSI-RS	*Channel Sate Information Router Solitication*
CSMF	*Communication Service Management Function*
CSP	*Communications Service Provider*
CU	*Central Unit*
CSMF	*Communication Service Management Function*
DB	*Data Base*
DCAE	*Data Collection, Analytics, and Events*
DCI	*Dowlink Control Information*
DEK	*Open Developer Experience Kits*
DFC	*DataFile Collector*
DG	*Directed Graph*
DN	*Data Network*
DNS	*Domain Name Service*
DPDK	*Data Plane Development Kit*
DU	*Distributed Unit*
DYI	*Do It Yourself*
E2AP	*Protocol by which nodes communicate over E2 interface*
E2E	*End-to-End*
EIR	*Equipment Identity Register*
EMS	*Element Management System*
eMBB	*Enhanced Mobile Broadband*
eMBMS	*Evolved Multimedia Broadcast Multicast Service*
eNB	*Evolved Node B*
EPC	*Evolved Packet Core*
ESR	*External System Register*
ETSI	*European Telecommunications Standards Institute*
FAPI	*fFnctional Application Platform Interface*
FGPS	*Fine-Grained Placement Service*
FH	*Front-Haul*
gNB	*Next Generation NodeBs*
GPRS	*General Packet Radio Service*
G-VNFM	*Generic VNF Manager*

GTP	*GPRS Tunneling*
HAS	*Homing and Allocation Service*
HMTC	*High-Performance Matchine-Type Communications*
HOT	*Heat Orchestration Template*
HTTP	*Hypertext Transfer Protocol*
HV-VES	*High Volume Virtual Network Function Event Streaming*
IP	*Internet Protocol*
ISG	*Industry Specification Group*
LCM	*Life-Cycle Management*
MANO	*Management and Orchestration*
MCC	*Mobile Country Code*
MD-SAL	*OpenDayLight for Service Abstraction*
MEAO	*Mobile Edge Application Orchestrator*
MEC	*Muti-access Edge Computing*
MEP	*MEC platform*
MEPM	*Mobile Edge Platform Manager*
MH	*Mid-Haul*
MIoT	*Massive Internet of Things*
ML	*Machine Learning*
MME	*Mobility Management Entity*
MNC	*Mobile Network Code*
MSO	*Master Service Orchestrator*
MS	*Micro Service*
MSB	*Micro Service Bus*
N3IWF	*Non-3GPP InterWorking Function*
NaaS	*Netwokring-as-a-Service*
NAS	*Non-Access Stratum*
NBI	*Northbound Interface*
NB-IoT	*NarrowBand Internet of Things*
NDAI	*Newotk Data Access Identifier*
NEF	*Network Exposure Function*
NF	*Network Function*
NFVI	*NFV infrastructure*

NFVO	*Network Function Virtualization Orchestrator*
NG	*Next Generation*
NGAP	*Next Generation Application Protocol*
NR	*New Radio*
NRF	*Network Repository Function*
NSA	*Non-StandAlone mode*
NSI	*Network Slice Instance*
NSMF	*Network Slice Management Function*
NSSAI	*Network Slice Selection Assistance Information*
NSSI	*Network Slice Subnet Instance*
NSSF	*Network Slice Selection Function*
NSSMF	*Network Slice Subnet Management Function*
NSSAAF	*Network Slice Specific Authentication and Authorization Function*
NFV	*Network Virtual Function*
NWDAF	*Network Data Analytics Function*
OAI	*Open Air Interface*
OAM	*Operation, Administration and Management*
OBU	*On Board Unit*
OCS	*Online Charning System*
ODL	*Open Day Light*
OFCS	*Offline Charging System*
OLT	*Optical Line Terminal*
OOF	*Open Network Automation Platform Optimization Framework*
OOM	*Open Network Automation Platform Operations Manager*
OPNFV	*Open Platform for Network Function Virtualisation*
ONAP	*Open Network Automation Platform*
OSS	*Operations Support System*
OSDF	*Optimization Service Design Framework*
PCF	*Policy Control Function*
PCI	*Physical Cell ID*
PDCP	*Packet Data Convergance Protocol*
PDP	*Policy Decision Point*
PDSCH	*Packet Data Scheduling Channel*

PDU	*Packet Data Unit*
PLMN	*Public Land Mobile Network*
PM	*Performance Measurement*
PNF	*Physical Network Function*
PRB	*Physical Resource Blocks*
RAN	*Radio Access Network*
RIC	*RAN Intelligent Controller*
RNIS	*Radio Network Information Service*
RPC	*Remote Processing Call*
RRC	*Radio Resource Control*
RRM	*Radio Resource Management*
RRU	*Remote Radio Unit*
RSU	*Road Side Unit*
RT	*Real Time*
RU	*Radio Unit*
SBA	*Service Based Architecture*
SCTP	*Stream Control Transport Protocol*
SD	*Slice Differentiator*
SD-RAN	*Software-Defined Radio Access Networking*
SDC	*Service Design and Creation*
SDNC	*Software Defined Network Controller*
SDK	*Software Development Kit*
SDR	*Software-Defined Radio*
SGW	*Serving gateway*
SIB1	*System Information Block*
SLA	*service level agreement*
SLI	*Service Logic Interpreter*
SM	*Slice-service Model*
SMF	*Session Management Function*
SO	*Service Orchestrator*
SON	*Self-Organized Networks*
SQL	*Structured Query Language*
SRVCC	*Single Radio Voice Call Continuity*

SST	*Slice Service Type*
TAC	*Tracking Area Code*
TCA	*Threshold Crossing Analytics*
TCP	*Transmission Control Protocol*
TN	*Transport Network*
TOSCA	*Topology and Orchestration Specification for Cloud Applications*
UAV	*Unmanned Aerial Vehicle*
UCMF	*UE Capability Management Function*
UDM	*Unified Data Management*
UDR	*Unified Data Repository*
UE	*User Equipment*
ULCL	*UP Uplink Classifier*
UP	*User Plane*
UPF	*User Plane Function*
URLLC	*Ultra-Reliable and Low Latency Communications*
US	*User Story*
UUI	*ONAP Use Case User Interface*
VDU	*Virtualisation Deployment Unit*
VES	*Virtual Network Function Event Streaming*
VFC	*Virtual Function Component*
VF-C	*Virtual function controller*
VIM	*Virtualized Infrastructure Manager*
VLD	*Virtual Link Descriptor*
VM	*Virtual Machine*
VNF	*Virtual Network Function*
VNFM	*Virtual Network Function Management*
VNFC	*VNF Component*
VNF-D	*VNF Descriptor*
VNFM	*VNF Manager*
VPC	*Virtual Private Cloud*
KPI	*Key Performance Indicator*
YAML	*Yet Another Markup Language*

1. Introduction

The main objective of this deliverable is to provide a complete description and comparison of components and tools involved in the OREOS platform. Such tools have been evaluated at distinct levels, namely, radio access, 5G network core, multi-access edge computing (MEC), and cloud infrastructure, considering that some of them are brand new in the open-source market. OREOS aims to design and implement an end-to-end orchestration platform for provisioning and managing critical services (such as vehicle communications, electrical distribution networks and emergency communications for public safety). The evolution in 5G networks, in particular the support of services based on Ultra-Reliable and Low Latency Communications (URLLC), enables critical services for vehicle communications and smart cities.

The Radio Access Network (RAN) can rely on diverse open-source solutions. For instance, the OpenAirInterface (OAI) (OpenAirInterface, 2021) is an open-source software-based ecosystem that targets the implementation of mobile networks prototypes. The UERANSIM is an open-source state-of-the-art ecosystem (UERANSIM, 2021) that implements the new 5G RAN covering North Bound and 5G User Equipment (UE). The difference between these solutions relies mainly on the coverage provided, in the functionalities included and planned in their roadmap towards compliance with 3rd Generation Partnership Project (3GPP) standards, and the compliance with current trends regarding the virtualization of RAN being promoted by the ORAN-Alliance. Within such criteria, this document identifies OAI as the best solution for the radio access component of the OREOS platform.

The 5G Core for the OREOS platform can rely as well on open-source solutions such as the Open Air Interface for the Core project (OAI-CN) (E. Zhukov, 2019), Open5GS (Niknam, 2021), and Free5GC. OAI is an open software that gathers a community of developers from around the world working together to build wireless cellular Radio Access Network (RAN) and, more recently, Core Network (CN) technologies. It implements the 3GPP technology for 5G Core Network. Open5GS is a C-language Open-Source project of 5GC and *Evolved Packet Core* that can be used to configure NR/LTE network and includes the control and the user planes. Free5GC is an open-source project (Free5G, 2021) for 5G mobile core networks where it is possible to implement the 5GC defined in 3GPP for release 15. The comparison performed in this document highlights OAI-CN as the feasible solution, mainly due to the planned support for Network Data Analytics Function (NWADF), which is required in the OREOS platform. FlexRIC is a Flexible and programmable RAN Intelligent Controller for Software-Defined Radio Access Networking (SD-RAN), it has interfaces with the OAI radio stack over the O-RAN-defined E2-interface to monitor and control the RAN in real-time. It supports real-time monitoring and control for 4G and 5G RAN, being a feasible solution to be used with OAI-CN.

Multi-access Edge Computing (MEC) offers cloud computing capabilities and an IT service environment at the edge of the network. MEC provides a new ecosystem and value chain where operators can open their RAN edge to authorize easy access and flexibility to third parties which enables the deployment of services or applications towards mobile subscribers. The MEC initiative is an Industry Specification Group (ISG) within the European Telecommunications Standards Institute (ETSI). The purpose of ISG is to create a standardized, open environment which will allow the efficient and seamless integration of applications from vendors, service providers, and third parties across multi-vendor MEC platforms (Hairuman, 2019; ETSI, ETSI NFV&MEC Plugtests report, 2020). The Open Network Edge Services Software (OpenNESS), now called Intel Smart Edge Open, is an open-source edge computing toolkit that allows building platforms optimized for the edge, capable of hosting several services and functionalities such as 5G Core or 5G RAN (Intel, 2021).

Cloud components can rely on diverse solutions, based on the ETSI *Network Function Virtualization* (NFV) model with OpenStack or following a container approach for Cloud Network Function (CNF) with Kubernetes. ONAP supports both modes, but decisions must be taken from the network design phase, considering Controller Design Studio (CDS) and the Configuration & Persistency Service (CPS) component that is designed to serve as a data repository for run-time configuration and operational data that needs to be persistent.

The OREOS platform leverages on the network slicing features supported by 5G. The vertical industries are diverse, and their requirements are defined by the service characteristics of the vertical segment. For example, for an enhanced Mobile BroadBand (eMBB) service it makes sense for the UPF to be placed in a regional Data Center, in order to maximize the number of subscribers attached to it; for an Ultra-Reliable Low Latency Communications (URLLC) service, the UPF must be located at the nearest point of the service subscriber in order to minimize latency while simultaneously maximizing reliability (but at the cost of having only a few subscribers attached to it). This document details how network slicing can be configured to meet the requirements of the Smart City and Autonomous Driving use cases.

This document provides an exhaustive step-by-step explanation of two major use cases to validate the OREOS platform: Use Case 1 - Smart City, Use Case 2 - Autonomous Driving. Deliverable E3.1 (OREOS, Deliverable E3.1, 2021) documented several use cases, such as the following: Pedestrian within safety vehicle mobility; End-to-end network slicing; 3GPP & O-RAN alignment; 5G SONs; and Intent Based Networking. These have been merged into the two major use cases. The Smart City use case incorporates three user stories that include pedestrian safety, air quality and crime prevention, considering recent trends towards sustainable cities. The

functional and non-functional requirements of OREOS use cases are described. In addition, sequence diagrams of the workflow steps of each use case are presented which explain the interactions among the various components of the OREOS platform.

This document entails information regarding the validation of various tools and components in terms of deployment and provisioning to enable the functionalities required by the OREOS platform. The validation methodology for the OREOS platform identifies an initial set of KPIs for the use cases and procedures to successfully evaluate the steps in the defined workflows.

Last but not least, this document includes a step-by-step tutorial which will be crucial for future deployments of the OREOS platform and associated components. The tutorial contains information regarding orchestration, infrastructure manager and slice management components. The document also discloses the planned timeframe of the project regarding the next steps in terms of enabling the OREOS prototypes and evaluation activities.

2. Software deployment

This section provides the technical specification for the software deployment of the OREOS platform. To perform the deployment validation, we must establish the following:

- The required Network Functions (NF) and their interfaces.
- A validation method to deploy the network will be presented for the following platform components:
    - Radio Access Network (RAN)
    - Core Network (CN), also known as 5G Core (5GC)
    - Mobile Edge Computing (MEC)
    - Management and Orchestration (MANO)
- A series of open-source tools have been studied in order to select the most appropriated ones (i.e., the ones that better fulfil the requirements) to deploy in the platform (Melodia, 2020). The alternatives were::

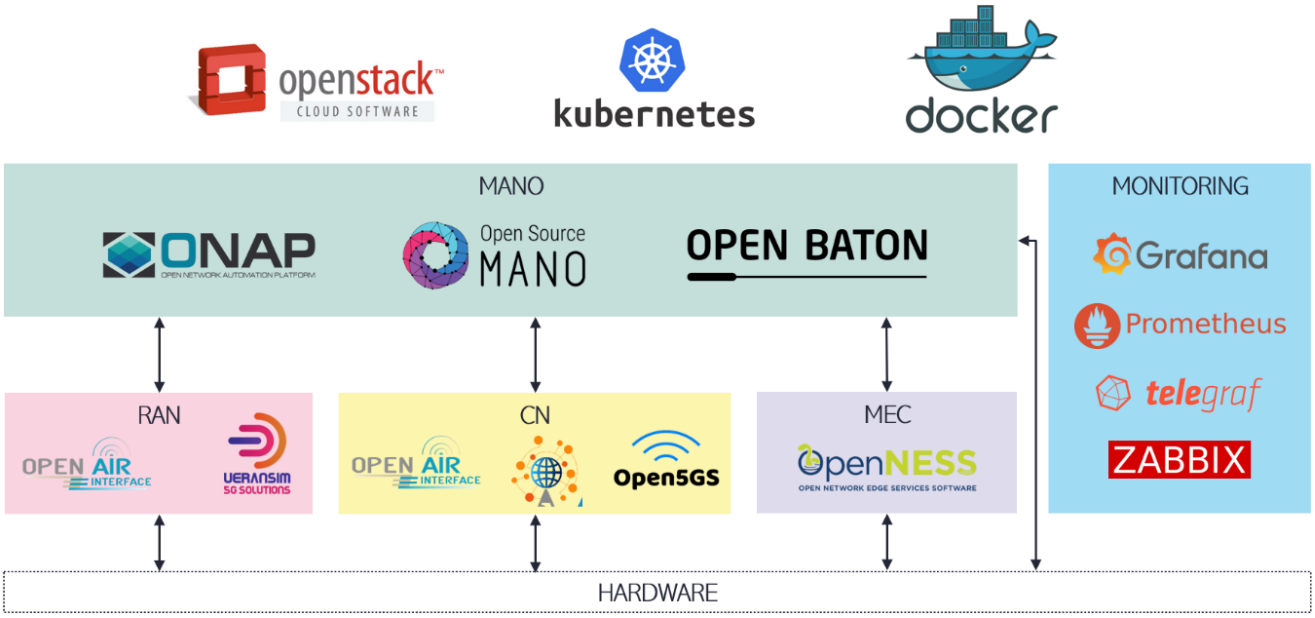

Figure 1: High-level architecture of the OREOS platform

Note that the comparison between MANO frameworks and monitoring solutions was already performed in the previous deliverables of the OREOS project (OREOS, Deliverable 2.1, 2021).

2.1 Open-source solutions

The following subsections will describe the most popular open-source solutions available for RAN, MEC and 5GC, including a comparison among them.

2.1.1 Radio Access Network

The Radio Access Network (RAN) is one of the four major domains within a mobile telecommunications network. It is composed of a set of Base Stations (BS), which are connected to the CN through the transport network. The RAN's main task is to provide connectivity to the devices that integrate its network through wireless radio links. Traditionally, it is composed of three major components: 1- the antennas, responsible for signal transmission and reception; 2- the Remote Radio Unit (RRU) which is mainly responsible for the analog signal processing operations on both transmission and receiving chains, and for the signal conversions (from digital to analog and vice versa); 3- the Baseband Unit (BBU) which is composed by a collection of tailor-made electronics designed to enable the high computational power required to, efficiently and securely, make use of the available (licensed) radio spectrum. In other words, it is within the BBU that the system capacity is determined. These components are illustrated in Figure 2:

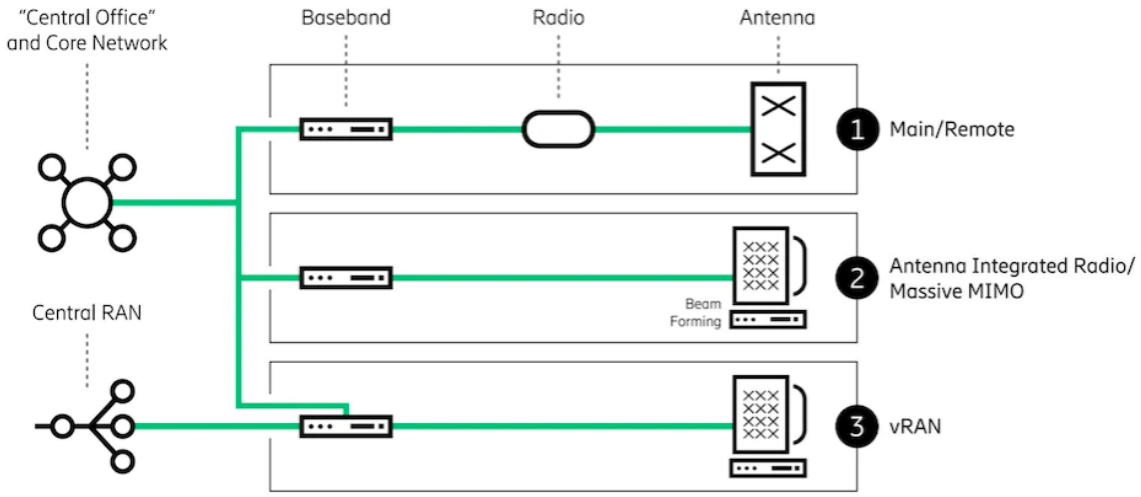

Figure 2: RAN main components

2.1.1.1 OAI-RAN

The OpenAirInterface (OAI) (OpenAirInterface, 2021) is an open-source software-based ecosystem that targets the implementation of mobile networks prototypes. It became popular by following the 3GPP technologies roadmap, by implementing at first, LTE (Rel 8), followed by LTE-A (Rel 10/11/12), LTE-A-PRO (Rel 13/14), and going on to NR-5G (Rel 15/16/…). OAI spans the full protocol stack of the 3GPP standard, from the RAN where evolved NodeB (eNB), next-generation NodeB (gNB), User Equipment (UE), and New Radio (NR) UE implementations are present, to the CN implementation which includes the Evolved Packet Core (EPC) and the 5GC (Florian Kaltenberger a).

With reference to OAI 5G RAN, there are two main scopes to consider within this domain: the Non-StandAlone mode (NSA) and the StandAlone mode (SA). At the present date, OAI has its 5G NSA solution already in the final stage of development. This type of architecture consists of an LTE eNB playing the role of master and the new 5G gNB the role of slave. In this option, the 5G cell will be connected to a 4G evolved packet network under the control of a 4G cell. On the 4G cell, all the Control Plane (CP) traffic is carried out and the S1 Mobility Manage Entity (MME) interface terminates. On the other hand, the S1-U interface connects the nodeB (NB) to the Serving Gateway (SGW), which can either terminate in eNB or gNB. This way, UEs will first connect to a 4G network, before connecting to a 5G cell through Radio Resource Control (RRC). This is a network model that is meant to allow a polished and smooth transition from 4G to 5G. Figure 3 shows a representative block diagram of the architecture described above.

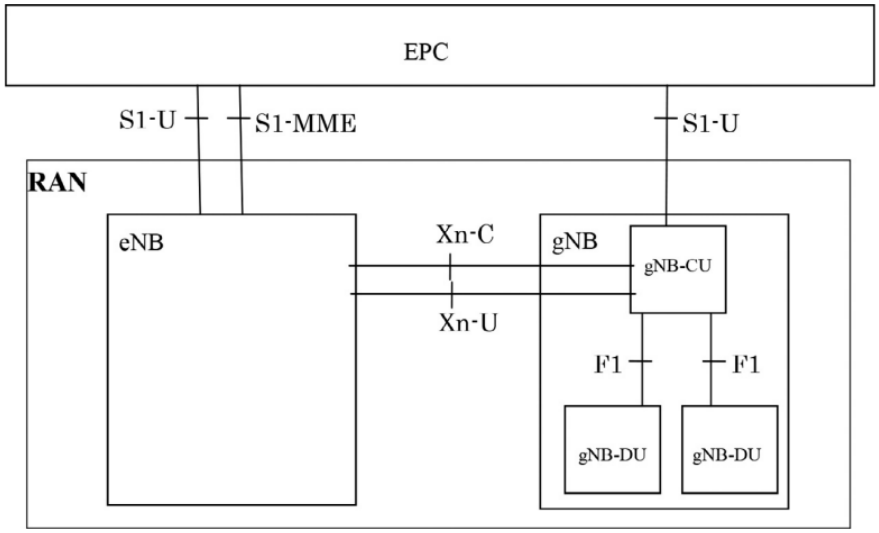

Figure 3: NSA network model for NR-RAN

The SA architecture (fully 5G), which is the most recent addition to the OAI ecosystem, is defined by 3GPP. It expects a direct and exclusive connection between 5G gNB and the 5GC network, which has a cloud-centered architecture. In the SA scenario, the Control Plane (CP) and the User Plane (UP) are both used on 5G NR and evolved LTE cells. Within the latest version of OAI, it already integrates the 5GC Network Functions (NF) of Access Management Function (AMF) and User Plane Function (UPF). The general SA scenario architecture is presented in Figure 4. In terms of supported platforms, OAI 5G NSA, works on the following RRUs (already tested): Software-Defined Radio (SDR) platforms, USRP B2x0 and B3x0 series (for both eNB and gNB), the Lime-SDR, the radio unit Benetel RRU (O-RAN 7.2), and lastly, it also supports AW2S remote radio head units (only for the LTE band 38 for the present moment). Besides, it is also possible to deploy, not only a complete LTE

network, from CN to UE, but also it is possible to deploy an OAI 5G CN, OAI gNB and OAI UE in a Radio Frequency (RF) simulator mode (SA architecture).

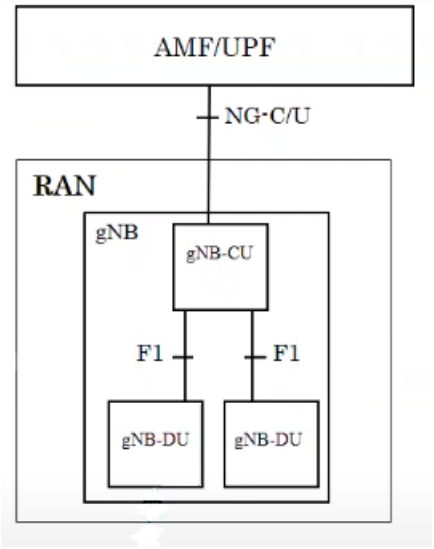

Figure 4: SA network model for NR-RAN

With regards to the eNB, at the present date, OAI implements the version of the RRC in the 3GPP Rel 15. enabling, this way, full coverage support directed towards the required signalling to and from the 5G UE. Whereas the gNB integrates three main modules: the Central Unit (CU), responsible for the RRC and Packet Data Convergence Protocol (PDCP) layers, the Distributed Unit (DU), where PHY, MAC, and Radio Link Control (RLC) layers are implemented, and the Radio Unit (RU), Figure 5 shows the gNB functional architecture.

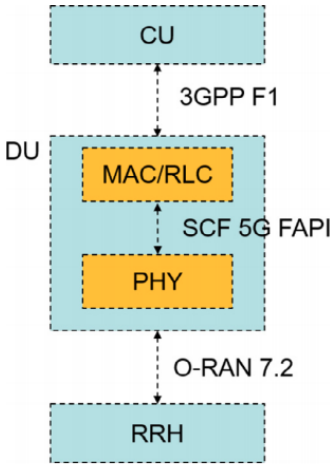

Figure 5: gNB functional architecture

OAI's ecosystem also follows the 3GPP standard regarding the interfaces between CU and DU, by implementing the F1 interface, while between the DU and RU the interface implemented is the O-RAN 7.2. Within the DU there is also an interface implemented between the MAC and PHY layer that follows the one specified by Small Cell Forum (EURECOM, 2021). This is illustrated in Figure 6.

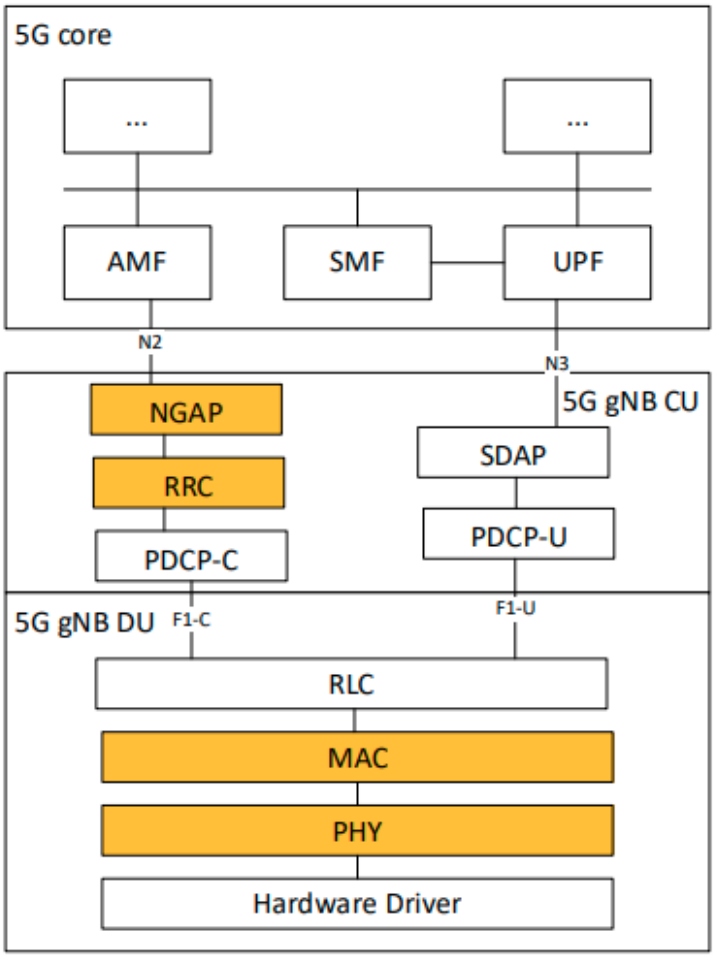

Figure 6: 5G SA network model

With respect to the PHY layer and gNB, OAI supports the following feature set:

- Static TDD;
- FDD;
- Normal Cycle Prefix;
- Numerology 1 (30 kHz subcarrier spacing), in FR1;
- Numerology 3 (120 kHz subcarrier spacing), in FR2;
- Bandwidths: 10, 20, 40, 80, 100MHz (273 Physical Resource Blocks);

- Slot format: 14 OFDM symbols in UL or DL;
- Generation of NR-PSS and NR-SSS;
- NR-PBCH;
- NR-PDCCH;
- NR-PDSCH;
- NR-CSI;
- NR-PUSCH;
- NR-PUCCH;
- NR-PRACH;
- Highly efficient 3GPP compliant LDPC encoder and decoder;
- Highly efficient 3GPP compliant polar encoder and decoder;
- Enconder and decoder for short blocks.

Moreover, moving up to the gNB's MAC layer, OAI integrates the following feature set:

- MAC - PHY configuration (unidirectional) using NR FAPI P5 interface;
- MAC - PHY data interface (bidirectional) using FAPI P7 interface for Broadcast Channel (BCH) PDU, Downlink Control Information (DCI), PDU, PDSCH PDU;
- Scheduler procedures for System Information Block 1 (SIB1);
- Scheduler procedures for RA;
- Scheduler procedures for CSI-RS;
- MAC downlink scheduler;
- MAC header generation (including timing advance);
- ACK / NACK handling and HARQ procedures for downlink;
- MAC uplink scheduler;
- MAC procedures to handle CSI measurement report;
- MAC scheduling of SR reception.

The RLC layer follows the guidelines present in the 3GPP 38.322 Rel.16 document with regards to send/receive operations, therefore makes sense that both PDCP and RRC also follow the Rel. 16 guidelines, which they do (38.322 and 38.323, respectively).

Currently, OAI presents some limitations. The main one concerns the achieved throughput, which presents poor results in two different scenarios, when higher complexity modulation schemes are utilized (starting on 16-QAM and beyond). Another concern refers to not scheduling all the possible resource slots within the resource block. Moreover, the OAI scheduler only supports one user when in 5G SA mode, although, this is a feature that

is being upgraded for multi-user scenarios. Lastly, the UE when connected to OAI ecosystem is not stable enough considering the system requirements (EURECOM, 2021). Figure 7 illustrates the features availability in the roadmap (source (EURECOM, 2021)).

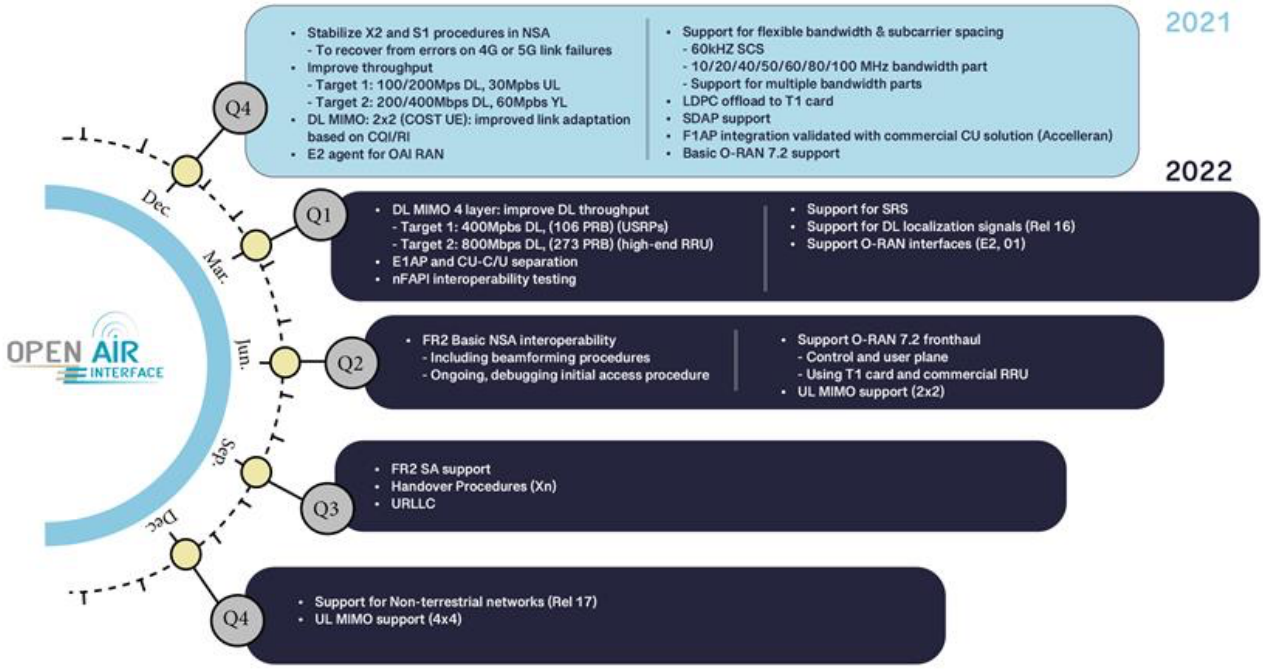

Figure 7: OAI-RAN roadmap

2.1.1.2 UERANSIM

UERANSIM is an open-source state-of-the-art ecosystem (UERANSIM, 2021) that implements the new 5G RAN covering both gNB and 5G UE. Accordingly, this project focuses on testing and studying 5G systems and it is in continuous development following the guidelines of 3GPP Rel 15.6.0. Unlike OAI, the UERANSIM project only targets the SA architecture.

From a feature set perspective, there are three main interfaces in UERANSIM: the control interface (between RAN and AMF), the user interface (between RAN and UPF), and the radio interface (between UE and RAN). The Control Plane (CP) is composed of two interfaces, the Non-Access Stratum (NAS) responsible for UE controlling functions and the NG Application Protocol (NGAP), responsible for providing control plane signalling between NG RAN and the AMF.

Within the NAS domain, the following features are already implemented in UERANSIM:

- Primary Authentication and Key Agreement;
- Security Mode Control;
- Identification;

- Generic UE Configuration Update;
- Initial and Periodic Registration;
- UE and Network initiated De-registration;
- UE initiated PDU session establishment;
- UE and Network initiated PDU session release;
- Service Request;
- Paging.

From the NGAP perspective, the following features are implemented:

- PDU Session Resource Setup;
- PDU Session Resource Release;
- Initial Context Setup;
- UE Context Release (NG-RAN node initiated and AMF initiated);
- UE Context Modification;
- Initial UE Message;
- Paging;
- Downlink NAS Transport;
- Uplink NAS Transport;
- NAS Non-Delivery Indication;
- Reroute NAS Request;
- NG Setup;
- Error Indication.

With regards to the UP traffic, only IPv4 is supported by implementing the GPRS Tunneling (GTP) protocol established by 3GPP.

To conclude, within the radio simulation domain, UERANSIM has yet to develop many of the features present in their project's roadmap. For instance, below there are listed the status of each planned feature associated with the radio interface:

- PHY: waiting;
- MAC: waiting;

- RLC: released;
- RRC: in progress;
- PDCP: waiting;
- Service Data Adaptation Protocol (SDAP): waiting.

2.1.1.3 Comparative

This section serves to compare both ecosystems, OAI and UERANSIM, regarding their main characteristics and features. Table *1* provides the comparison matrix.

Table 1: 5G RAN open-source features comparison

Feature	OAI	UERANSIM
5G Radio Interface	Yes	In progress
Control Plane	Yes	Yes
User Plane	Yes	No
PHY	Yes	No
MAC	Yes	No
RLC	Yes	No
RRC	Yes	In progress
PDCP	Yes	No
SDAP	In progress	No
IPv4/IPv6 Support	Both	IPv4 only

In order to conclude this section, the authors of this document recommend the use of OAI based on the following aspects:

- Provided coverage;
- Current implementation status and roadmap;
- Community and support;
- Available documentation;
- ORAN-Alliance (Alliance, 2021) compliance.

2.1.2 5GC

The core network is one of the main subsystems of a cellular network. In 4G it is referred as EPC and more recently in 5G it is called 5GC. The mobile core has the following main functionalities (Approach, 2021):

- Provide Internet (IP) connectivity for both data and voice services;
- Ensure this connectivity fulfil the promised QoS requirements;
- Track user mobility to ensure uninterrupted service;
- Track subscriber usage for billing and charging.

To simulate a 5G Core for this project, three different software solutions were compared to reach a conclusion of which would be more suitable for the project, as depicted in the next subsections. The analysed solutions include Open Air Interface for the Core project (OAI-CN), Open5GS and Free5GC.

2.1.2.1 OAI-CN

OAI is an open software that gathers a community of developers from around the world working together to build wireless cellular Radio Access Network (RAN) and more recently Core Network (CN) technologies (OpenAirInterface, 2021). It implements the 3rd Generation Partnership Project (3GPP) technology for 5G Core Networks and the open-source code is hosted in Github (OpenAirInterface, which can be adapted for different use cases and functionalities.

OAI has the objective of building an open cellular ecosystem for low-cost and flexible 4G/5G deployment and experimentations. The 4G/5G architecture functions using OAI can be checked in Figure 8, where the 5G network is represented with orange background, while the 4G LTE-EPC is represented with blue background.

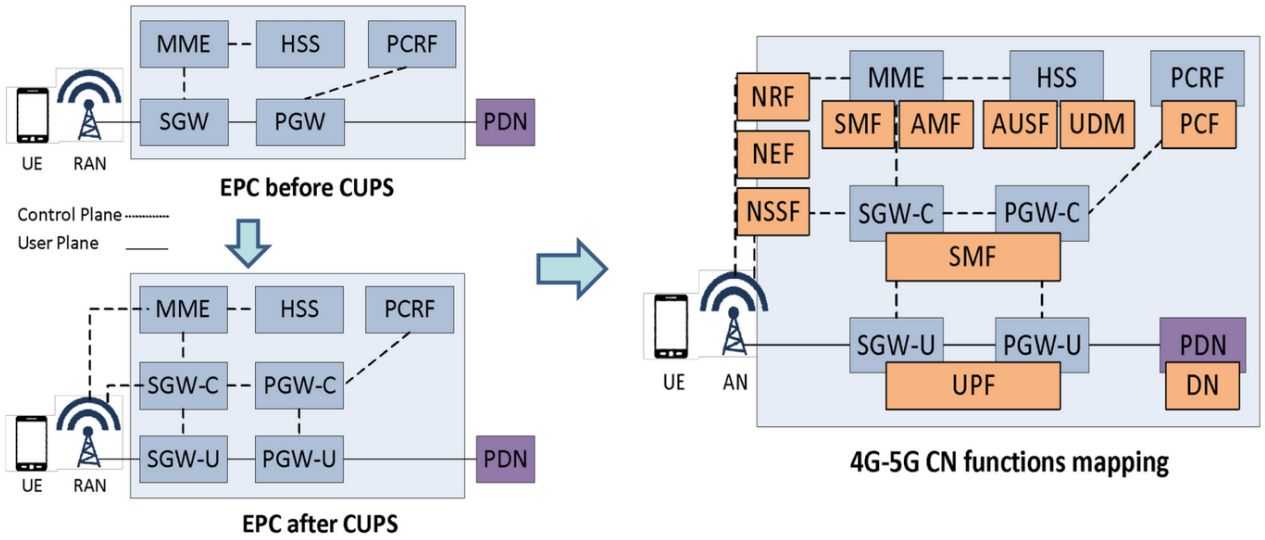

Figure 8: 4G and 5G CN block diagram for OAI

The CN project is still ongoing as shown on the OAI roadmap in Figure 9 (source (EURECOM, 2021)), with planned functionalities to be fully integrated bythe end of September 2022. Currently there are still some pending important updates such as NEF, UDSF and NWDAW integration as well as the support for URLLC.

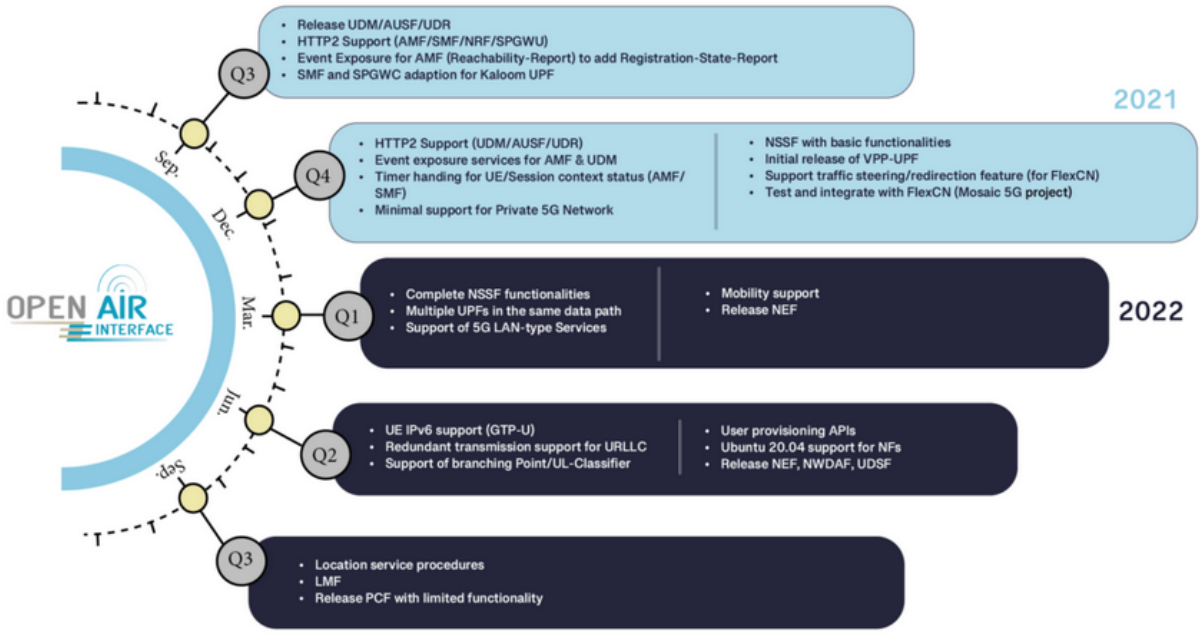

Figure 9: OAI 5G CN roadmap

Table 2 presents the main functionalities of Core 5G network included in Open Air Interface software.

Table 2: OAI-CN included or planned functionalities

Services/ Functionalities	OAI-CN	Note
AMF	YES	Present in official OAI roadmap
SMF	YES	Present in official OAI roadmap
AUSF	YES	Present in official OAI roadmap
UDM	YES	Present in official OAI roadmap
UDR	YES	Present in official OAI roadmap
NSSF	YES	Present in official OAI roadmap
UPF	YES	Present in official OAI roadmap
NRF	YES	Present in official OAI roadmap
PCF	Planned (09/2022)	Present in official OAI roadmap – Not released yet
BSF	No info found	
NEF	Planned (06/2022)	Present in official OAI roadmap - Not released yet
NWDAF	Planned (06/2022)	Present in official OAI roadmap - Not released yet
UDSF	Planned (06/2022)	Present in official OAI roadmap - Not released yet
AF	No info found	
N3IWF	Planned (06/2022)	Planned in roadmap for OAI 5G CN Project Group [14]

SMSF	Planned (06/2022)	Planned in roadmap for OAI 5G CN Project Group [14]
URLLC support	Planned (06/2022)	Present in official OAI roadmap - Not released yet

Important notes:

- The Policy Control Function (PCF), an important network function of a 5G Core network, is only planned in OAI official roadmap in September 2022 and with limited functionality. However, the policy control information (e.g. packet flow descriptions, QoS requirement) can be locally defined by AMF and Session Management Function (SMF). For example, in case of Application Function (AF) influence on traffic routing, the AF can send the request to SMF to influence UPF selection and route UE traffic to a Data Network.
- Network Data Analytics Function (NWDAF), which has a direct communication with PCF through N23 interface, can collect a wide range of information. For example, the statistics of the UE mobility, of the UE communication or of the Network Slice. It can also retrieve and send data from other NFs (such as AMF, SMF or UDM) and provides analytic services to other NFs and 3rd applications.
- The design of the 5GC is based on the Service-Based Architecture (SBA), a type of architecture standardized by 3GPP for 5G core networks. The 3GPP defines an SBA to include service-based interfaces between control plane functions, with user plane functions connecting over point-to-point links (3GPP, 3GPP TS 23.501, 2021).

2.1.2.2 Open5GS

Open5GS is a C-language Open Source project of 5GC and EPC that can be used to configure NR/LTE network. Figure 10 presents the basic architecture of the software.

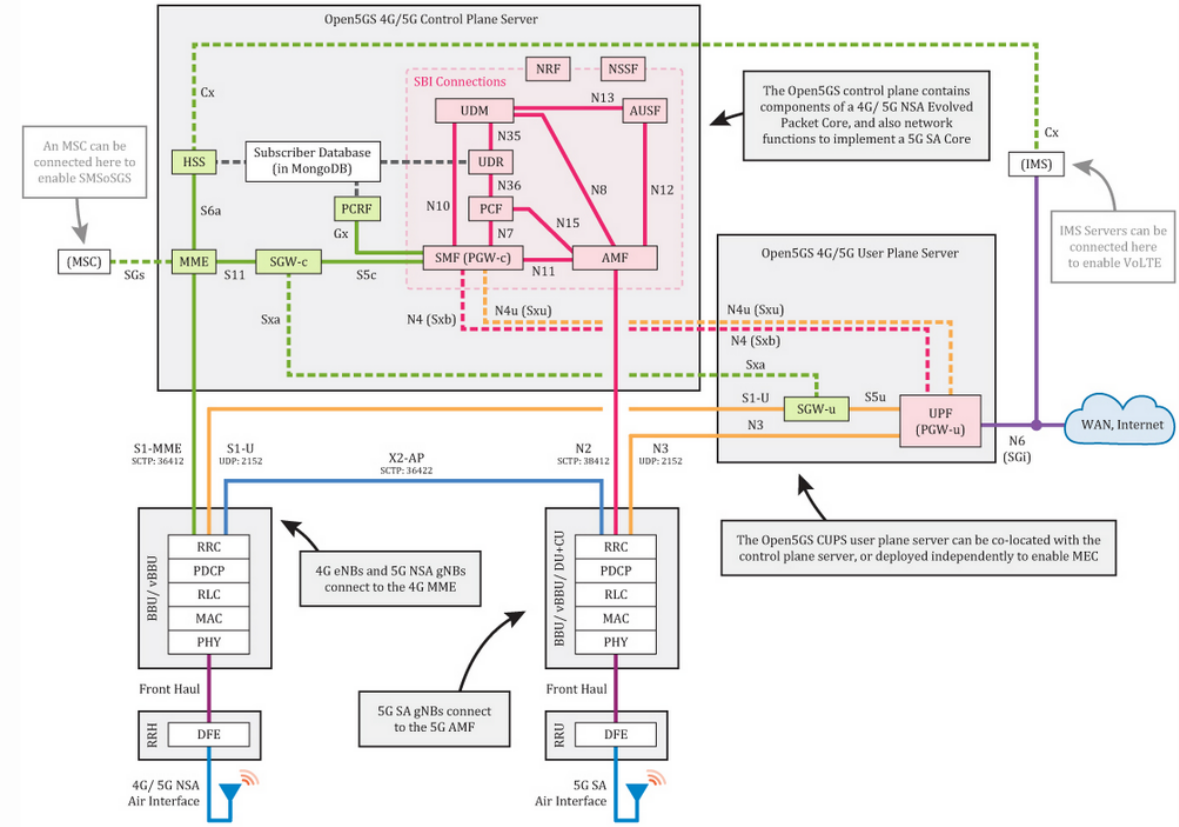

Figure 10: Open5GS Core network block diagram

The core has two main planes: the control plane and the user plane. These are physically separated in Open5GS as CUPS (control/ user plane separation) is implemented.

Table 3 depicts the main functionalities of Core 5G network included in Open5GS software (Open5Gs, .

Table 3: Open5GS included or planned functionalities

Function	Available	Note
AMF	YES	Present in official Open5GS roadmap
SMF	YES	Present in official Open5GS roadmap
AUSF	YES	Present in official Open5GS roadmap
UDM	YES	Present in official Open5GS roadmap
UDR	YES	Present in official Open5GS roadmap
NSSF	YES	Present in official Open5GS roadmap
UPF	YES	Present in official Open5GS roadmap
NRF	YES	Present in official Open5GS roadmap

PCF	YES	Present in official Open5GS roadmap
BSF	YES	Present in official Open5GS roadmap
NEF	No info found	
NWDAF	No info found	
UDSF	No info found	
AF	No info found	
N3IWF	No info found	
SMSF	No info found	
URLLC support	No info found	

Important notes

- These Open5GS components have configuration files which contain the component's IP bind addresses/ local Interface names and the IP addresses/ DNS names of the other components it needs to connect to.
- There are some known limitations on this software (Open5Gs, :

  - No Voice over NR;
  - No Interworking with EPC;
  - No NB-IoT;
  - No OCS/OFCS;
  - No eMBMS;
  - No SRVCC;
  - No Roaming;
  - No Emergency Call.

2.1.2.3 Free5GC

Free5GC is an open-source project for 5G mobile core networks where it's possible to implement the 5GC defined in 3GPP from release 15 (Free5G, 2021). Currently the biggest contributor for this project is the National Chiao Tung University and the project was composed with 3 stages where the last one was released in 2020 (Free5G, 2021):

- Stage 1 (in January 2019): migrated 4G Evolved Packet Core into 5GC Service-Based Architecture (SBA);
- Stage 2 (in October 2019): implementing the standalone 5GC features;
- Stage 3 (in April 2020): a fully operational 5GC.

The demo architecture diagram for this project for the last stage is present in Figure 11 (Free5G, 2021).

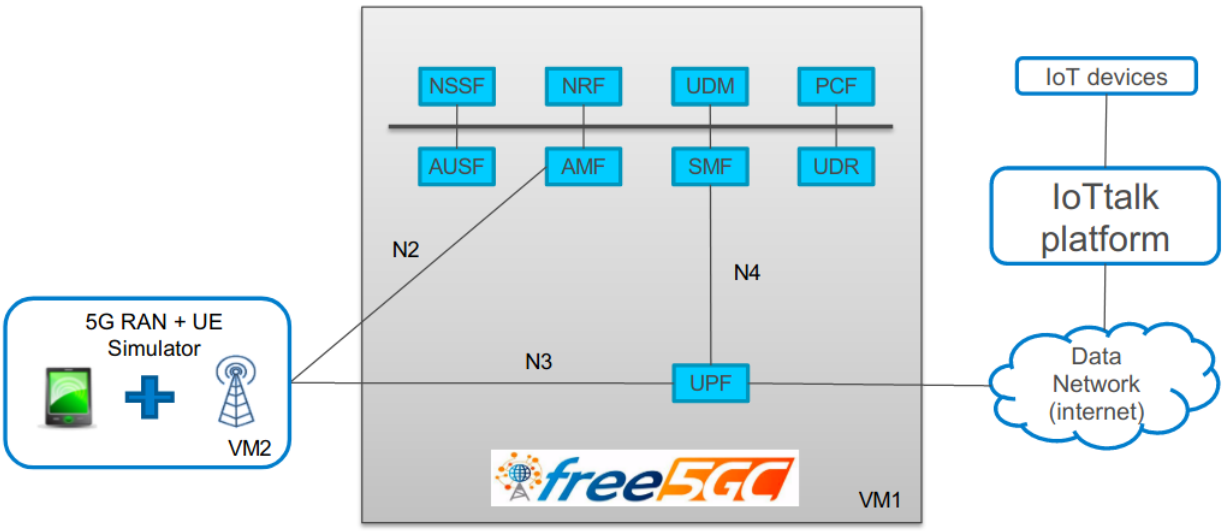

Figure 11: Free5GC Core network block diagram

Table 4 presents the main functionalities for Core 5G network included in Free5GC software.

Table 4: Free5GC included or planned functionalities

Function	Available	Note
AMF	YES	Present in official Open5GS roadmap
SMF	YES	Present in official Open5GS roadmap
AUSF	YES	Present in official Open5GS roadmap
UDM	YES	Present in official Open5GS roadmap
UDR	YES	Present in official Open5GS roadmap
NSSF	YES	Present in official Open5GS roadmap
UPF	YES	Present in official Open5GS roadmap
NRF	YES	Present in official Open5GS roadmap
PCF	YES	Present in official Open5GS roadmap
BSF	No info found	
NEF	No info found	
NWDAF	No info found	
UDSF	No info found	
AF	No info found	
N3IWF	YES	Present in official Open5GS roadmap
SMSF	No info found	
URLLC support	No info found	

Important notes

- Added interfaces in stage 2: SBI N1 (UE - AMF), N2 (AN - AMF), N8 (Namf - Nudm), N10 (Nsmf – Nudm), N11 (Namf - Nsmf), N12 (Namf – Nausf), and N13 (Nausf – Nudm) reference points;

- Added features 5G Next Generation Application Protocol (NGAP), 5G Non-Access Stratum (NAS), 5G authentication, handover procedure, paging, PCF and QoS (to RAN) in stage 2 of roadmap. In stage 3 were added also Operation, Administration and Management (OAM) of 5GC, 5G Orchestrator, non-mobile access network (N3IWF) and UP Uplink Classifier (ULCL).

2.1.2.4 Comparative

Table 5 presents a functionality comparison for the three open-source solutions described considering the most important functionalities of 5G Core Network.

Table 5: Software simulators for 5G Core

Function	OAI-CN	Open5GS	Free5GC
AMF	YES	YES	YES
SMF	YES	YES	YES
AUSF	YES	YES	YES
UDM	YES	YES	YES
UDR	YES	YES	YES
NSSF	YES	YES	YES
UPF	YES	YES	YES
NRF	YES	YES	YES
PCF	Planned (09/2022) Planned (04/2022)*	YES	YES
BSF	No info found	YES	No info found
NEF	Planned (06/2022)	No info found	No info found
NWDAF	Planned (06/2022)	No info found	No info found
UDSF	Planned (06/2022)	No info found	No info found
AF	No info found	No info found	No info found
N3IWF	Planned (06/2022)*	No info found	YES
SMSF	Planned (06/2022)*	No info found	No info found
Interface N9	No info found	No info found	No info found
MEC support	No info found	No info found	No info found
URLLC support	Planned (06/2022)	No info found	No info found

Currently Open5GS and Free5GC seem to be the core software simulators with more functionalities, followed by the OAI-CN. However, until the end of September 2022, significant updates are planned to be included for OAI-CN, making it the most complete and powerful software to be used in the near future. Free5GC project was already released in 2020 and Open5GS is being regularly updated, although no information was found about what will be updated or included for the future.

NWDAF is one function that is considered important for this project and for now the only platform that plans to include it in the near future will be OAI. It is possible for us to include NWDAF in other platforms, but there is a risk that integration will not be straightforward when mixing 5GC NFs from different vendors.

Documentation available

OAI-CN appears to be the software with more documentation available, followed by Free5GC.

Conclusion

Taking into account the network functions that will be available in the near future for the three software tools and also the documentation available that could support the simulation of a 5G Core, the Open Air Interface seems to be the most suitable option for the end of 2022. NWDAF is considered an important function to be used for this project and the only software currently with the plan to include is OAI.

2.1.3 MEC

Multi-access Edge Computing (MEC) offers cloud computing capabilities and an IT service environment at the edge of the network. This environment can be described by high bandwidth, Ultra Low Latency and also real-time access to information from radio network that can be used by applications.

MEC provides a new ecosystem and value chain where operators can open their RAN edge to authorize easy access and flexibility to third parties which enables the deployment of services or applications towards mobile subscribers.

The MEC initiative is an Industry Specification Group (ISG) within the European Telecommunications Standards Institute (ETSI). The purpose of the ISG is to create a standardized, open environment which will allow the efficient and seamless integration of applications from vendors, service providers, and third parties across multi-vendor MEC platforms (Foundation, 2020).

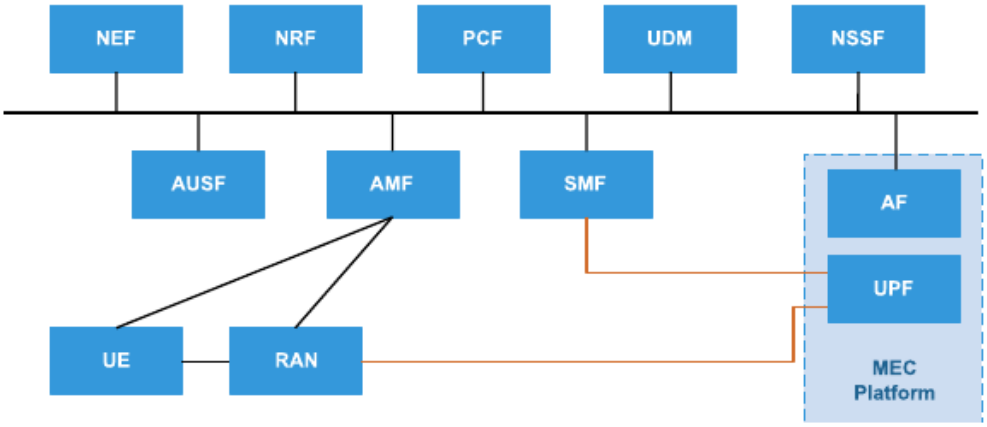

Figure 12: Overview of MEC in a 5G network (Filali, 2020)

MEC is a natural development in the evolution of mobile base stations and the convergence of IT and telecommunications networking. MEC will enable new vertical business segments and services for consumers and enterprise customers. Use cases include (ETSI, 2021):

- Video analytics;
- Location services;
- Internet-of-Things (IoT);
- Augmented reality;
- Optimized local content distribution;
- Data caching.

2.1.3.1 Intel Smart Edge Open

Open Network Edge Services Software (OpenNESS), now called Intel Smart Edge Open is an open-source edge computing software toolkit that allows building platforms optimized for the edge, capable of hosting several services and functionalities such as 5G Core or 5G RAN (Intel, 2021). When compared to cloud platforms, edge platforms require higher network performance, autonomy and have resources limitations (Intel, 2021). Figure 13 presents the Intel Smart Edge Open architecture (Intel, 2021).

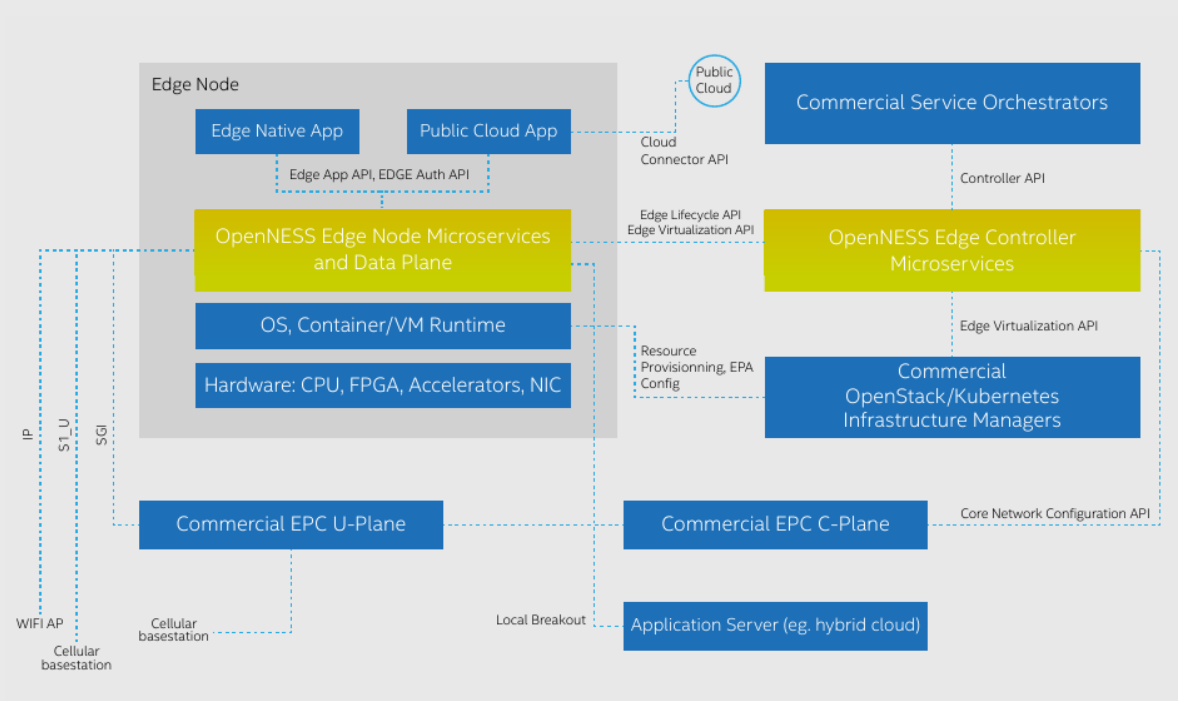

Figure 13: Intel Smart Edge Open architecture

The Intel Smart Edge Open edge node manages the edge services, including the APIs used to discover those services. Features of the edge node include (Intel, 2021):

- Support for the Docker container runtime and virtualization infrastructure (libvirt*, Open vSwitch, etc.) to support Virtual Machines (VM);
- Platform pods consisting of services that enable the configuration of hardware resources on the node for a particular deployment, operators for accelerators and device plugins enabling hardware resource allocation to an application pod;
- System pods consisting of services that enable reporting the hardware and software features of each node to the control plane, providing resource isolation service for pods and DNS service to the cluster;
- Agents that expose edge node configuration to the control plane services (DNS, 4G, 5G, Wi-Fi and Telemetry);
- Telemetry for the edge node at the hardware, operating system, infrastructure, and application levels. This provides user the ability to monitor performance and health of cluster nodes which allows to receive information about the platform, the underlying hardware, cluster and applications deployed. The data collected by telemetry can be used to visualize metrics and resource consumption, set up alerts for certain events and assist in making scheduling decisions based on the received telemetry.

Currently, the support for telemetry is focused on metrics; support for application tracing telemetry is planned in the future. The telemetry components used in Intel® Smart Edge Open are deployed from the Edge Controller as Kubernetes pods, which include:

- Collectors;
- Metric aggregators;
- Monitoring and visualization tools.

The telemetry components can be deployed as *Deployment* or *Deamonset*, depending on the role of the component. Usually, the global components that receive inputs from local collectors are deployed as *Deployment* type with a single replica set and local collectors running on each host are deployed as *Daemonsets*. Local collectors running on Edge Nodes that collect platform metrics are deployed as privileged containers, using host networking. Communication between telemetry components is secured with TLS either using native TLS support for a given feature or using a reverse proxy running in a pod as a container. All the components are deployed as Helm charts. The deployment of telemetry components using this software can be configured with the open-developer-experience-kits (DEK).

2.1.4 Controllers

In the following subsections, the two most popular controllers for 5G RAN and CN will be described.

2.1.4.1 FlexRIC

FlexRIC is a Flexible and programmable RAN Intelligent Controller for Software-Defined Radio Access Networking (SD-RAN). It has interfaces with the OAI radio stack over the O-RAN-defined E2-interface to monitor and control the RAN in real-time. It supports real-time monitoring and control for 4G and 5G RAN. FlexRIC is composed of a RAN agent that allows communication with the radio stack and a real-time controller. This software has a monitoring and Slice-service Model (SM) that can be customized to be used in 5G use cases and permit SM creation at any time. It will act as a booster for Machine Learning (ML) algorithms deployed in 5G and can enable real 5G deployments. It is designed to be lean without adding unnecessary features. The adoption of the E2 protocol structure becomes fully compliant with the industry, and it is vendor-independent. Its modular SDK structure permits a smooth composition of specific controllers, including radio scheduling.

FlexRIC SDK consists of a server library and an agent library with extension to a controller-internal application (iApps) and communication interfaces. Its objective is to facilitate the realization of specialized SD-RAN controllers to target specific use cases while being simple to use. In the simple form, the FlexRIC SDK can be used to implement an SD-RAN controller using the E2 protocol. It has an agent library that provide an API that

permits the implementation, monitoring and control of RAN functions. The controller is built using the server library, iApps and can have the communication interface or not. The server library has the function to manage the agent connection and deliver the messages between iApps and agents. Using the iApps is possible to build specialized modular controllers based on each use case. It can implement SMs themselves or expose information to the xApps via northbound interfaces. The FlexRIC SDK provides an abstraction of the E2 interface via an internal representation of E2 messages, permitting the user not to be concerned with encoding and transporting the messages. It makes it easier to integrate the FlexRIC with the existing SD-RAN infrastructure. However, the standard imposes the encapsulation of ASN.1-encoded data inside of ASN.1 and transport over the SCTP protocol. This may cause some inefficiency when the data size is too large. So, the FlexRIC SDK has the encoding scheme and transport protocol allowing a more efficient integration.

Figure 14 illustrates the FlexRIC SDK, having an agent and a server library where the library in the Agent provides an integration with the controller in the base station that can be used by an iAPP;

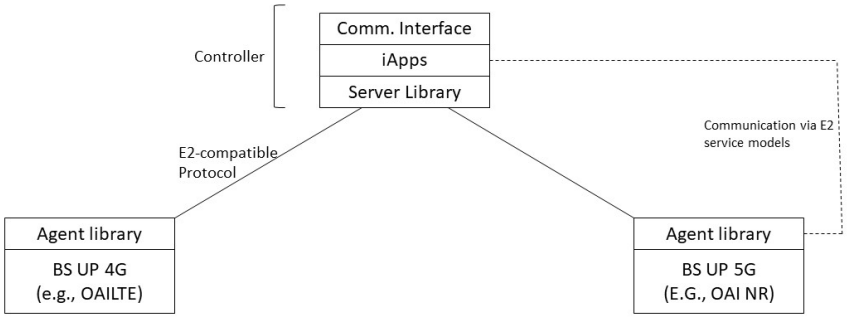

Figure 14: FlexRIC SDK

- FlexRIC Agent provides easy integration with various base station implementations and deploying several scenarios interfacing with an E2 controller. The Agent provides the necessary support to connect to a controller such as O-RAN RIC. It consists of the E2AP abstraction, a message handler and generic RAN functions. The Agent also provides the generic RAN function API to implement the RAN function with the custom SM logic. The API call for E2AP messages, the subscription request, subscription delete request, and control messages. Finally, considering the use cases, the SMs, namely the slicing control and traffic control. The base station provides basic node information, such as the public land mobile network (PLMN), and registers RAN functions according to the underlying node's capability. It uses the interface of pre-defined RAN functions to expose data and handle control messages, as depicted in Figure 15;

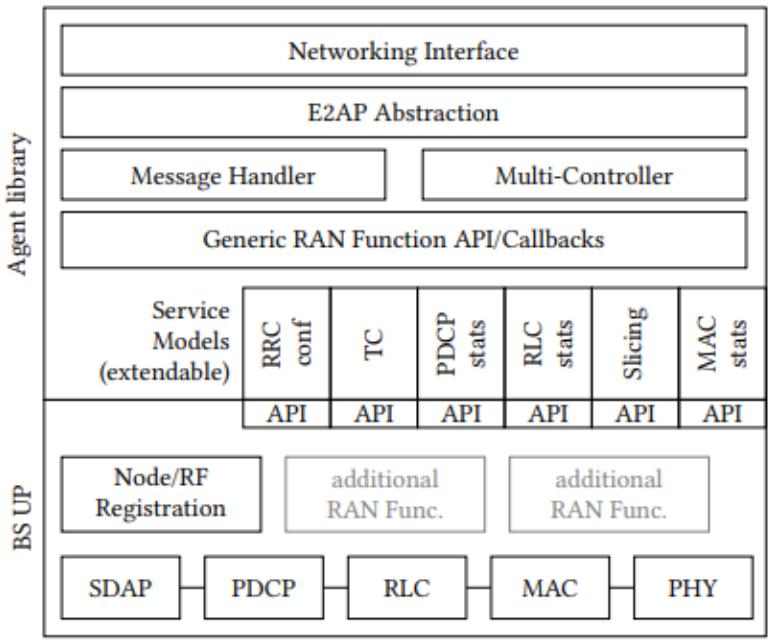

Figure 15: FlexRIC agent architecture and integration with user plane

- FlexRIC Controller consists of the FlexRIC server library, which communicates with the agents using an E2AP abstraction and a controller specialization. The controller specialization implements functionalities related to SD-RAN and is realized with the internal applications. These iApps implement specific controller behavior either using SMs or allowing control through the xApps. A specialized control typically exposes a northbound interface using custom protocols (like REST or E2AP) to act with other controllers. The FlexRIC Server Library has the objective to control the agent connections and dispatch the E2AP messages. It is designed as an event-driven system, and it invokes iApps only when new messages arrive at the system, different from FlexRan that uses polling. The RAN management functionality manages connection-related events such as agent connections, storing information into the database, and disaggregating deployments merging agents that belong to the same base station, facilitating control across agents. When iApps request a new subscription directly or on behalf of a xApp the subscription management receives a message about this subscription. It simply selects the iApp for which the message is sent and forwards it to the provided callback. Figure 16 depicts the connection between xApp and the FlexRIC server.

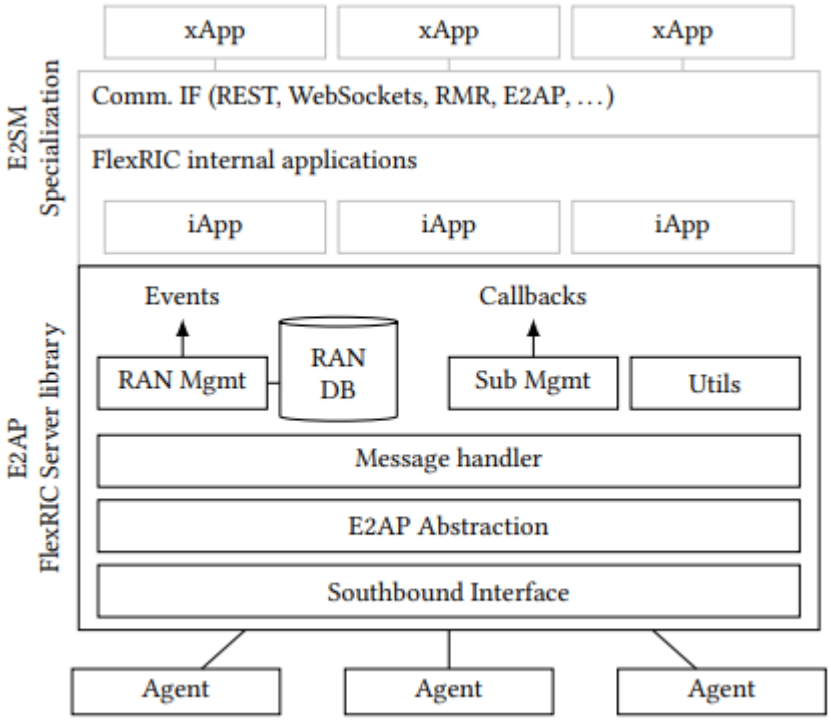

Figure 16: FlexRIC server library and E2AP abstraction

2.1.4.2 FlexCN

FlexCN is a solution that brings programmability to the Core Network, a characteristic needed in the 5G networks. It allows users to manipulate the Core Network to satisfy their requirements. This enables the delivery of the benefits of the 5G networks.
This controller's purpose to enable mobile network monitoring, control, and programmability while having 3GPP compatibility and ETSI specifications.
The FlexCN can be used on end-to-end slicing scenarios by leveraging SDN towards an appropriate resource allocation, improving the performance of slices, or in use cases that demand a lot of resources but need low latency and high reliability.

2.2 Container and cloud tools

The next subsections describe the container and cloud tools that are relevant to the OREOS platform.

2.2.1 OpenStack

OpenStack is an open-source software for building private and public clouds, that controls large pools of compute, storage, and networking resources throughout a datacentre, as depicted in Figure 17.

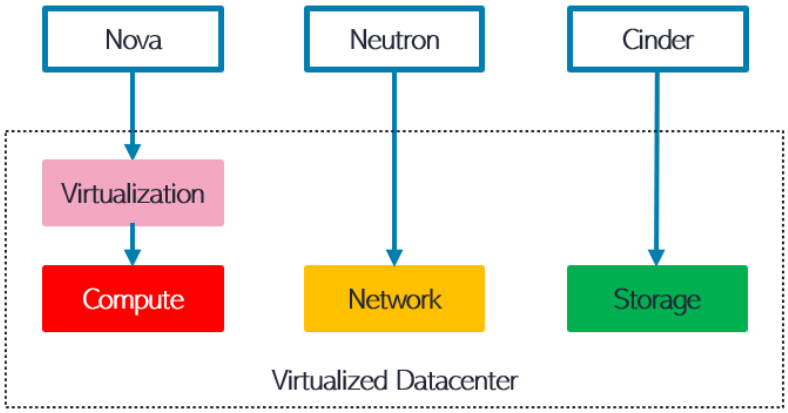

Figure 17: OpenStack

- Nova supports various hypervisors for virtual machines such as KVM, and VMware. It also supports Linux Containers. Its purpose is to implement services and associated libraries to provide massively scalable, on demand, self-service access to compute resources, including bare metal, virtual machines, and containers;
- Neutron provides an API for creating ports, subnets, networks, and routers. Additional network services such as firewalls and load balancers are provided in some OpenStack deployments. It is focused on delivering networking-as-a-service (NaaS) in virtual compute environments;
- Cinder provides block storage to the Nova virtual machines. Its subsystems include a volume manager, a SQL database and an authentication manager. It virtualizes the management of block storage devices and provides end users with a self-service API to request and consume those resources without requiring any knowledge of where their storage is actually deployed or on what type of device. This is done through reference implementations or plugin drivers for other storage.

2.2.2 Containers

A container is an executable unit of software that packages application code with its dependencies, enabling it to run on any IT infrastructure. The biggest advantage of containers is that a container stands alone, i.e., it is abstracted away from the host operating system.

Docker is a platform that makes it easy to develop, deploy, and run applications as standalone, portable containers. Figure 18 illustrates the concept of containerization applied to the OREOS project.

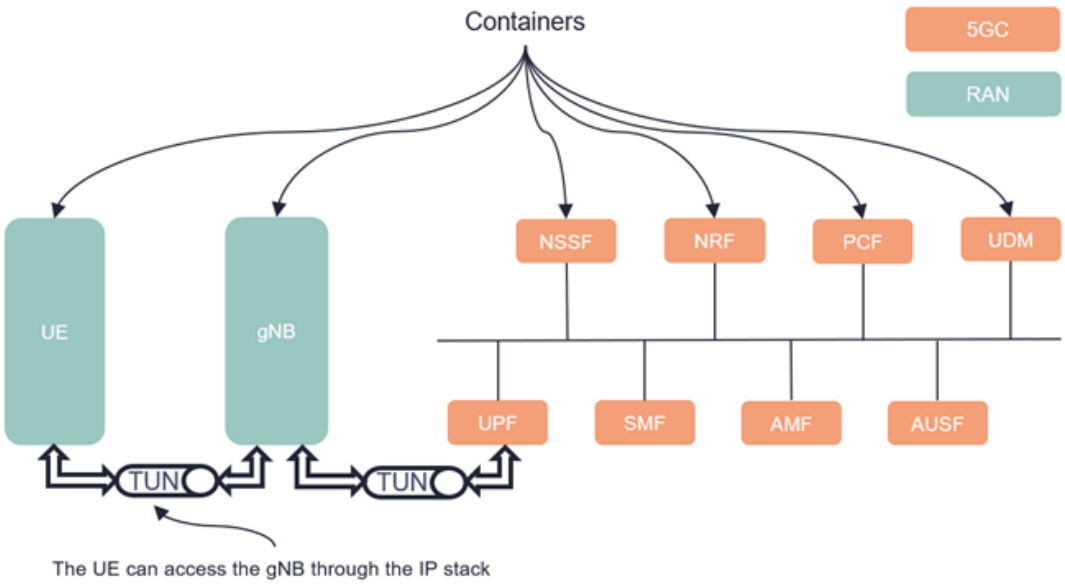

Figure 18: Containerization in 5G networks

2.2.3 Kubernetes

In simple words, Docker is a container and Kubernetes is a container orchestration tool, something which can create, destroy, and manage containers at scale. Both are complementary, as Kubernetes does not create containers. It requires a container tool to run, of which Docker is the most popular option.

Kubernetes distributes and schedules containerized applications across a cluster of physical or virtual machines—rather than running them on a server. This way, applications running in Kubernetes function like a single entity, although they may comprise an assortment of loosely coupled containers. In other words, Kubernetes clusters allow containers to run across multiple machines and environments: virtual, physical, cloud-based, and on-premises.

Two important concepts are clusters, nodes and pods:

- A Kubernetes cluster includes a container designated as a *master node* that schedules workloads for the rest of the containers — or *worker nodes* — in the cluster. The master node determines where to host applications (or Docker containers), decides how to put them together and manages their orchestration. By grouping containers that make up an application into clusters, Kubernetes facilitates service discovery and enables management of high volumes of containers throughout their lifecycles;
- A node is a VM or a physical computer that is used as a worker machine in a Kubernetes cluster. Every node from the cluster is managed by the master. A node may be a virtual or physical machine, depending on the cluster. A Kubernetes cluster is a set of nodes that runs containerized applications. By definition, a node is always considered to be a member of a cluster;

- Kubernetes does not run containers directly. Instead, it wraps one or more containers into a higher-level structure called a pod, as illustrated in Figure 19. Any containers in the same pod will share the same resources and local network.

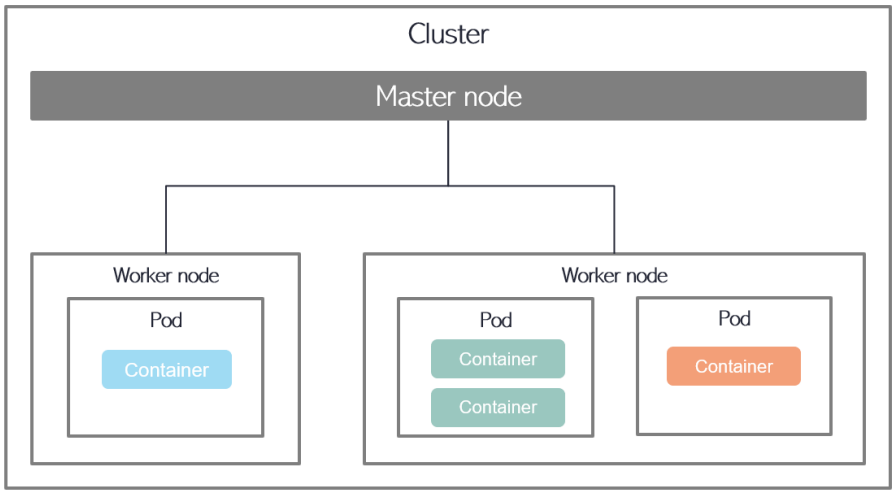

Figure 19: Kubernetes

2.3 Network design

The following figure (Florian Kaltenberger a)[1] illustrates the high-level architecture of the OREOS platform, considering the chosen open-source external components (inspired from (Meng, 2020)).

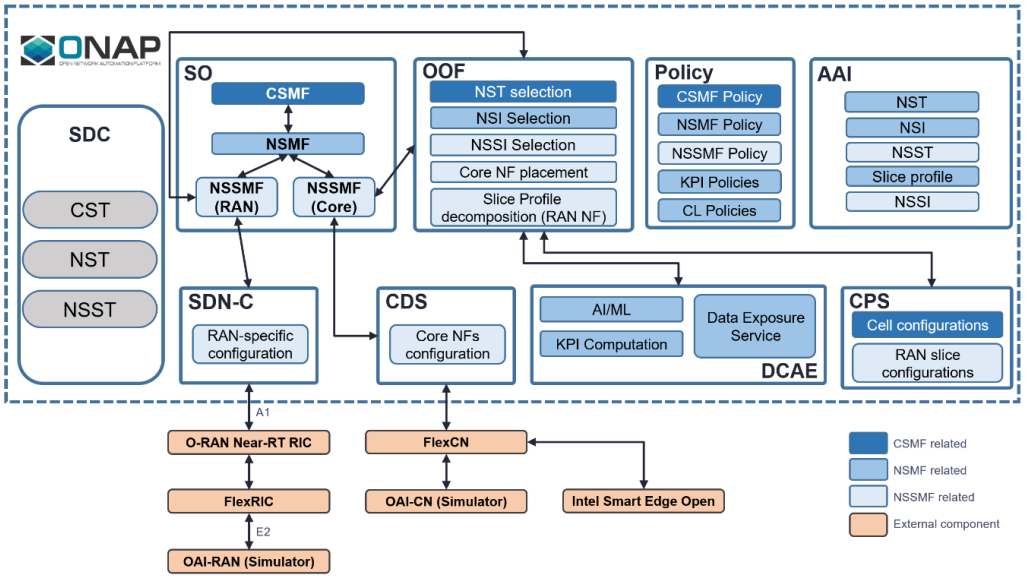

Figure 20: Low-level design of ONAP

[1] The transport network was hidden, but its external component connects to the SDN-C (Bhattacharjee, et al., 2021).

Two new important ONAP components are illustrated in this figure:

- ONAP brings cloud native functionality with seamless configuration of Helm based Cloud Network Functions (CNF) and K8s resources. This functionality is implemented in the Controller Design Studio (CDS);
- The Configuration & Persistency Service (CPS) is a platform component that is designed to serve as a data repository for run-time configuration and operational data that needs to be persistent. CPS offer the ability for service operators to visualize and manage data in a RAN network (Physical Network Functions (PNF), Virtual Network Functions (VNF), and logical constructs, such as RAN cells).

3GPP in its technical specification TS 28.801, defines three-layer slice management functions[2] which include:

- CSMF (Communication Service Management Function): responsible for translating the communication service-related requirement to network slice related requirements and for the communication with Network Slice Management Function (NSMF);
- NSMF (Network Slice Management Function): responsible for the management and orchestration of Network Service Instance (NSI), to derive network slice subnet related requirements from network slice related requirements and for the communication with the Network Slice Subnet Management Function (NSSMF) and CSMF;
- NSSMF (Network Slice Subnet Management Function): responsible for management and orchestration of Network Slice Subnet Instance (NSSI) and for the communication with the NSMF. Table 6 illustrates the functions and classification of NSSMF in ONAP[3].

Table 6: NSSI functionalities mapped to ONAP components

Function	Description	ONAP Component(s)
NSSI Allocation	Determine if an existing NSSI can be used, or a new one is needed	OOF (triggered by SO)
	Determine resources to be allocated to the NSSI (for a new NSSI)	OOF (triggered by SO)
	Instantiate/scale up necessary (virtual) resources (using NFVO)	SO
	Perform necessary (re) configuration (incl. NSSI stitching configs)	Domain controllers
NSSI (De)Activation	(De)Activate the NSSI resources	Domain controllers

[2] In ONAP, the Service Orchestrator (SO) already includes these functions.
[3] SO triggers the NFVO (ref. 3GPP) to perform all virtual resource allocation & orchestration actions related to an NSSI. NFVO may be slice unaware. Domain Controllers – include SDN-C (R), VF-C, CDS (CDS may be used to configure VNFs).

NSSI Modification	Determination of modifications to be done	OOF
	Resource modification (re-allocation, scaling)	SO, Domain controllers
	(Re)Configuration of resources	Domain controllers
NSSI details exposure	Provide details of NSSI to NSMF, inventory (resources, capabilities)	SO, Domain controllers
NSSI Closed Loop	Perform closed loop control actions such as scaling	SO, SDN-C
NSSI De-Allocation	Tear down the NSSI and reclaim resources used	SO
	Update configurations	Domain controllers

Figure 21 illustrates the state-diagram of the affected components.

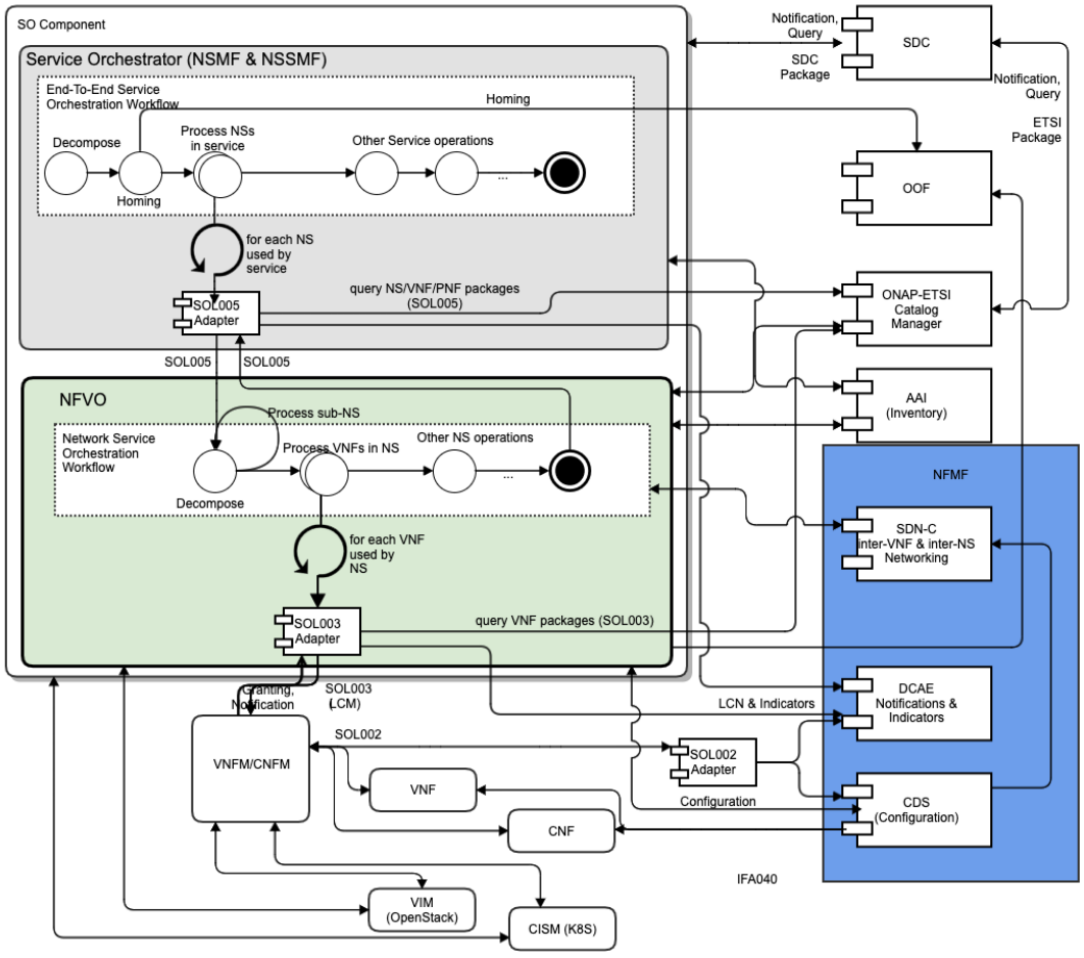

Figure 21: State-diagram of ONAP

Appendix – Slice Instances & Templates provides an exhaustive description of the templates and instances.

2.3.1 Network slicing

Vertical industries are very diverse, and their requirements are defined by the service characteristics of the vertical segment. For example, for an enhanced Mobile BroadBand (eMBB) service it makes sense for the UPF to be placed in a regional Data Center, in order to maximize the number of subscribers attached to it; while for an Ultra-Reliable Low Latency Communications (URLLC) service, the UPF must be located at the nearest point of the service subscriber in order to minimize latency while simultaneously maximizing reliability (but at the cost of having only a few subscribers attached to it). The following figure[4] illustrates this concept (Samsung, 2020) , (Huawei, 2020).

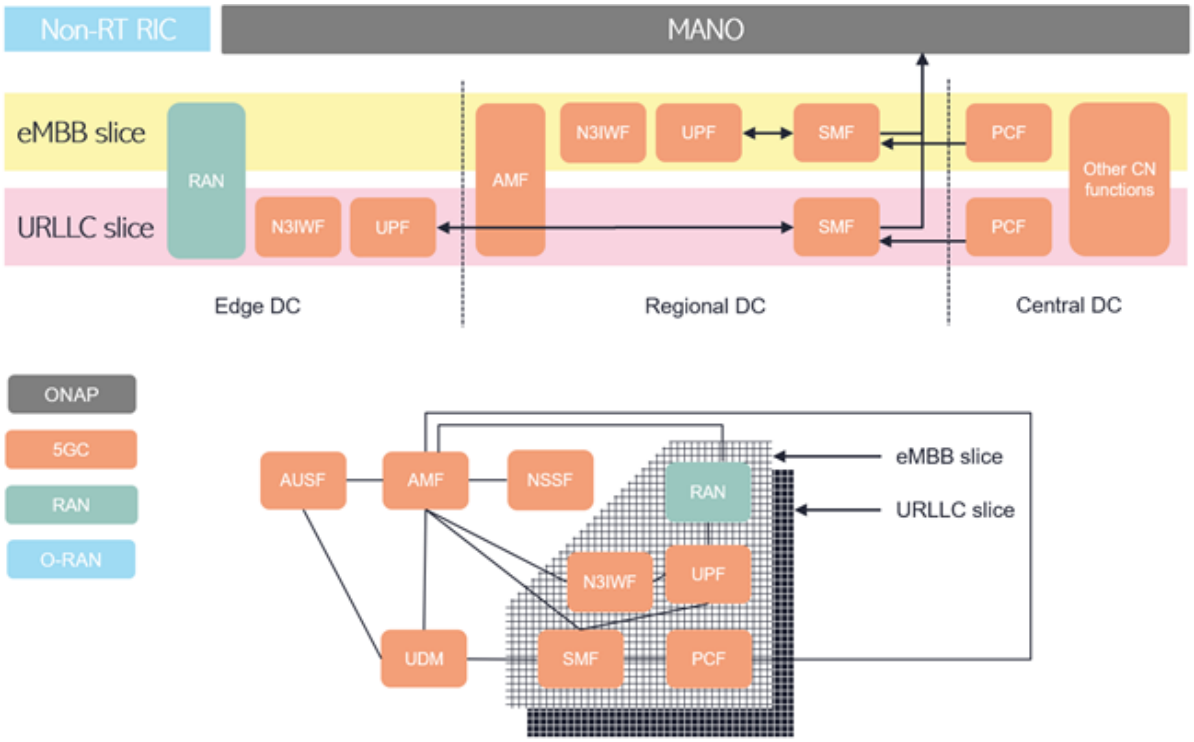

Figure 22: High-level view of network slicing in 5G networks

Typically, an Edge datacenter is the place where the base station – more specifically the CU – is located; while the regional datacenter is the place where the Optical Line Terminal (OLT) is located, i.e., the optical fibre central office. This configuration is very efficient, because in the case of the Edge datacenter it permits the quasi-simultaneous orchestration of resources at both AF level and at the RAN level; while in the case of the Regional

[4] The N3IWF is only used for non-3GPP communications. For this project we will assume that all communications are 3GPP compliant, and thus, we will not implement this network function.

datacenter it permits the quasi-simultaneous orchestration of resources at both Transport Network (TN) level and CN level.

Some of the network functions can be shared between network slices, while some network functions are deployed only for a specific service within a sliced network. For RAN, the RAN Intelligent Controller (RIC), defined by O-RAN, is added as a component for quick policy application and real time control.

- Management and Orchestration (MANO) oversees network slice capacity, SLA, and network health monitoring. It has a responsibility to provide life cycle management, reservation, assign, scale, and control functions for compute, storage and network resources per RAN, TN, and CN domain;
- Near-RT RIC performs slicing optimization based on real-time monitoring and control of RAN data;
- Distributed Unit (DU), Centralized Unit - Control Plane (CU-CP), and Access and Mobility Management Function (AMF) are typically shared by several network slices;
- CU-UP, Session Management Function (SMF) and User Plane Function (UPF) are typically dedicated to particular network slices;
- User Data Management (UDM) and Network Slice Selection Function (NSSF) are shared by all network slices;
- Network Repository Function (NRF) and Policy Control Function (PCF) can be common or network slice specific;
- The slice configured by the SLA provides isolated services for each divided bearer path (DU, CU-UP, UPF, and DN).

Figure 23 illustrates an example of a network with three slices.

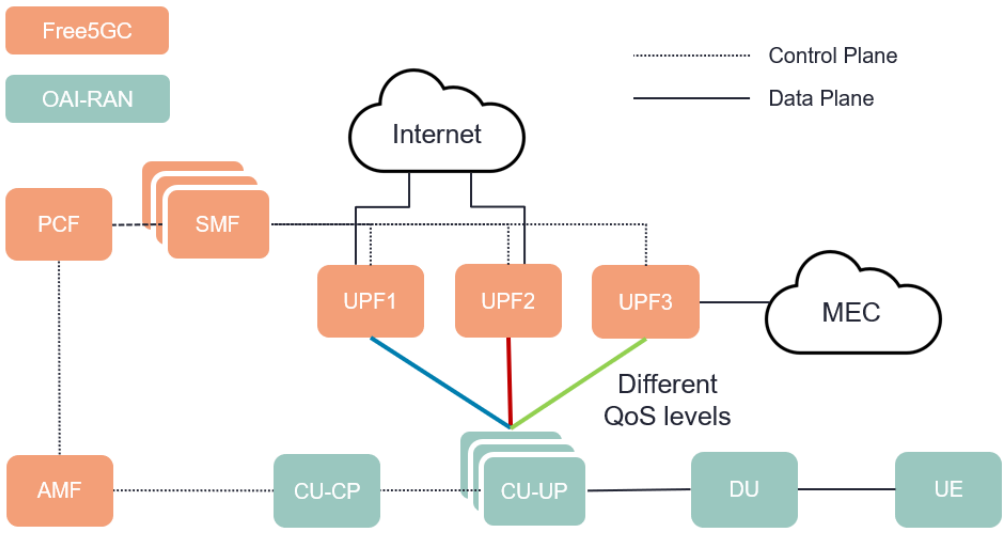

Figure 23: 5G network slicing example

In this example, the slicing occurs on the CU-UP at RAN level and on the SMF at CN level. Note that at CN level the slicing is optional on the PCF and also on the NRF.

2.3.1.1 5GC

Slicing at the CN level is simple since it consists of finding the optimal UPF location for the service, as illustrated in Figure 24.

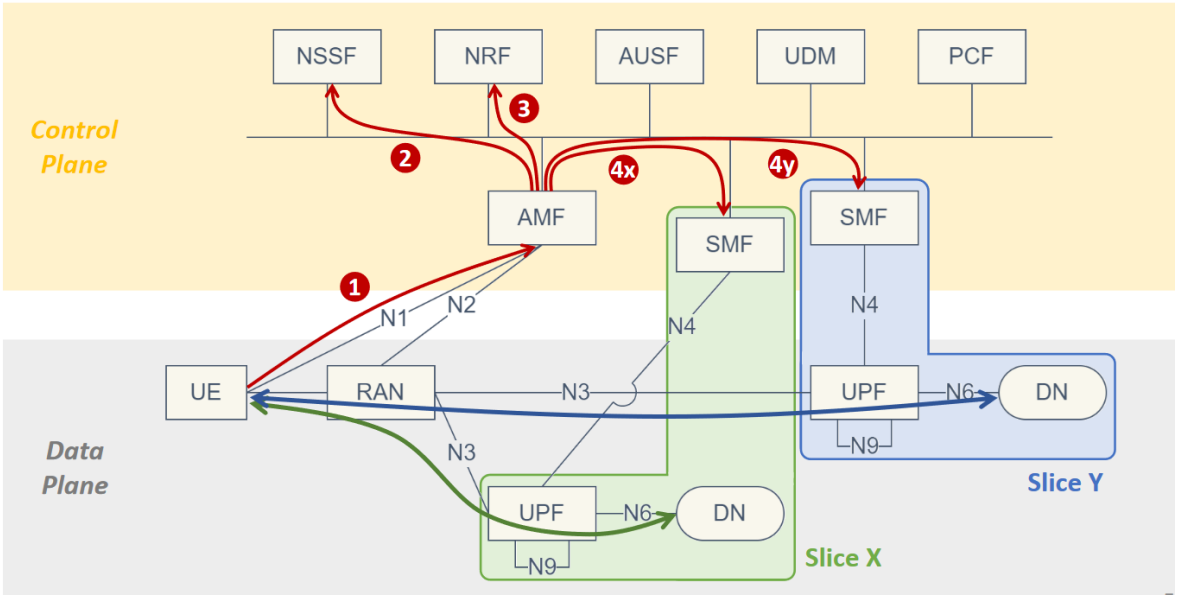

Figure 24: Slicing at 5GC

The setup process is triggered as part of the registration procedure by the first Access and Mobility Management function (AMF) that receives the registration request from the UE. The AMF retrieves slices accessible to the user and by interacting with the Network Slice Selection Function (NSSF) and selects the most appropriate slice instance. Then the AMF checks with the Network Resource Function (NRF) the locations of the UPFs and assigns the most appropriate instance of the Service Management Function (SMF) for the service requested. Afterwards, the Policy Control Function (PCF) sets routing control on the SMF instance and traffic is routed to the most appropriated User Plane Function (UPF).

2.3.1.2 RAN

Slicing at the access network offers much more possibilities than simply traffic routing. Figure 25 depicts four possible RAN slicing configurations:

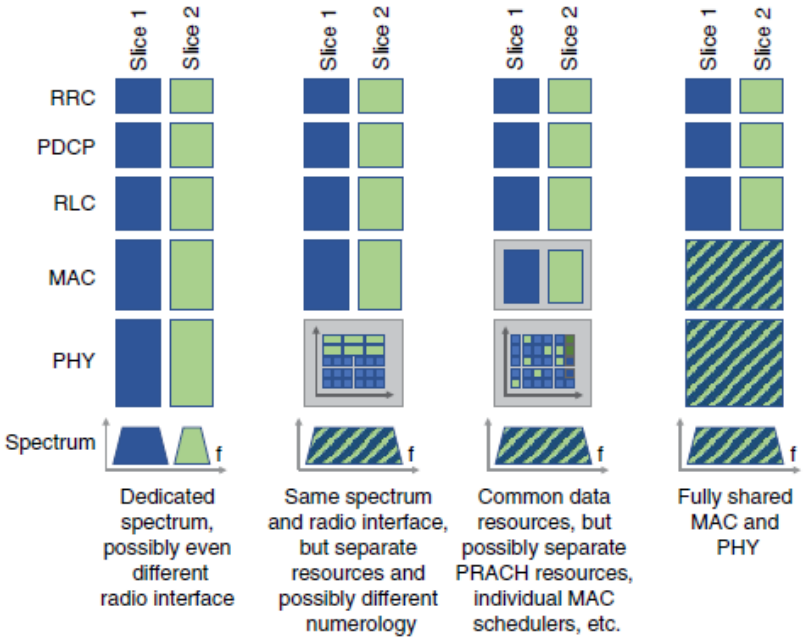

Figure 25: RAN slicing possibilities

- Physical layer (PHY) provides data transport services (synchronization of time and frequency, MIMO antenna signals processing, beam forming, etc.) to the higher layers, being defined in a bandwidth agnostic way, i.e., allowing it to adapt to various spectrum allocations;
- The MAC (Medium Access Control), RLC (Radio Link Control) and PDCP (Packet Data Convergence Protocol) are responsible for the mapping of logical channel onto physical channel and for data delivery assurance between the network nodes;
- RRC (Radio Resource Control) is responsible for intercell interference control and also for mobility management (including admission control and load balancing).

Summarizing, there are three frameworks for wireless virtualization:

1) Flow-based virtualization: deals with the isolation, scheduling, management and service differentiation between traffic flows;
2) Protocol-based virtualization: allows to isolate, customize and manage multiple wireless protocol stacks on a single hardware. Consequently, each tenant can have their own MAC and PHY configuration parameters;
3) Spectrum-based virtualization: focuses on the abstraction and dynamic allocation of spectrum and decouples the RF frontend from protocols, allowing a single frontend to be used by multiple virtual nodes.

Frameworks (2) and (3) run on the RAN side.

2.3.1.3 Slice creation across the network

Once ONAP starts initiating a network slice instance, the Network Function Virtualization Orchestrator (NFVO) / Virtual Network Function Manager (VNFM) determines which Network Functions (NF) are needed to establish the virtual network resources (VMs/Containers). The Element Management System (EMS) sends the provisioning data to the instantiated NFs, and the Software Defined Networks (SDN) controller connects the NFs. The NSSF records the network slice instance information, so that when a UE requests the service, the NSSF can deliver the target AMF and Network Slice Selection Assistance Information (NSSAI) for the service. In addition, ONAP optimizes network resources by supporting auto-scaling, which increases or decreases the network slice resources according to the traffic demands being placed on the network slice.

Once the UE starts an application that needs a network slice, the UE receives an NSSAI that corresponds to the service needs. For a user needing an eMBB network slice, the Central Unit Control Plane (CU-CP) that acquires the NSSAI from the UE, recognizes that it needs the eMBB network slice, selects the specific Central Unit User Plane (CU-UP) and AMF. Then the AMF selects an eMBB-specific SMF, and the SMF selects eMBB-specific UPF(s). The RAN's DU acquires the NSSAI from the CU-CP and allocates the dedicated resources to the slice, such as high bandwidth and capacity resource pooling. The Figure 26 illustrates this concept.

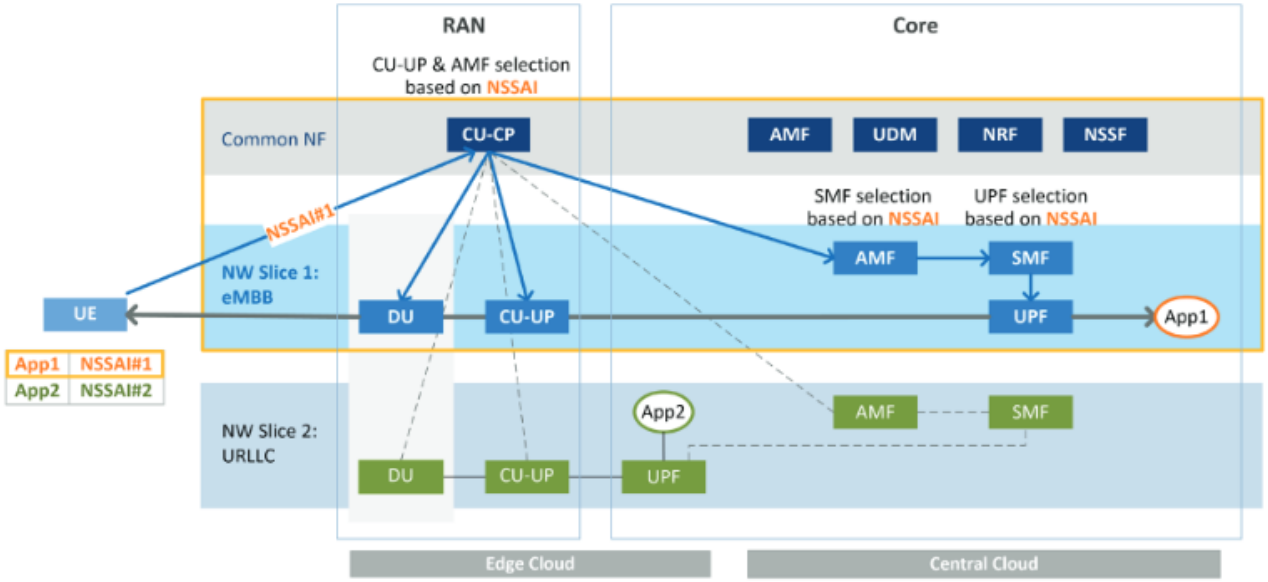

Figure 26: Slice creation across the network

The NSSAI is a collection of Single NSSAIs (S-NSSAI). An NSSAI may be a Configured NSSAI, a Requested NSSAI or an Allowed NSSAI. There can be at most eight S-NSSAIs in Allowed and Requested NSSAIs sent in signaling messages between the UE and the Network.

The S-NSSAI is divided into SST (Slice Service Type), which indicates the type of service or slice, and SD (Slice Differentiator), an optional parameter that indicates different slices of the same type, thus with the same SST.

In the technical specification TS 23.501, 3GPP provides a standardized classification that groups different services, such as eMBB, URLLC, Massive Internet of Things (MIoT), Vehicular-to-everything communications (V2X, where the X means vehicle, infrastructure, pedestrians, etc.), High-Performance Machine-Type Communications (HMTC) within five SST categories, as shown in the Table 7. This classification can be used as a baseline or template to implement network slices for most customer requirements. In addition, the TS 23.501 does not limit the creation of other categories if necessary, and the assigned SST values do not give priority to one category over the others.

Table 7: Network slice identifiers

Slice/Service type	SST value	Characteristics
eMBB	1	Slice suitable for the handling of 5G enhanced Mobile Broadband.
URLLC	2	Slice suitable for the handling of ultra- reliable low latency communications.
MIoT	3	Slice suitable for the handling of massive IoT.
V2X	4	Slice suitable for the handling of V2X services.
HMTC	5	Slice suitable for the handling of High-Performance Machine-Type Communications.
Reserved	6-127	
Operator-specific	128-255	

2.3.2 Mobile Edge Computing (MEC)

Figure 27 illustrates the MEC according to the ETSI specifications (Frangoudis, 2020).

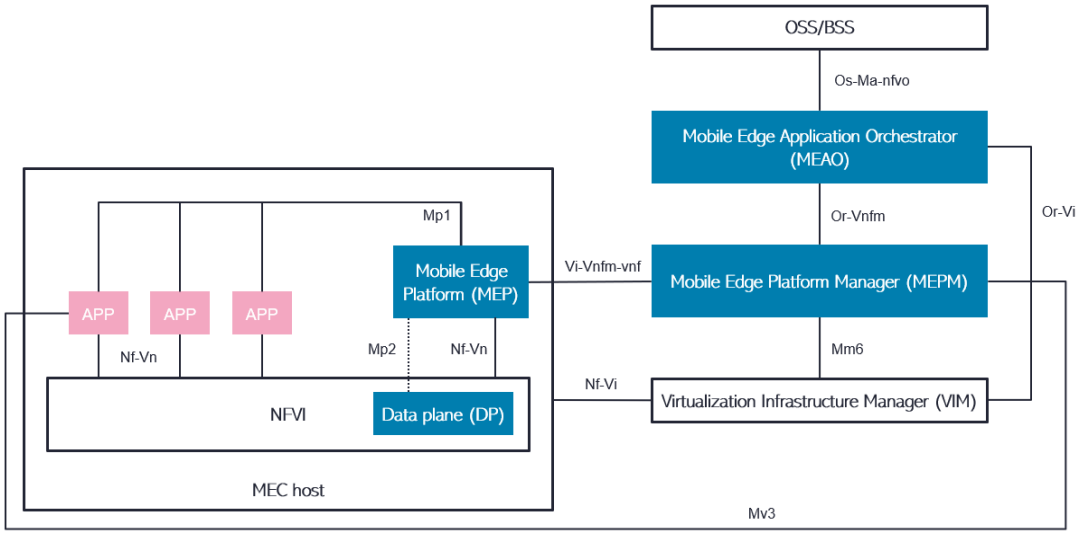

Figure 27: ETSI MEC

The mapping of the ETSI MEC to the OREOS platform is simple:

- MEP ≡ 3GPP's CN AF component;
- DP ≡ 3GPP's CN UPF component;
- MEAO ≡ ONAP's Network Function Virtualization Orchestrator (NFVO) component;
- MEPM ≡ ONAP's Generic Virtual Network Function Manager (G-VNFM) component;
- Mm6 interface ≡ ONAP's MultiCloud component.

In the 5G architecture, the Mobile Edge Platform (MEP) will be integrated as a 5G Application Function (AF), depending on the use-case. It may request traffic redirection for a MEC application as per the request of the Mobile Edge Application Orchestrator (MEAO) via the Mobile Edge Platform Manager (MEPM). Therefore, if MEP is a trusted 5G AF, it can use directly the PCF to generate a policy to offload traffic towards the MEC application. Otherwise, it uses the NEF to access the SMF via its traffic filter policy API and requests traffic redirection.

Integrating MEC data plane with the 5G system for routing traffic to the local data network and steering to an application is straightforward. The AF interacts with 5G control plane functions to influence traffic routing and steering. Figure 28 illustrates how MEC maps to the 5G system architecture, where the data plane defined in UPF elements is mapped to a MEC Platform (MEP). The MEP would perform traffic routing and steering function in the UPF. The PCF and the SMF can set the policy to influence the traffic routing in the UPF.

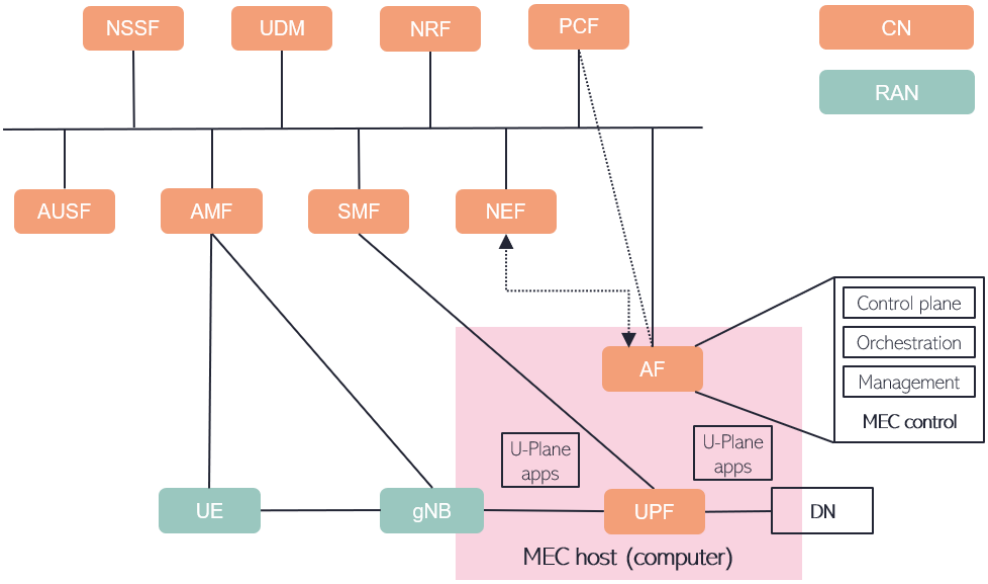

Figure 28: 3GPP MEC

2.3.3 ONAP

ONAP can run on Kubernetes, as illustrated in Figure 29.

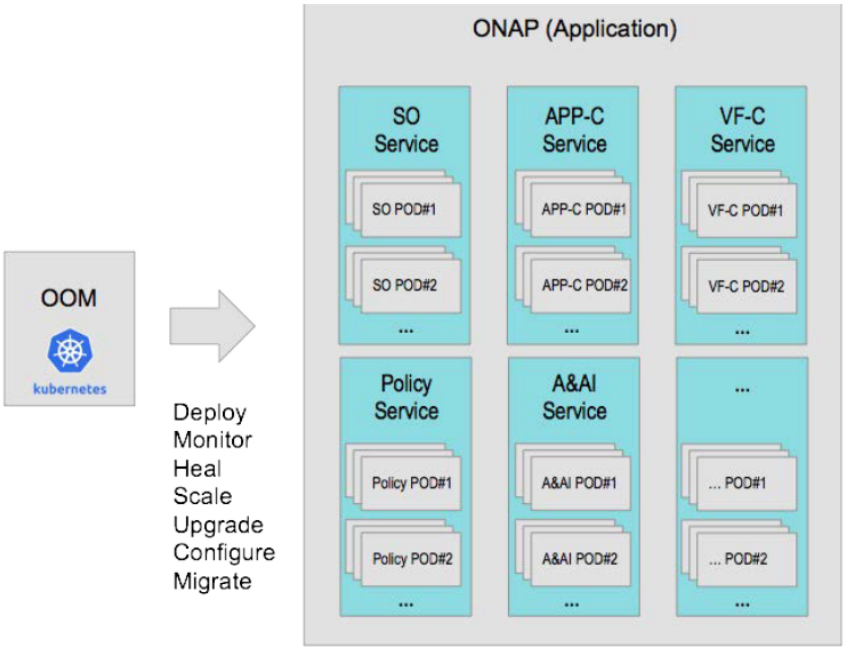

Figure 29: ONAP on K8s

The ONAP Operations Manager (OOM) is responsible for end-to-end lifecycle management and monitoring of ONAP. OOM does not provide support for containerization or orchestration of network services or VNFs that are managed by ONAP. Instead, OOM orchestrates the lifecycle of the ONAP platform components. OOM takes advantage of several Kubernetes innate features to provide ONAP with functionality such as:

- Initial deployment with in-built dependency rules;
- Unified configuration across all ONAP components;
- Ongoing real-time health monitoring of ONAP components;
- Healing, where failed ONAP components are restarted automatically;
- Scaling, where ONAP components can be clustered and seamlessly scaled;
- Upgrade with little to no impact on the overall ONAP availability.

Containers are inherently more compact than VMs. With OOM, ONAP can be deployed with a minimum resource requirement of 156-220GB memory and 54 vCPUs. In contrast, an equivalent VM-based deployment of ONAP requires 336GB memory and 148 vCPUs. For this reason, Kubernetes is a much more efficient method to deploy ONAP. Nonetheless, ONAP can be configured to run on Kubernetes on top of OpenStack, as illustrated in Figure 30.

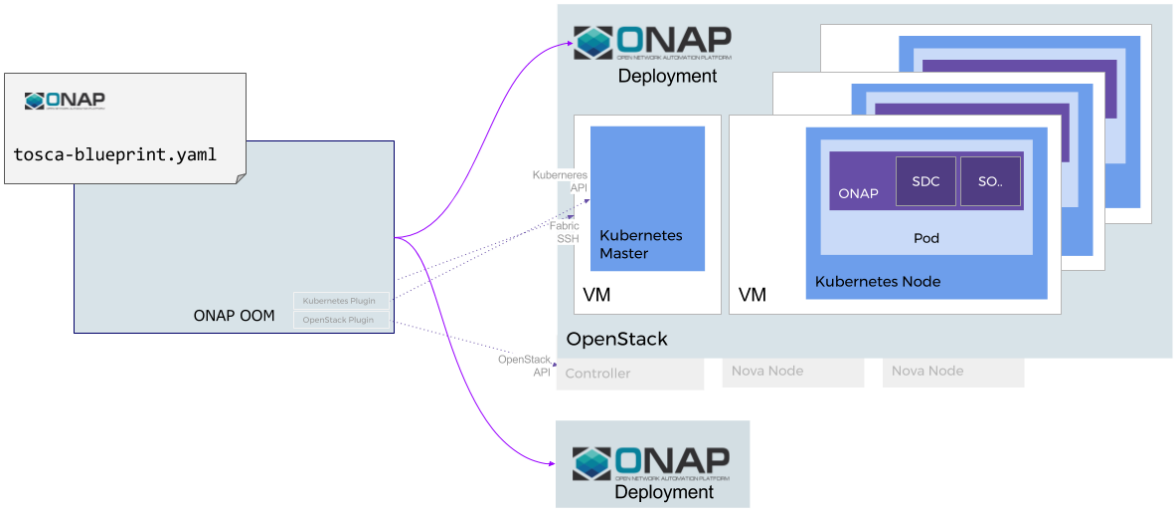

Figure 30: ONAP on K8s on top of OpenStack

Running Kubernetes on OpenStack allows combining the exposition of resources in OpenStack such as network, compute and storage, with in the consumption of these resources by Kubernetes.

2.4 Configuration set-up

This chapter focuses on exploring the configuration requirements and procedures to deploy all the components required by the OREOS platform, from infrastructure-level components, such as cloud and cluster technologies (e.g., OpenStack and Kubernetes), along with their required supporting services, to application level-components, such as ONAP and 5G components.

2.4.1 Topology

OREOs aims at providing a unified framework to tackle the challenges around E2E orchestration of mission critical network services (i.e., with minimal latency). The requirement demands for these services entails extremely efficient usage of the inherently distributed infrastructure, across IoT-Fog-Edge-Cloud, on top of which the services will run. This transports many of the optimization problems for these services and applications, to the realm of distributed system. For such distributed applications, Cloud Native practices and technologies have emerged as the *de facto* methodology to tackle the challenges they entail. Especially relevant for this section, **Kubernetes,** the Data Center-level container orchestration platform, natively provides many capabilities that are specifically targeted at highly available and distributed systems.

Although Kubernetes covers distributed applications, it does so at the Data Center level. If such applications are also distributed across multiple geographically dispersed Data Centers, the solution requires the use of Multi-Cluster Kubernetes topologies.

These reasons lead the OREOS project to settle as the on Kubernetes as its container-based application orchestrator, hosting most applications that are part of the OREOS platform, ranging from the 5G components to the MANO components. As such, a multi-cluster is also a requirement for the diverse set of applications workloads targeted by OREOS.

Figure 31 illustrates the topology solution of the OREOS platform, including infrastructure and application components.

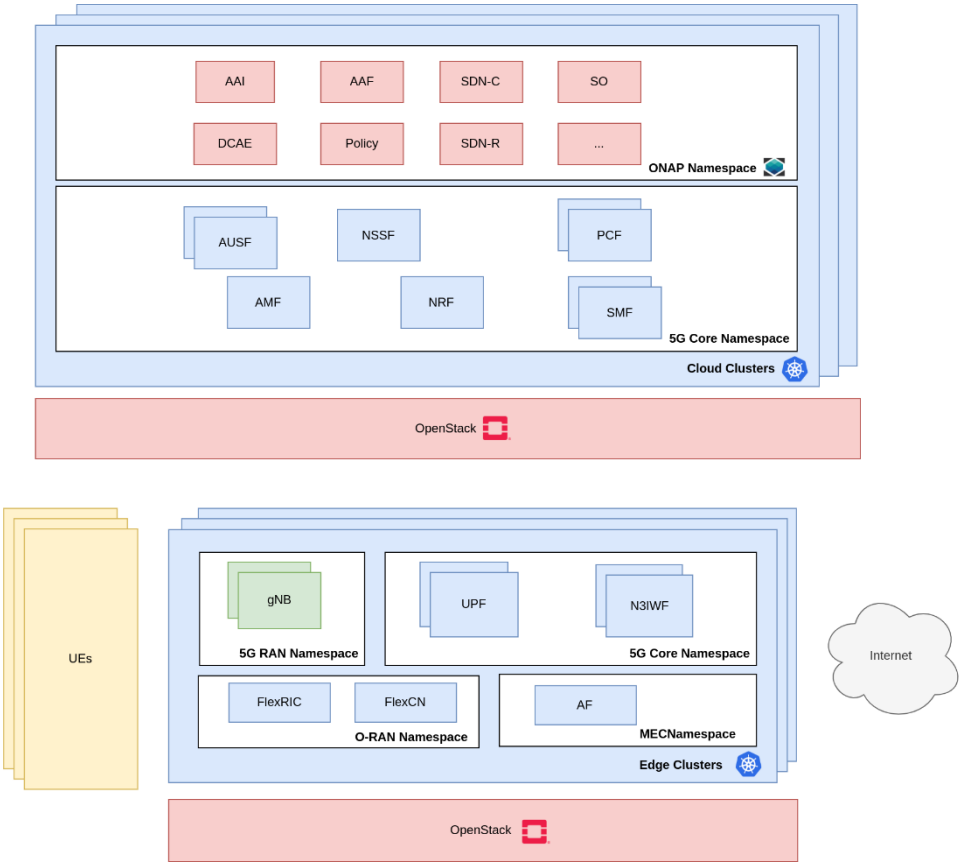

Figure 31: OREOS topology

In the following subsection, more information on the elements of both the infrastructure and the OREOS application described in Figure 31 is discussed, as well as the application configuration information that contributes to the topology.

2.4.2 Infrastructure Components and Tooling

This subsection presents the infrastructure components, including OpenStack and Kubernetes.

2.4.2.1 OpenStack

OpenStack is a robust and stable private cloud platform that provides a consistent and well-defined API abstraction layer to access and control data center resources, such as compute, storage and network (Silverman, 2018). In the OREOS project, the pool of distributed infrastructure resources available will primarily be orchestrated by OpenStack. This means that multi-cluster Kubernetes setup will be primarily built on top of Kubernetes. This infrastructure setup has many advantages, especially for telecom network applications, such as: allowing for mixed virtualized and containerized workloads, for both legacy and cloud native applications, allowing Kubernetes clusters to leverage the Neutron networking capabilities to connect multiple networks to the cluster workers for applications that require access to network multiple data planes, along with multiple performance improvements that OpenStack provides, in contrast with bare metal Kubernetes deployments.

2.4.2.2 Kubernetes

Kubernetes is widely adopted, and there are many options for the setup of a cluster. Popular opinions include both DYI and cloud solutions, ranging from *kubeadm* to managed clusters by cloud providers (Diouf, 2020). One popular distribution tool for kubernetes is Rancher's RKE, and it is capable of running entirely on top of Docker containers (Seittenranta, 2018).

Like most solutions, it is a tool aimed at solving the installation complexity of a cluster, especially as the number of nodes scale. In addition, this solution is able to automate and simplify the deployment to both bare-metal and virtualized servers, all the being independent of the operating system and the platform the nodes are running on.

RKE leverages a custom developed CLI tool to orchestrate the installation of the containers running Kubernetes. As dependencies to run it, on the machine running RKE's CLI, an SSH connection from must be established with the node machines of the cluster, and on the nodes, a supported version of the Docker runtime must be available. In addition, RKE's requirements, Kubernetes services are also required to communicate with each other across the multiple cluster nodes. Table 8 lists the ports required to be opened between nodes, along with their purposes in the Kubernetes architecture.

Table 8: K8S ports with Ranch RKE

Protocol	Port Range	Source	Purpose
TCP	443	Worker Nodes, API Requests, and End-Users	Kubernetes API server.
UDP	8285	Master & Worker Nodes	flannel overlay network - *udp backend*. This is the default network configuration (only required if using flannel)
UDP	8472	Master & Worker Nodes	flannel overlay network - *vxlan backend* (only required if using flannel)
TCP	2379-2380	Master Nodes	etcd server client API
TCP	10250	Master Nodes	Worker node Kubelet API for exec and logs.
TCP	10255	Heapster	Worker node read-only Kubelet API.
TCP	30000-32767	External Application Consumers	Default port range for external service ports. Typically, these ports would need to be exposed to external load-balancers, or other external consumers of the application itself.
TCP	ALL	Master & Worker Nodes	Intra-cluster communication (unnecessary if vxlan is used for networking)
UDP	8285	Master & Worker Nodes	flannel overlay network - *udp backend*. This is the default network configuration (only required if using flannel)
UDP	8472	Master & Worker Nodes	flannel overlay network - *vxlan backend* (only required if using flannel)
TCP	179	Worker Nodes	Calico BGP network (only required if the BGP backend is used)

RKE itself follows the Kubernetes best practices around declarative configuration, providing a single YAML based configuration file where cluster specifications can be expressed. The options that can be expressed in this specification allow the configuration of most aspects of a cluster, such as the nodes and Kubernetes version, but also the necessary configurations for RKE to connect and operate on the nodes.

2.4.2.3 Helm - Kubernetes Package Management

The Helm is the tool used in the management of Kubernetes package called charts. Helm is broadly divided into two categories namely Helm Client and Helm Library (Pervaiz, 2021). It is the tool used to simplify installing and managing Kubernetes application rendering the designed templates and communicate with the Kubernetes API. The Helm Client is a command-line client for end users, and it is responsible for local chart development, repositories management and managing the releases. Helm client also interfaces with the library ensuring

charts are installed and the upgrade installation and uninstallation requests of the new releases and existing charts respectively. The Helm library however encapsulates the Helm logic in order to be used by various clients. Applications developed to run on top of Kubernetes can quickly reach a significant level of complexity as more applications are added to deliver a service, and as these applications are interactively developed. Helm is package manager for Kubernetes applications, similar to what the tool *apt* deliver for Linux applications. Helm, apart from specifying the structure and syntax for constructing charts, also provides a client CLI tool that can interact directly with the target Kubernetes cluster to manage deployment, undeployment, upgrades and rollbacks for the applications in a chart.

This makes it simple to deploy services comprised of multiple Kubernetes resources with a single command (or equivalent operation). Again, Helm itself builds on top of the best practices around declarative configuration, and all charts are described in YAML manifests combined in a single package that can be advertised to your Kubernetes clusters. Some of the Helm Concepts include:

- Helm Chart which is the package with YAML written templates relating to the given application;
- Helm Repository is the folder where stable carts can be stored, collected, and shared;
- Helm Release is a chart instance patch released for running on the clusters.

2.4.2.4 Kubernetes Advanced Networking

Most Kubernetes distributions eg Rancher, Red Hat Openshift do not come with the required functionality to implement advanced network configurations, e.g., multiple network connections for pods, as typically, each pod only has one network interface (apart from a loopback). Although such capabilities have not yet been normalized in a Kubernetes release, they have been delivered in the form of plugins, which take advantage of the well-established Kubernetes extensibility functionalities, such as Custom Resource Definitions (CRDs). This is case of telco-like workloads, which have more specific requirements on networking configuration for most of the services, such as 5G. There are many third-party plugins that can be used in advanced networking system of Kubernetes which are mostly opened sourced, this includes Cisco Application Centric Infrastructure (ACI), AWS Virtual Private Cloud (VPC) networking for Kubernetes clusters, Azure CNI, Calico; Fannel, Google Compute Engine etc.

The ability to allow and configure a pod to connect to multiple networks has been most popularly provided by the Multus plugin (Qi, 2020). Multus is a Container Network Interface (CNI) plugin for Kubernetes that enables attaching multiple network interfaces to pods. Multus enables the creation of a a multi-homed pod that has multiple interfaces. This is accomplished by Multus acting as a "meta-plugin", a CNI plugin that can call multiple CNI plugins. Multus CNI also follows the Kubernetes Network CRD De-facto Standard to provide a standardized method by which to specify the configurations for additional network interfaces.

OpenVSwitch is an open-sourced rendering of virtual multilayer switch (Nascimento, 2011). Open vSwitch's major goal is to offer a switching stack for hardware virtualization settings while also supporting a variety of protocols and standards used in computer networks. Open vSwitch has been employed in variety of virtualization platforms, switching chipsets, and networking hardware accelerators because it can function as a software-based network switch operating within a virtual machine hypervisor, as well as the control stack for dedicated switching hardware. Without the usage of a kernel module, Open vSwitch may also run purely in userspace. It should be easier to port this userspace implementation than the kernel-based switch. In userspace, OVS may connect to Linux or Data Plane Development Kit (DPDK) devices.

2.4.2.5 Multi-access Edge Computing

The MEC Platform considered for the OREOS project, as stated above, is based on the Intel® Smart Edge Open platform. It is a cloud-native edge computing platform that allows applications to be deployed and managed in a variety of different edge locations, environments where resources constraints and high network performance must coexist. To accomplish this, it builds on Kubernetes and on the rich cloud-native ecosystem to create a set of experience kits, tailored for each type of edge location with necessary software and software configuration for optimally hosting applications.

To deploy and configure the platform, on any of the edge locations of the OREOS' network infrastructure, a set of configuration scripts, based on the Ansible configuration management platform (Hochstein, 2017), are made available by the project. These allow both the base installation of the platform and the additional features provided by the experience kits to be easily and automatically be deployed and configured. These scripts provide a set of configuration parameters that can be customized to the project’s topology needs, providing the ability to personalize which components are installed in which edge locations, based on requirements, hardware, and other topology relevant considerations. Details for the use of these procedures is made available in the project's documentation.

For the OREOS project, the features provided by the 5G Private Wireless with Integrated RAN, the Access Edge, and Near Edge Experience Kits exposed in the documentation, provide the necessary building blocks for the project. These include components responsible for advanced hardware and network integration, but also 5G capable services, such as Application Function (AF). Although the project is in part open source, most of the features required by OREOS, are not available in the open components. The remaining necessary components, as can be seen in the figure below, are available under a license, which can be requested from Intel. At the time

of writing of this document, this license has been requested and is being evaluated. We are expecting this to be made available in time and without causing any inconvenience in the work plan, as depicted in Figure 32.

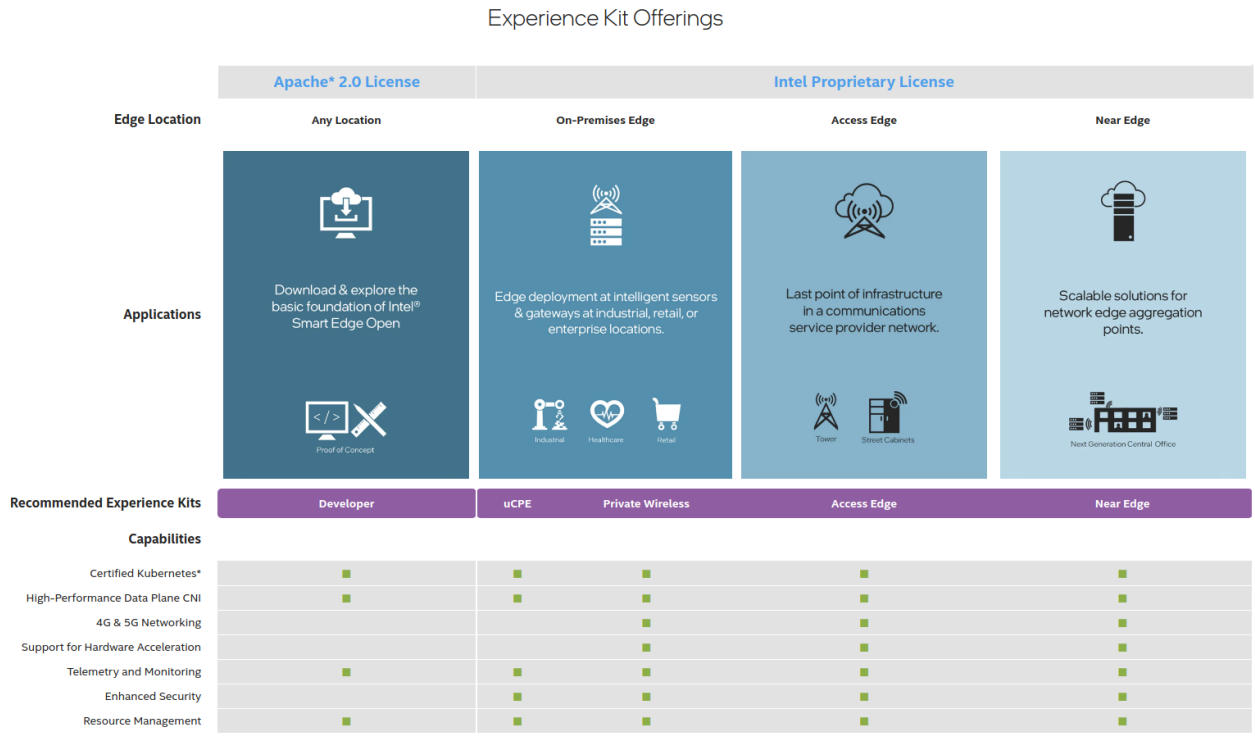

Figure 32: Experience Kit in OREOS

2.4.3 OREOS Applications

This section describes the configuration and provisioning processes for the main applications in the OREOS platform.

2.4.3.1 ONAP

ONAP, as had been designed and directed at Cloud Native infrastructure and practices. Thus, all its components and services run, preferably, on Kubernetes – although there was support for both containerized and virtualized environments as the project first started, it quickly started to be more opinionated towards Kubernetes-based deployments. Given the complexity of the platform, the ONAP Operations Manager (OOM) was developed to accommodate ONAP-specific logic for end-to-end lifecycle management and monitoring the platform.

OOM does not provide support for containerization or orchestration of network services or VNFs that are managed by ONAP. Instead OOM orchestrates the lifecycle of the ONAP platform components. OOM takes advantage of several Kubernetes innate features to provide ONAP with functionality such as:

- Initial **deployment** with in-built dependency rules;

- Unified **configuration** across all ONAP components;
- Ongoing real-time **health monitoring** of ONAP components;
- **Healing**, where failed ONAP components are restarted automatically;
- **Scaling**, where ONAP components can be cluster;
- Cluster ONAP services to enable seamless **scaling;**
- **Upgrade** with little to no impact on the overall ONAP availability.

In practice, OOM's functionalities are mostly delivered by Helm's (with additional plugins for advanced deployment) and Kubernetes' native capabilities to provide that additional lifecycle logic. As far as resource requirements are concerned, as of the Honolulu release, the entire ONAP can be deployed with 224GB of Disk, 160GB of RAM and 112 vCPUs.

2.4.3.2 5G Core

As of the writing of this document, many of the available 5G Core implementations have already been developed targeting Cloud Native and containerized environments, specifically Kubernetes. To this end, most 5G network functions have been built on container images and entire 5G Core implementations have been described and configured in Helm charts (e.g. see also section 2.1.2).

Although they have been targeted at these environments, 5G workloads, more significantly the User Plane functions (UPF), are dependent on more advanced networking options, such as the ability for the actual hardware network interfaces that transport the user traffic to be plugged into a pod running a UPF, AMF or SMF. These connections correspond to the interfaces to the N2, N3, N4 and N6 subnets required for that type of traffic. As specified before, configuring the hosts of the clusters with the Multus or SR-IOV network plugins. With the required networking configurations applied to the multi-cluster environment of the OREOS platform, 5G can be deployed in similar topologies to the one presented in Figure 33. The actual location of the 5G components in the infrastructure will vary significantly with the network slicing configurations of a particular deployment, as such, this image represents only one of these deployment possibilities. This is the case for Open Air Interface's 5G Core, which can also be deployed and configured with Helm charts provided by the project.

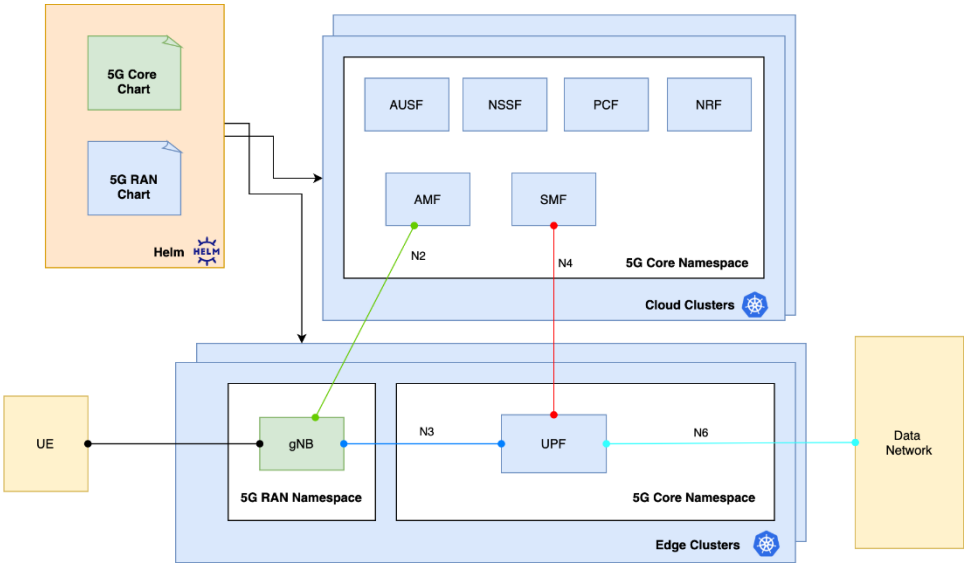

Figure 33: 5G Core deployment with FlexCN

2.4.3.3 5G Radio

Aiming at a fully capable 5G network environment, the OREOS project also deploys the components necessary for a functional 5G RAN. Here, again, most of the implementations explored in technology assessment carried out, also target cloud-native deployment environments. This also signifies that the whole software stack of the RAN has also been described and configured using Helm charts. OAI's implementation, follows this and provided a Helm chart describing the deployment and configuration of not only the RAN components, but also UE simulators. The latter component in particular will be explored extensively, in combination of an additional geolocation-based channel simulator in many of the project's use cases.

Although this greatly simplifies the deployment stage, the targeted infrastructure for their network applications must also be properly setup with the required hardware integration and cluster configurations. To this end, Intel Smart Edge Open, is the component responsible for providing the cluster and configuration management required for hosting RAN components in the edge locations.

This project, in most of its use cases, will also rely on a 5G RAN controllers capable of providing a production-like and ONAP-driven configuration environment for its network services, targeting dynamic Network Slicing environments. For this purpose, the OAI project and ONAP project also provide implementations of the O-RAN specified Near-RT RIC and Non-RT RIC components, respectively the FlexRIC and SDN-R services. As far as deployement and configuration goes, SDN-R is deployed along with ONAP, and FlexRIC is deployed along with the RAN components using the same method.

3. Use case description

This section describes two major use cases to validate the OREOS platform:

- Use Case 1 - Smart City;
- Use Case 2 - Autonomous Driving.

The E3.1 deliverable (OREOS, Deliverable E3.1, 2021) documented several use cases, such as: Pedestrian safety within vehicle mobility; End-to-end network slicing; 3GPP & O-RAN alignment; 5G SONs; and Intent Based Networking. These have been merged into the two major use cases.

3.1 Smart City

This section presents the Smart City use case and three user stories that include pedestrian safety, air quality and crime prevention.

3.1.1 Description

The environment in which this use case runs in is of urban nature, i.e., city context, which aims to enhance the safety of people in their daily life, enhance the quality of life with environmental monitoring and enhance the efficiency of actions against crime. Such user stories are inspired in the documentation available by United 4 Smart Sustainable Cities (U4SSC) initiative further detailed in Appendix – Smart cities initiatives and related standardization activities, and Appendix – European Innovation Partnership on Smart Cities & Communities (EIP-SCC) (U4SSC, 2021). The safety perspective considers pedestrians crossing the road, where it is necessary to inform approaching vehicles to reduce the velocity and stop the vehicle before reaching pedestrians.

The detection of pedestrians is performed through video cameras, employed for city surveillance, which through intelligent mechanisms can detect the movement of citizens, namely persons walking and pedestrians crossing streets. With the network information regarding connected, nearby or approaching vehicles, there is the possibility to provide in advance the information of the pedestrian's presence on the roads. This concept is illustrated in Figure 34 (Linget, 2020), whereas a traditional vehicle needs an unobstructed view to proceed safely, by means of telecommunications technology, this view can be provided by a camera or other vehicles with a better view.

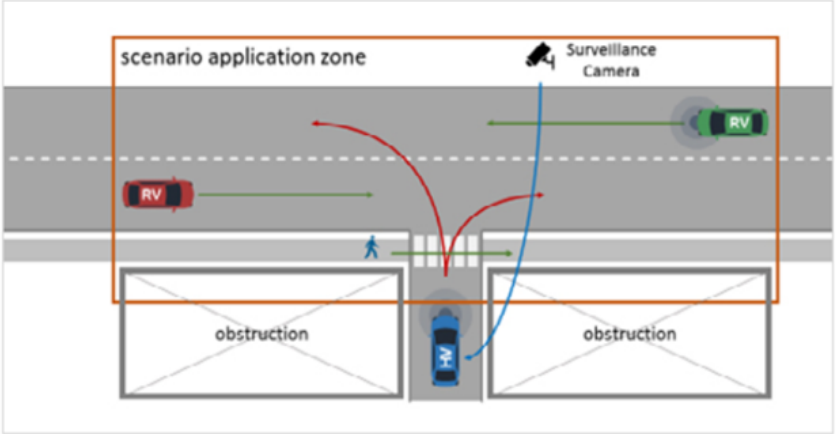

Figure 34: Pedestrian Crossing

The pedestrian safety use case occurs in the communication infrastructure of a smart city, where other services are also provided. For instance, content delivery networks or Unmanned Aerial Vehicles (UAV) to aid in the surveillance of special events like concerts, public manifestations (Naqqash Dilshad, 2020), among others. The communication infrastructure also supports the communication of vehicles, regarding the exchange of messages with safety purpose. Given the diverse applications/services that run in the communication infrastructure, the support of network slicing is required to isolate the application/service flows and to guarantee QoS settings.

The Air quality management user story occurs in a setup based on multiple vendors which require an infrastructure to connect sensors and other devices to gather data, such as weather or emission rates, in order to determine accurately the air pollution, as well as to enable predictive weather forecasting. The collected data needs to be analysed in quasi real-time and to multiple clouds, since vendors provide analytics services in different platforms, such as Azure, Amazon or Google Cloud, for real-time decision and response to incidents. The collected data is analysed and made available to citizens in the city for awareness regarding any possible pollution incident (i.e., industry or other sources making more pollution than usual) or even to impact the paths followed by users, to avoid roads/streets with high pollution.

The crime prediction is useful for agile policing and to enhance the safety of people visiting a given city. Not all neighbourhoods provide the same levels of safety, and some may have high rates of crime (e.g., robbery, and even murders) thus, there is the need to inform citizens or visitors to avoid such places for their own safety. Such information can be widespread using applications, which leverage from ML models that are able to identify malicious actions (e.g., robbery) within reports from platforms used by the police force, generating the risk information of the distinct neighbourhoods.

3.1.2 Objectives

The objectives of the Smart City use case are described per user story (US):

1. US#1 Pedestrian safety
   1.1. Enable the detection of pedestrians crossing the streets;
   1.2. Support the analysis of video in (quasi) real-time;
   1.3. Support sharing of information in real-time with vehicles;
   1.4. Enable feedback to users crossing the street regarding vehicles that are approaching.

2. US#2 Air quality monitoring
   2.1. Enable the collection of data from several sensors, IoT devices;
   2.2. Support the aggregation of data for analysis at the edge;
   2.3. Enable feedback regarding measured data and predictions regarding air quality;
   2.4. Support the integration of pollution information with other services (navigation systems).

3. US#3 Crime prevention
   3.1. Enable the detection of robbery (via video analysis or other sources of information);
   3.2. Support prediction of crime considering historical data of crimes in a certain area;
   3.3. Inform users regarding risk through applications specific to the city;
   3.4. Support the collection of data from citizens/tourists regarding their perception of safety in each location. For instance, to report robberies, or other incidents where video surveillance is not present.

3.1.3 Workflow

Figure 35 describes a generic workflow for the user stories being a reference scenario for the Smart City use case, with UEs taking advantage of the distributed processing offered by MEC under the orchestration of the OREOS solution. The diverse applications in the UEs connect to the respective AF through a network slice.

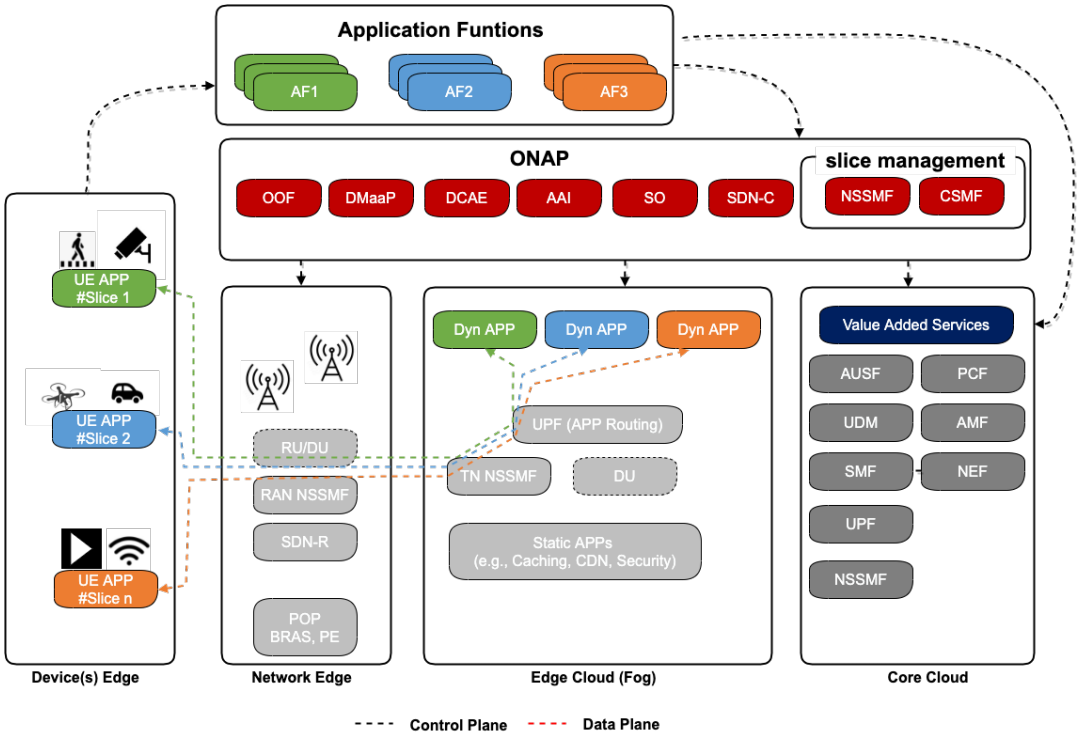

Figure 35: Mobility workflow using ONAP in Edge networks

The general workflow depicted in Figure 35 includes the main components of the OREOS architecture and the place where the several functions are executed. The edge devices include the end-user devices with the diverse applications employed in the distinct use cases or with IoT devices collecting data, which can be pre-processed at the edge or at the core. A control plane is deployed to manage and orchestrate the distinct components, the network slices and the services of the 5G core.

3.1.4 Actors

The main actors are common to all the user stories (#US01, #US02, #US03) and include:

- OSS - full stack of systems which include systems realizing the Orchestration, Monitoring, Analytics, Policy and Inventory operations on network operator's ecosystem;
- RAN Intelligent Controller (RIC) - systems that are responsible for the management of radio resources in 5G Networks;
- OTT service providers – to deploy application functions in the infrastructure, for the #US03 with services for tourism and sharing of data regarding the perception of safety in the different streets of a city;
- End-users who access services, either through subscription models, or in the model of free services, as provided by municipalities (e.g., information of city events).

3.1.5 Assumptions

The following generic assumptions are formulated:

- The network infrastructure and respective services are deployed and fully operational (see Section 2);

- Network slices are configured by network providers as per SLAs established with municipalities;
- Service providers configure their services as per SLAs established with municipalities;
- All the radio equipment has been configured properly to connect to the network;
- Devices are deployed (video cameras) by service providers;
- The multiple clouds – "multiclouds" are properly configured and can act as the Edge of a network;
- Each use case is assigned to a specific slice, given the use case purpose (see Table 9).

The use cases in the Smart City scenario assume that the network is deployed with support for network slicing. It is assumed that the use case has at least three network slices configured, as detailed in Table 9 (base slices as in Table 7).

Table 9: Slice configuration in the Smart City use cases

Slide Id	Slice Description / User story	Slice / Service type
Slice 1	*#US 01 – Pedestrian safety* City Emergency Services with reliability and latency requirements	URLLC (SST=2)
Slice 2	*#US 02 – Air quality monitoring* The monitoring of the air quality, of the levels of pollution is performed through the collection of a high number of devices - Massive IoT. M2M is relevant in this context for scalability purposes and efficient data collection.	MIoT (SST = 3)
Slice 3	*#US 03 – Crime prevention* Access to content regarding safety, tourism information, as well as image analysis (or other collection mechanism) to gather accurate information to identify crime on a given neighbourhood/street is correlated with feedback provided by users, regarding their perception of safety.	eMBB (SST =1)

3.1.6 Trigger

Each user story in the Smart City scenario is triggered through specific actions, as described next:

- **US#1 Pedestrian safety** is triggered by the event of a citizen crossing a road, in a local covered by video surveillance cameras. The initial step is the capture of video and transfer of such information for analysis;
- **US#2 Air quality monitoring** is triggered with the deployment of IoT devices in several city locations and the necessary configuration of the data pipeline for collection, aggregation and processing;

- **US#3 Crime prevention** is triggered by the detection of an abnormal event regarding a citizen covered by the video surveillance. Such abnormal event (such as a fall or robbery) is detected by video cameras. The user story can also be triggered by the sharing of information from a tourist/citizen regarding the perception of safety in a certain place. Such feedback is performed through the "Citizen App" made available by the municipality.

3.1.7 Step-by-Step description

A step-by-step description of the actions carried out by the modules in ONAP, within the OREOS platform, is presented in this section.

3.1.7.1 Pedestrian Safety

The sequence diagram depicted in Figure 36 shows the interaction of the modules in ONAP, the OREOS platform, and the O-RAN environment involved in the pedestrian safety functionality of this use case. The two conceptual levels described in Figure 36 are:

1. **Object and event-based analysis:** This level implements pedestrian motion path detection using video data. An additional evaluation regarding the network performance and resource utilisation helps to perform optimisations to determine new network configurations, thus granting adaptability;
2. **Processing:** Covers the data collection (via surveillance cameras) as well as the notification to approaching vehicles using an RSU. This way, the vehicles can reduce their velocity (even to a standstill) and thus prevent accidents.

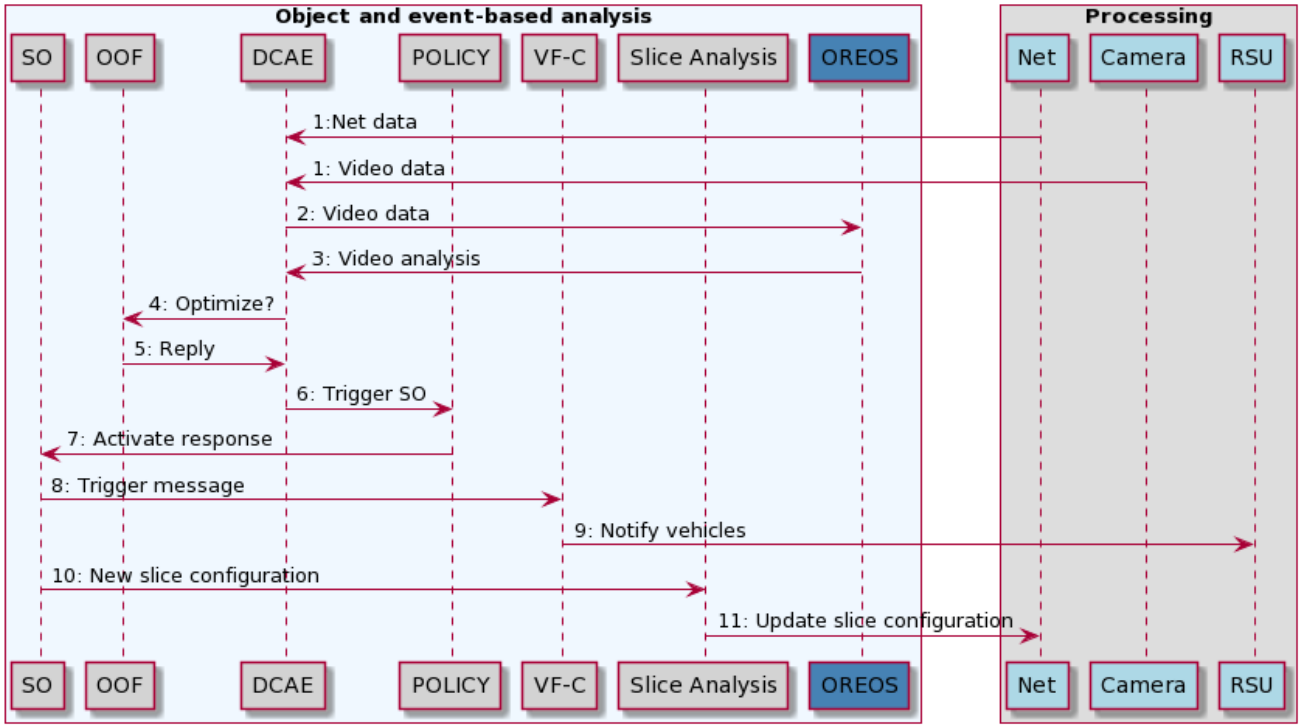

Figure 36: Modules interaction for the Pedestrian Safety functionality

The interaction between the modules in the different levels is described below:

1. Video data from surveillance cameras and data from the state of the network is collected by DCAE;
2. Video data collected by DCAE is sent to an OREOS AI/ML module for analysis;
3. OREOS AI/ML module identifies any pedestrian (if any) and sends analysis result to DCAE (if there is a pedestrian, an alert should be sent to the vehicles via RSU);
4. DCAE sends net statistics to OOF asking about the need to optimize the configuration of the network, for instance to determine if more resources are needed for a given network cell (e.g., high volumes of data);
5. OOF analyses the data and replies to DCAE;
6. DCAE triggers SO via Policy;
7. Policy requests the activation of the proper response via the SO;
8. SO requests the activation of the RSU-VF via the VF controller (VF-C);
9. VF-C activates the service to message the RSU, which will send the proper message to approaching vehicles;
10. According to the network utilisation, SO asks for a new configuration to the slice manager;
11. Slicing manager updates the slices configuration.

On the ***Processing*** level, the Camera and RSU relates to *User Equipment (UE)*, while the Net component relates to *FlexRIC*. The data generation is done using *OpenAirInterface*.

3.1.7.2 Air quality management

The Air quality management use case considers the employment of multiple clouds for the placement of service functions and for data analysis, as depicted in Figure 37. Two conceptual levels are considered:

1. **Placement and data analysis:** This level implements placement of VNFs in the MultiCloud setup (which can be an Edge in the OREOS infrastructure) and the data analysis. An additional evaluation regarding the network performance and resource utilisation helps to perform optimisations to determine new network configurations, thus granting adaptability;
2. **Processing:** Covers the data collection (via IoT devices).

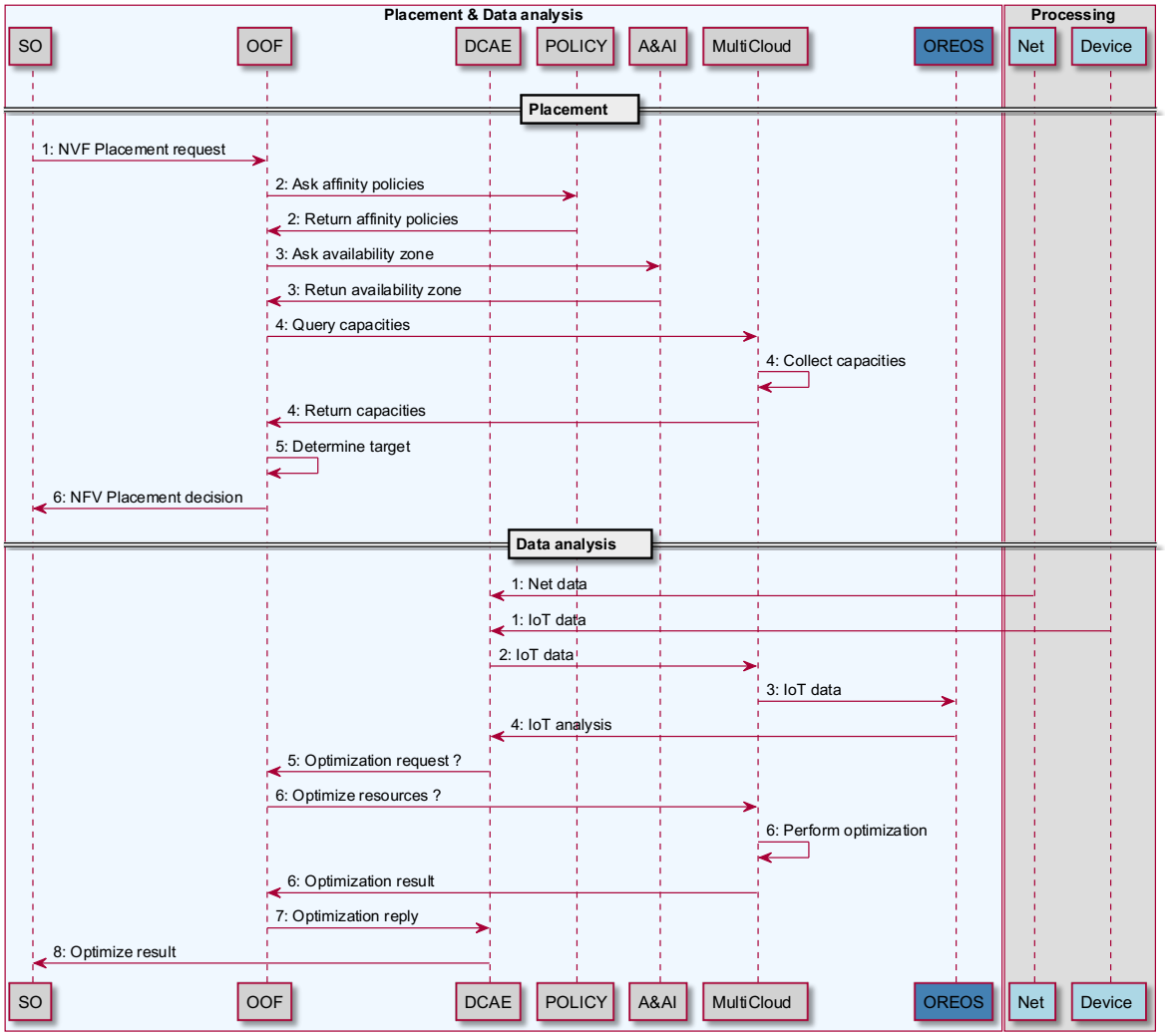

Figure 37: Modules interaction for the Air quality/pollution monitoring

The interactions between the different models include the placement, which can be performed at design phase, but also at the production phase:

1. Request to Place a VNF for data analysis, collected by the SO;
2. The OOF processes the request and checks the affinity policies, the Policy component returns the required information;
3. The OOF requests an update regarding the inventory to the A&AI component, which replies with the inventory information (how many edges or multiclouds are configured);

4. The OOF triggers the capabilities discovery in each of the multiclouds configured. The multiclouds component collects the request and the required information from the underlying infrastructure (i.e., Azure, Google cloud);
5. The OOF performs a decision regarding the most suitable target regarding the VNF for data analysis;
6. The SO is informed regarding the placement decision.

The data analysis component also includes the following interactions:

1. Devices send data, after the network access configuration;
2. DCAE collects IoT data from devices and data from the state of the network;
3. The collected data is sent to the multicloud to request analysis of results;
4. The OREOS AI/ML modules receive the results of the analysis configured at the multiclouds and perform enhanced analysis to determine the accurate pollution levels;
5. Given the time to achieve the results exceeding preconfigured thresholds, DCAE decides to optimize resources in the multiclouds, for instance to scale the VNFs performing analysis;
6. The OOF is instructed to proceed with the optimization request, which informs the multicloud component to perform optimization. The optimization is carried out and the OOF is informed regarding such the results of such operation;
7. The DCAE component that has triggered the decision is informed about the optimization operation;
8. The SO receives the update of the operation.

3.1.7.3 Crime Prediction

Figure 38 demonstrates the possible relations between the ONAP modules, the OREOS platform, and the O-RAN environment involved in crime detection user story, within two conceptual levels:

1. **Processing –** The background modeling, objects and pedestrian detection is implemented through video covering. After the Object and event-based analysis, if a crime is detected, the users and the competent authorities can be notified about the historical data of crimes or an emergency. Moreover, a 3-D scene reconstruction can be conducted to visualize the scene for further analysis. This level could be simple for home/public use (e.g., to consult historical data of crimes) or advanced for professional service (e.g., crime analysis in 3-D);
2. **Object and event-based analysis –** Performs trajectory estimation and individual posture classification. Therefore, it is possible to track each moving trajectory and model each interaction relationship to infer a multiple-person event. Given the results of this analysis, it is possible to take advantage of the

historical data to classify certain areas concerning the security level or call the authorities if there is some emergency.

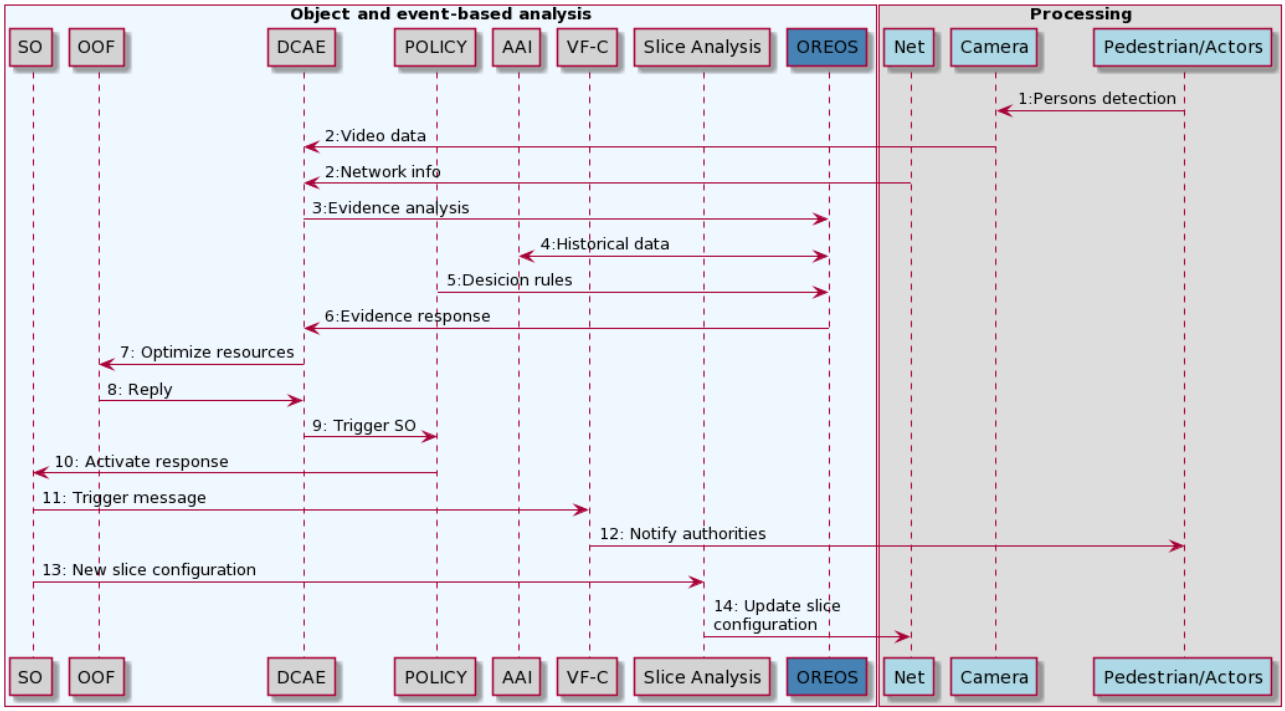

Figure 38. Modules interaction for the Crime Prediction functionality

Furthermore, to achieve the objectives proposed in the two conceptual levels presented, we can classify each model interaction as follows:

1. Background modelling, objects and pedestrian detection by cameras;
2. DCAE collects video data from the cameras and data from the state of the network;
3. The collected video is sent to OREOS AI/ML modules to identify whether there is an emergency. In multi-person events, the event analysis is achieved by understanding the interactions between the involved people;
4. To achieve a better analysis, OREOS AI/ML modules request the historical data about the local to the AAI module. Thus, it is possible to predict trends considering the history of the place;
5. The policy module is the decision-maker component, that can be used in conjunction with other OREOS AI/ML modules to enforce policies;
6. OREOS sends analysis results to DCAE (if there is an emergency, an alert should be sent to authorities);
7. DCAE provides net statistics to OOF to assess if more resources are required for a specific network cell;
8. OOF examines the data and responds to DCAE;
9. DCAE initiates SO through Policy;
10. Policy requests an appropriate reaction to be activated via the SO;

11. Request that the Cameras-VF be activated through the VF controller (VF-C);
12. The VF-C initiates the service;
13. Based on network utilization, SO requests a fresh configuration from the slice manager;
14. The Slicing Manager updates the slice configuration.

3.1.8 Final considerations

During the journey, pedestrians are identified on the crossroads and such identification is provided to nearby vehicles. For the pedestrian safety functionality, the pedestrians are identified on the crossroads and an alarm/information is sent to approaching vehicles via an RSU. Additionally, the configuration of the network slices is adaptive to changing network traffic conditions.

The Air pollution management use case can analyse data from IoT devices and is able to optimize the resources in the network, by placing VNF functions close to the user or in different cloud infrastructures.

The crime prediction use case can detect robbery events through image processing, providing the means to update information of crime in real-time.

3.1.9 Requirements

The requirements of the Smart City use case are described considering functional and non-functional aspects of the several user stories.

3.1.9.1 Functional

The functional requirements are documented in Table 10, a more detailed list is documented in OREOS E3.1 (OREOS, Deliverable E3.1, 2021).

Table 10: Functional requirements of Smart City scenario

Requirement ID	Requirement Description	User Story
#FReq-A01	Network slicing	ALL
#FReq-A02	Video sharing support	ALL
#FReq-A03	V2X communications	#UC01
#FReq-A04	M2M communications	#UC02

3.1.9.2 Non-Functional

The non-functional requirements are documented in Table 11.

Table 11: Non-Functional requirements of Smart City scenario

Requirement ID	Requirement Description	User Story
#NFReq-A01	Dynamic provision of resources (e.g. support scaling and placement of functions)	ALL
#NFReq-A02	Authentication of users	ALL

3.2 Autonomous Driving

Autonomous vehicles are extremely dependent on URLLC in 5G vehicular networks due to possible conflicts in reliability and latency performance. It's predicted that during this decade vehicular networks will represent one of the main transmitters/receivers of 5G vehicular networks.

3.2.1 Description

This use-case will focus on a particular aspect that has the purpose of optimizing autonomous vehicles communication processes, namely a *follow-me* functionality for safety-critical services: due to the stringent low latency requirements of autonomous driving, provisioning of applications must be placed as close as possible to the vehicle, which means that the services are required to migrate in unison with the vehicle trajectory.

The basic idea behind the follow-me cloud concept is that services, provided by a cloud, are following users throughout their journey. As soon as a user moves and thereby changes his attachment point to the network, the optimal data centre for providing the services being received by the user is determined. If the optimal data centre is different from the currently used one, a decision is made for or against moving the service to the optimal location. As a result, the services follow the user throughout his movements (T. Taleb, 2013).

Current Operation Support Systems (OSS's) in network operator's ecosystem were not designed to fulfil these types of requirements, since typically services are deployed once using static rules. Addressing these scenarios require dynamic management capabilities in OSS's to be able to respond to service context's changes in real-time. For this use case will be necessary the acquisition of live location information of the vehicle and of the closest Edge Data Centres.

3.2.2 Objectives

The goal targeted in this use-case is the use of *follow-me* application for safety-critical services. These functionalities are not available in today's network operators' ecosystems, which means next-generation OSS's will need to evolve and provide this in order to support operational challenging scenarios.

3.2.3 Workflow

The workflow of the Autonomous Driving use case is illustrated in Figure 39.

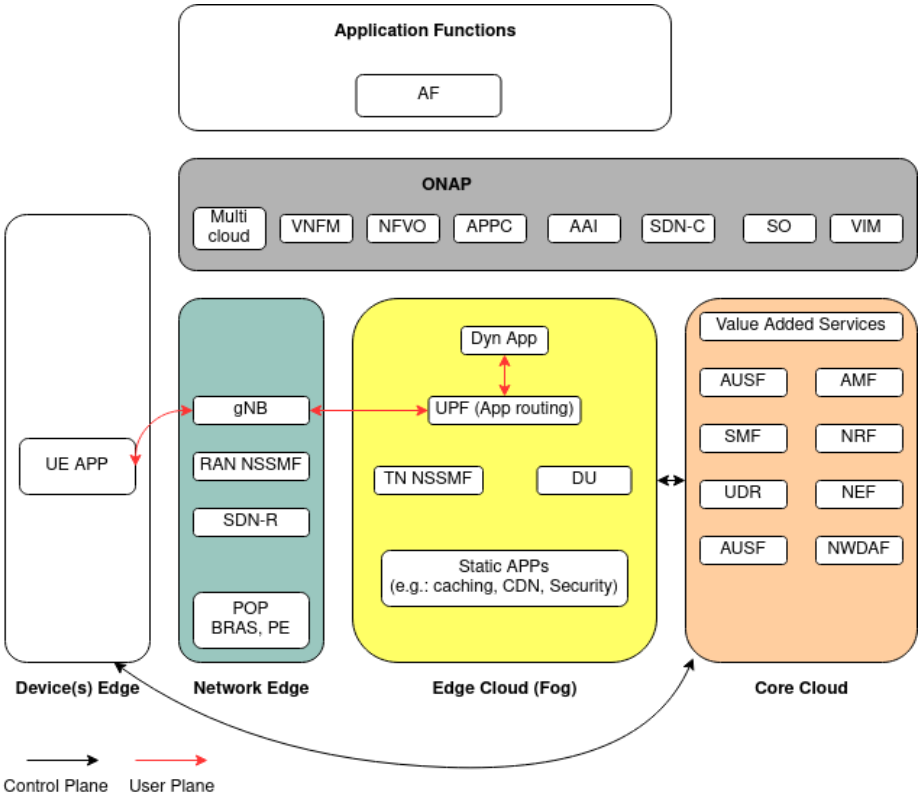

Figure 39: Workflow for Autonomous Driving use-case

In the current state-of-the-art, ONAP and 3GPP's 5GC deal with different domains of the mobile network:

- 3GPP is in charge of handling the mobility and user session management of the UE, which includes knowing the location of the UE, the services it is using, the required QoS, etc;
- ONAP on the other hand, is in charge of handling the cloud orchestration of the network nodes, by managing the onboarding and Life Cycle Management (LCM) of the network nodes (e.g., Virtual Network Functions (VNFs)) and Virtual Machines (VMs) or containers.

But in the end, both ONAP and 5GC have an objective to assist on network automation by continuously monitoring system data and perform anal, i.e., to continuously receive events and current network status data, perform analytics on that said data in order to extract patterns, so that feedback can be fed into the network for it to adapt to changing or predicted conditions. In this use case, we aim at removing the existent separation between the ONAP and 3GPP, in what concerns analytics for network automation. This will be done by ensuring that data from both entities can be correlated and that we can create a cause-and-effect situation.

From the 3GPP part, the NWDAF collects data from the NFs to extract traffic patterns and other data, such as UE location and used applications, QoS requirements, what data networks they are connected to, etc. Then a

closed loop occurs: NWDAF requests data from ONAP's DCAE and sends requests ONAP's SO and controllers to perform actions (such as UPF selection for traffic routing and AS onboarding from the ONAP). The figures of the subsection *Step-by-step (section 3.2.7) description* will detail this concept.

3.2.4 Actors

The main actors in this use-case are:

- OSS - full stack of operational systems which includes systems realizing the Orchestration, Monitoring, Analytics, Policy and Inventory operations on network operators ecosystem;
- Analytics – 5G Core Network with analytics functionality via NWDAF;
- 5G Core network functions for CP and UP;
- RAN CU/DU;
- RAN Intelligent Controller (RIC) - system(s) that are responsible for the management of radio resources in 5G Networks;
- Users in Vehicles (On Board Units – OBUs) with services and applications.

3.2.5 Assumptions

The vehicle must be able to change UPF and Data Network (DN) while being on the move and with no interruption in the service. When the UE moves from location 1 to location 2, the IP address of the server may change. With current networking solutions, an IP session between two peers will simply be torn down if the IP address of any of the two peers changes during the course of the session. The usage of usage of SDN technologies and OpenFlow provide scalability to avoid dealing with this issue (Ksentini, 2013), (D. Lake, 2021). It will be assumed that such technologies are being used and thus, the handover between DNs and UPFs does not cause service disruption.

Also, to guarantee service continuity at app level, it will be assumed that the applications that run on the local DNs are of type stateful, further detaila are provided in

3.2.6 Trigger

When the vehicle starts a new travel itinerary, while being currently connected to gNB-1, AF-1 and UPF-1, it performs a handover to gNB-2.

3.2.7 Step-by-step description

Important note: the NWDAF periodically receives:

a. UE ID from the AMF;

b. gNB ID from the AMF;

c. DNAI[5] list from the SMF.

And the application instance must be instantiated in a DC a-priori, with connectivity to the 5GC:

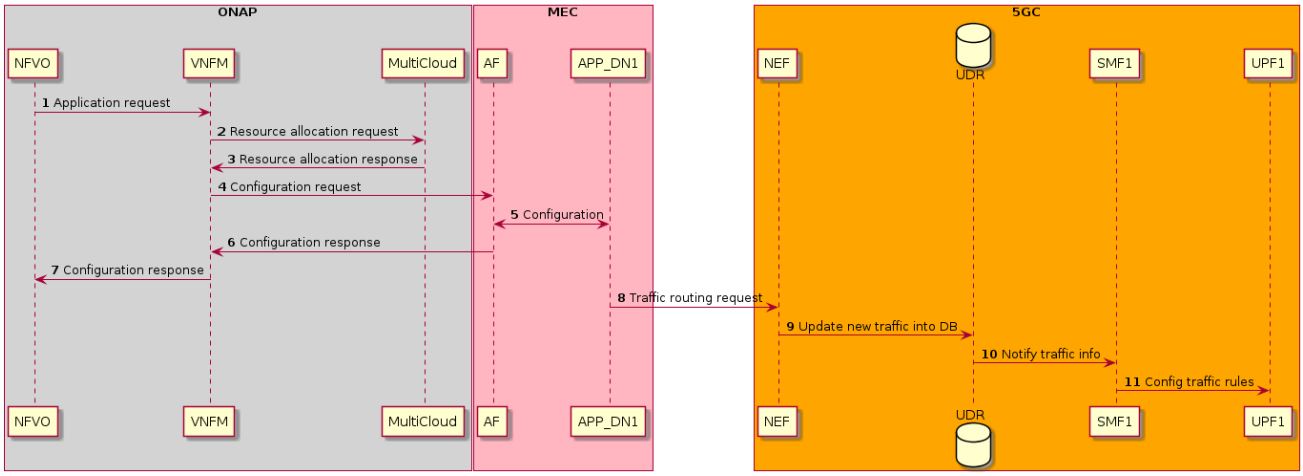

Figure 40: MEC application instantitation

This phase includes the following time-steps (inspired from (Sabella, 2021)):

3.2.7.1 t_0

The UE attaches to gNB #1 and the MEC app, which was a-priori instantiated at local DN #1, begins to send traffic to the UE. User plane traffic goes through UPF #1 and control plane through AF #1 (belonging to DC #1), as illustrated in Figure 41.

[5] The NDAI is an identifier of a user plane access to one or more Data Networks (DN) where applications are deployed, to activate traffic routing (the SMF activates traffic routing to the local DN by setting the requested DNAI).

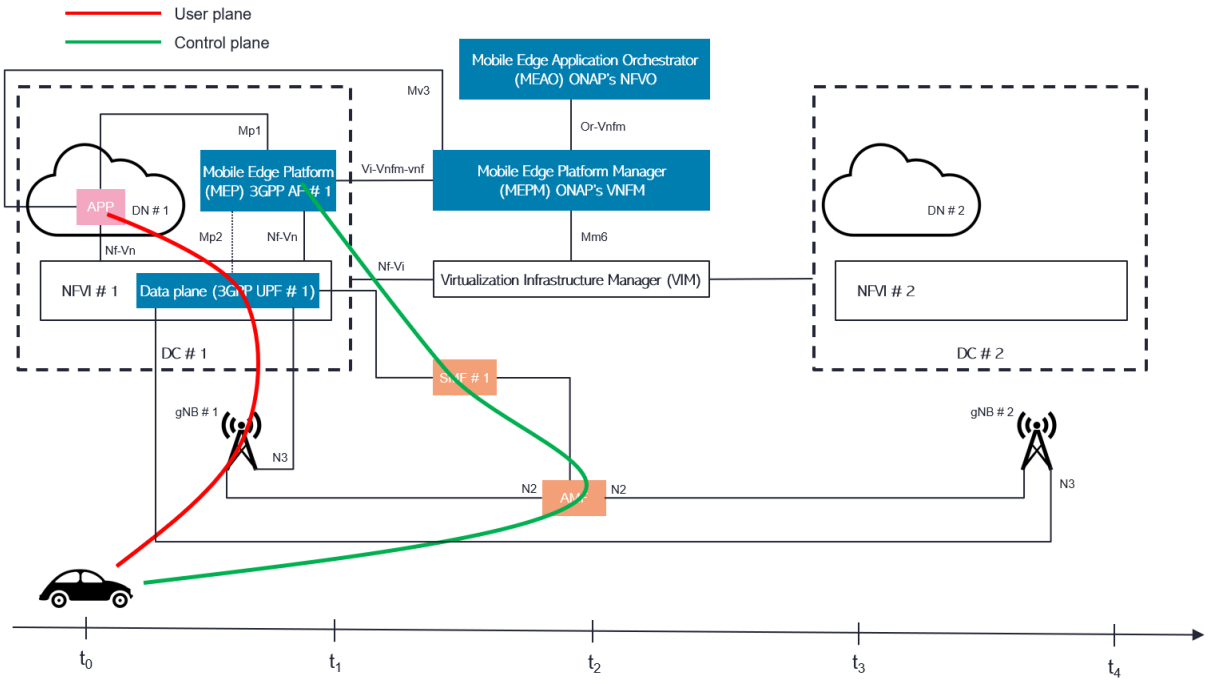

Figure 41: Autonomous Driving use-case description at time t_0

This is detailed in the sequence diagram of Figure 42.

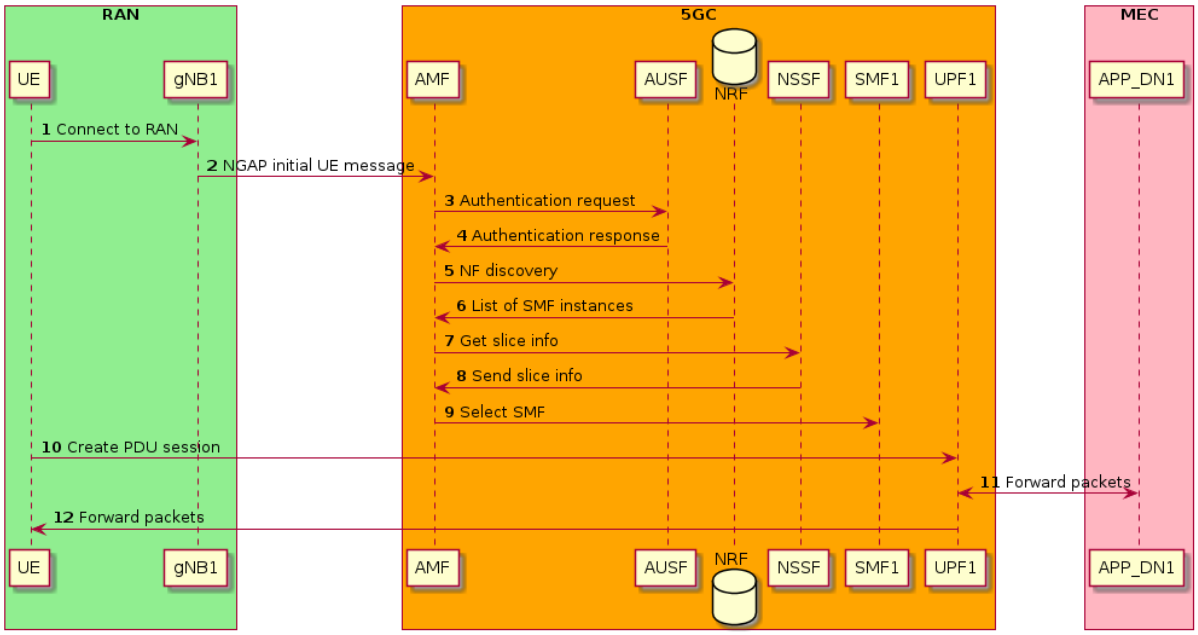

Figure 42: Autonomous Driving use-case sequence diagram at time t_0

3.2.7.2 t_1

A trigger happens (UE handover to a different gNB) that causes NWDAF to request a *Get UPF load* message to ONAP's DCAE. DCAE responds with a message including the list of UPF-IDs and load values. NWDAF decides

what UPF to recommend and sends UPF-ID to SMF. NWDAF figures out that none of the available UPFs are feasible to meet the SLAs of the NSSAI for this session and consequently NWDAF requests a new UPF onboarding to ONAP's SO. SO responds with newly created UPF-ID (in the figure below this would be UPF #2). UE is now attached to gNB #2, but still user plane traffic goes through UPF #1 and control plane through AF #1 (both belonging to DC #1), as illustrated in Figure 43.

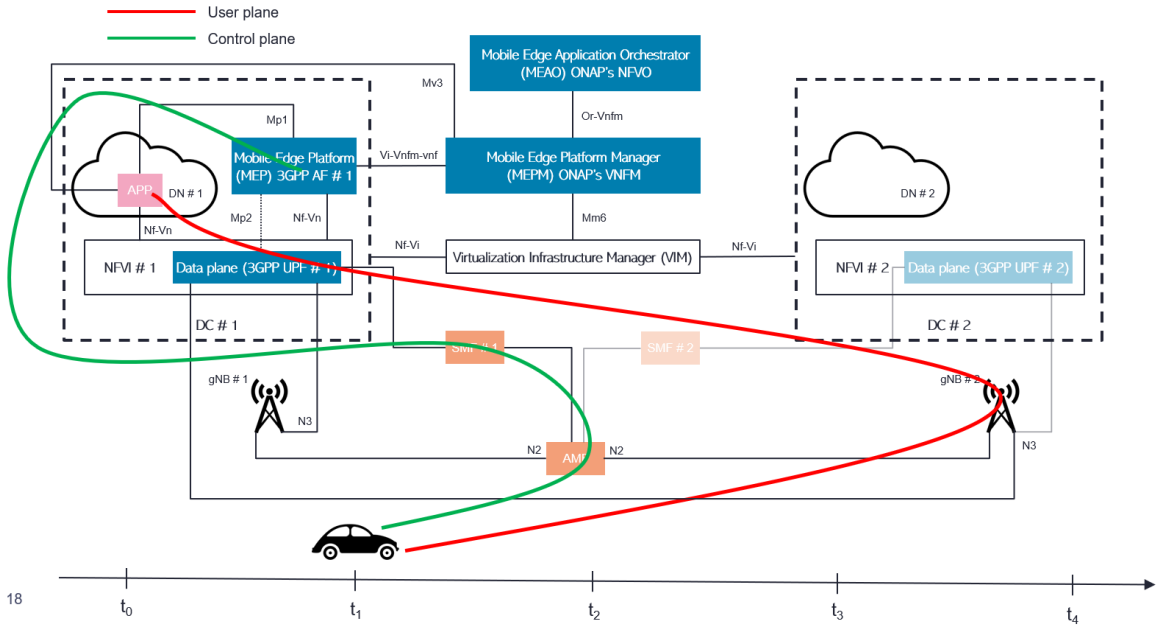

Figure 43: Autonomous Driving use-case description at time t_1

This is detailed in the sequence diagram, of Figure 44.

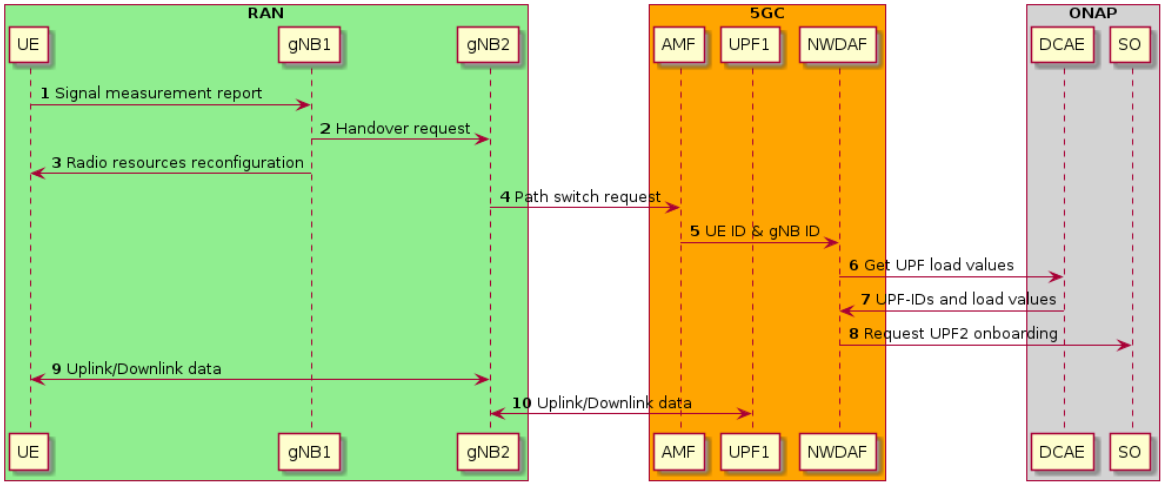

Figure 44: Autonomous Driving use-case sequence diagram at time t_1

3.2.7.3 t_2

The SMF sends a message to the selected UPF (UPF #2) to establish the PDU session. UPF #2 acknowledges the session establishment. The UE, through the app, sends traffic to the newly selected UPF. Both UPF #1 and UPF #2 will maintain connectivity to the local DN #1. The UE request a *user context transfer* (see

Appendix – Multi-Edge networking) to AF #2. This permits the creation of the user context in an APP running on local DN #2, as illustrated in Figure 45.

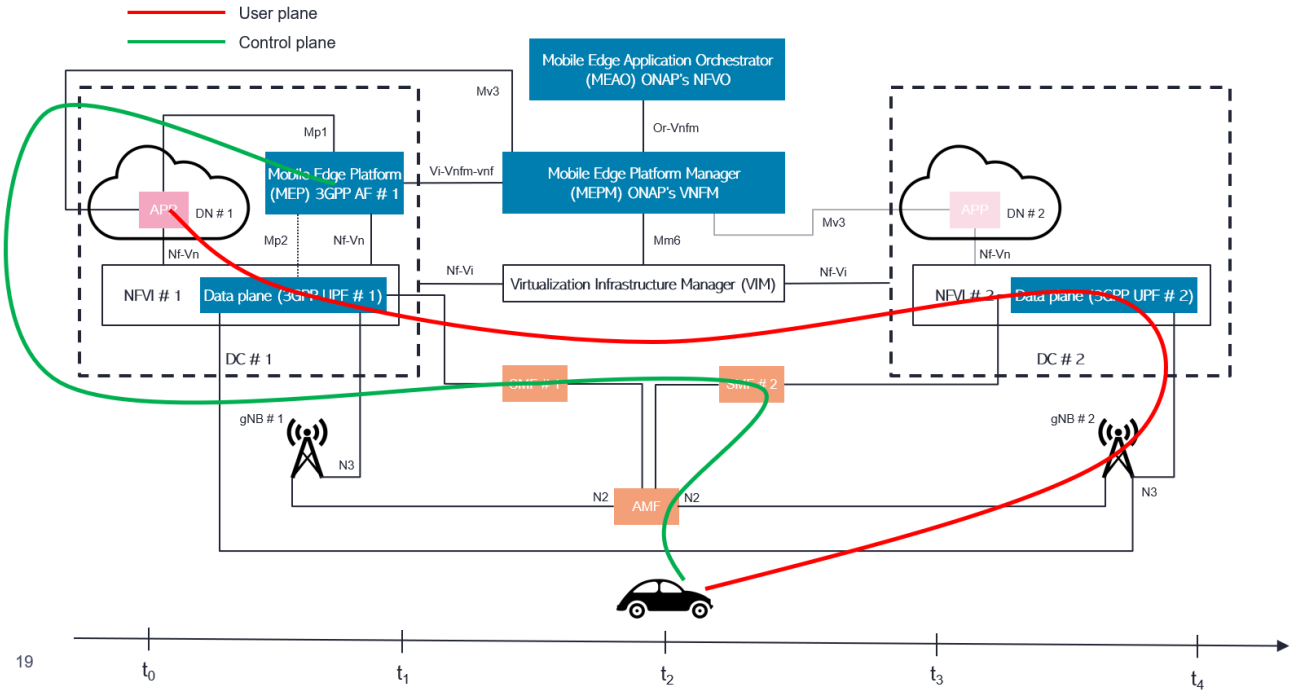

Figure 45: Autonomous Driving use-case description at time t_2

This is detailed in the sequence diagram of Figure 46.

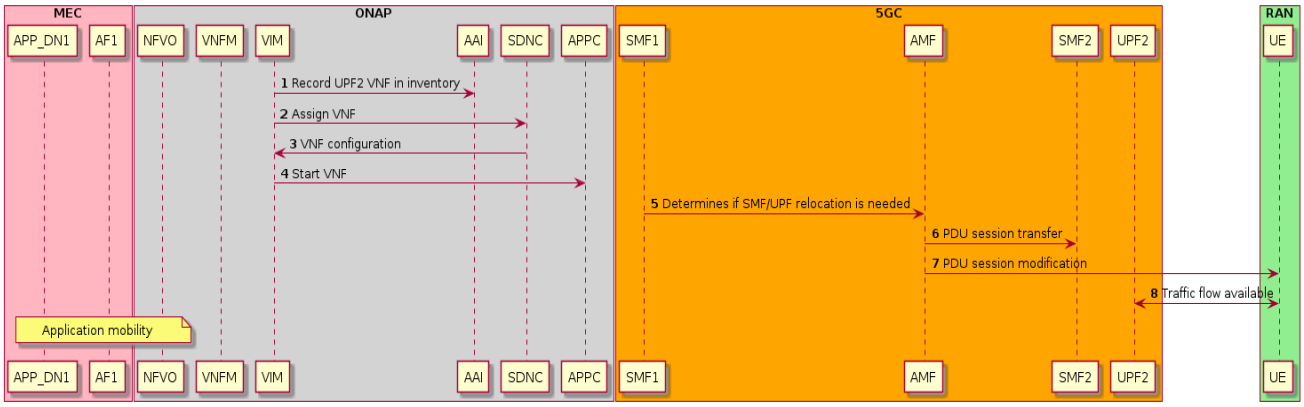

Figure 46: Autonomous Driving use-case sequence diagram at time t_2

3.2.7.4 t_3

The UE is attached to gNB #2 with the user plane going through UPF #2, but the control plane still runs through to MEC platform #1; however, the UE is now associated to the MEC app instantiated at local DN #2. NWDAF requests a *Get AS locations* message to ONAP's DCAE using App-ID as an argument. DCAE responds by sending the AS locations for the given App-ID. NWDAF request the deployment of an AS if there is a UPF that gives

connectivity to an Edge Cloud, but there is no AS for the user application(s) in that Cloud. ONAP's SO responds with the AS-ID (in the figure below, this would be AF #2), as illustrated in Figure 47.

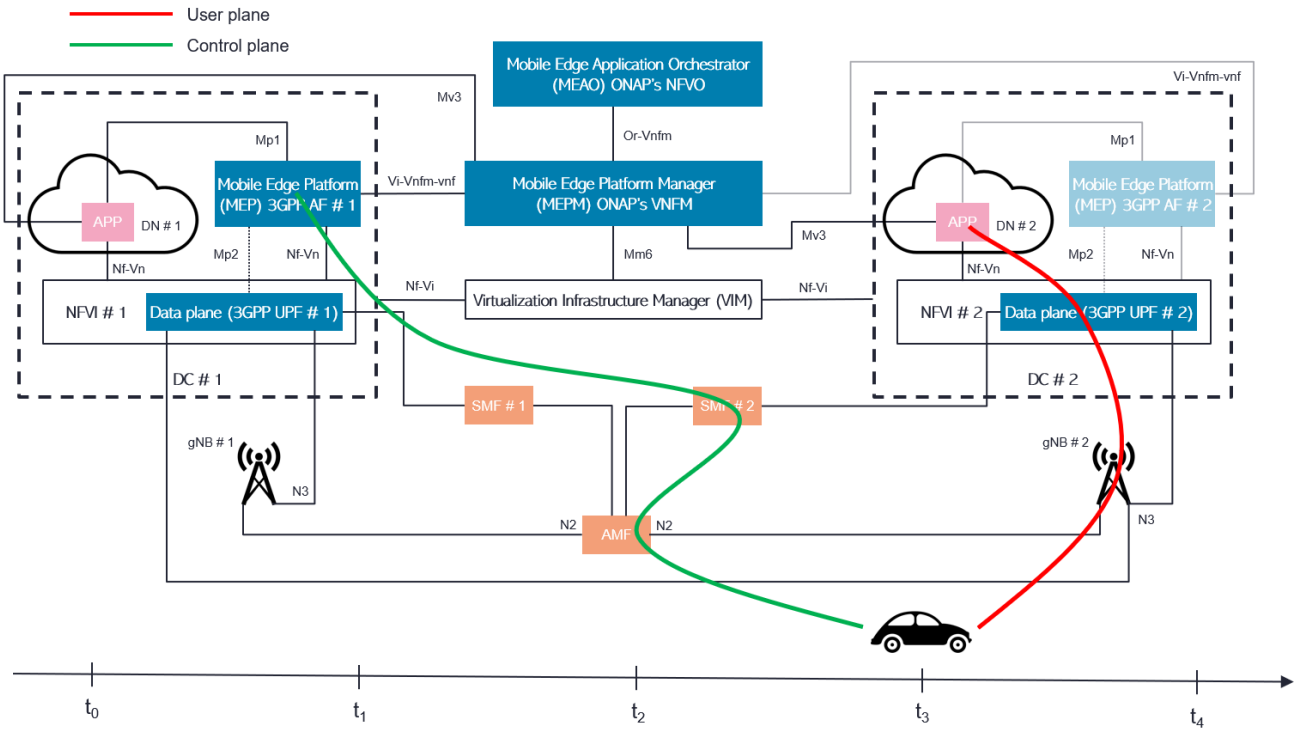

Figure 47: Autonomous Driving use-case description at time t_3

This is detailed in the sequence diagram of Figure 48.

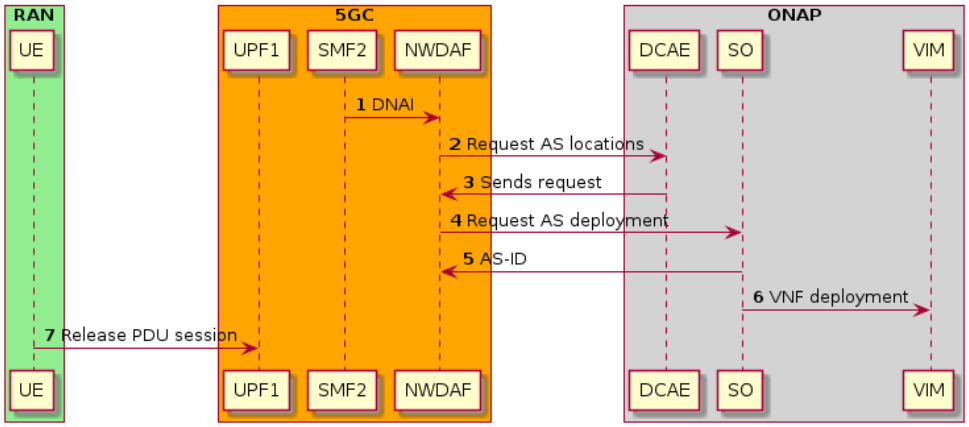

Figure 48: Autonomous Driving use-case sequence diagram at time t_3

3.2.7.5 t_4

All control plane signalling and user plane traffic goes through DC #2, as illustrated in Figure 49.

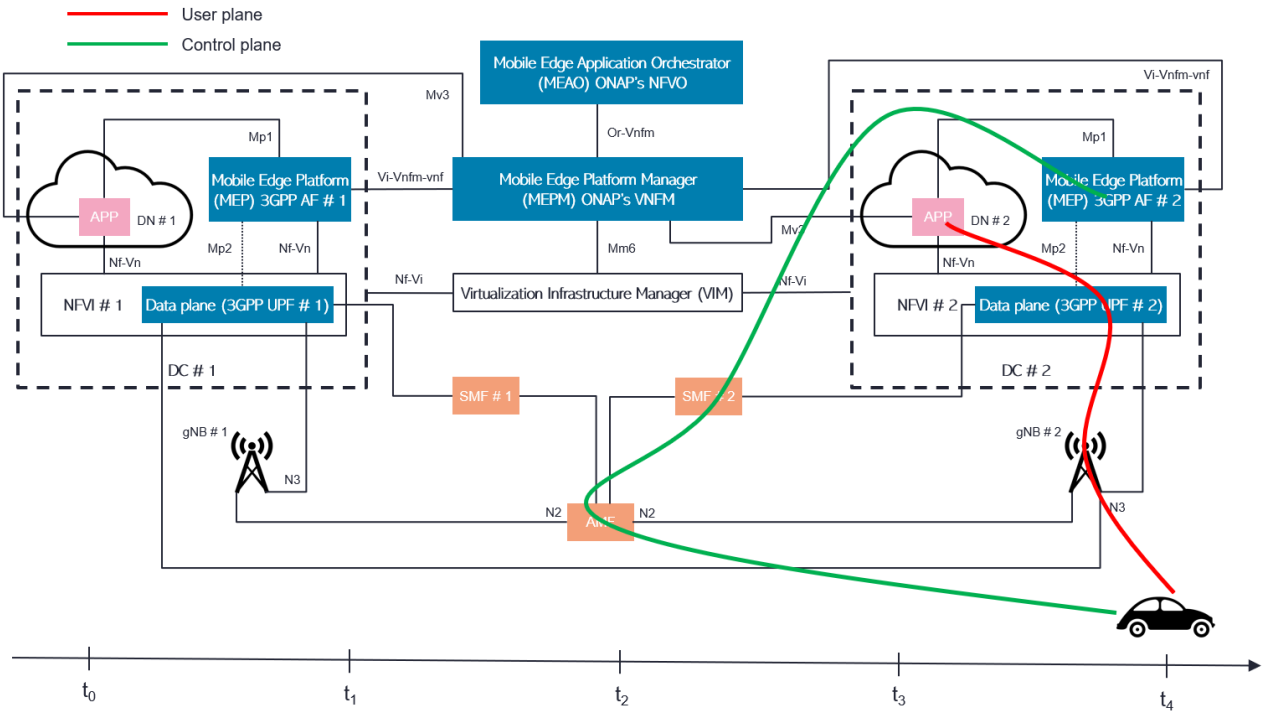

Figure 49: Autonomous Driving use-case description at time t_4

This is detailed in the sequence diagram of Figure 50.

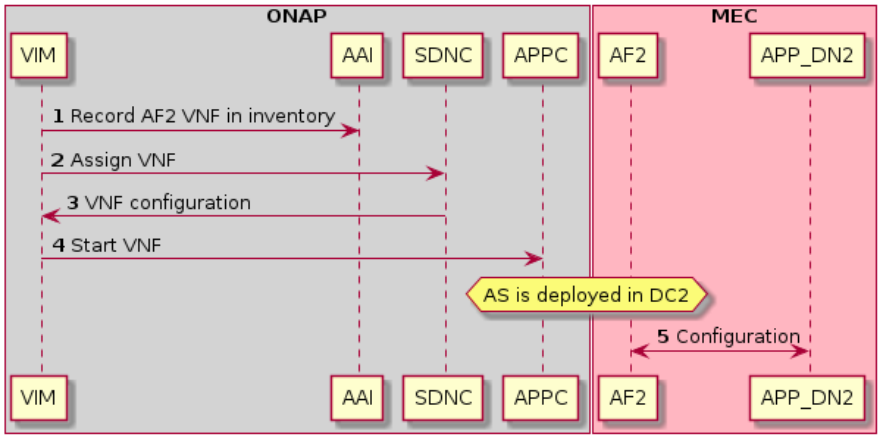

Figure 50: Autonomous Driving use-case sequence diagram at time t_4

3.2.8 Final conditions

During the journey the vehicle will automatically use the follow-me application.

3.2.9 Requirements

Herein are described the overall requirements for the use case.

3.2.9.1 Functional

The functional requirements are summarized in Table 12.

Table 12: Functional requirements of Autonomous Driving use case

Requirement ID	Requirement Description
#FReq-B01	End-to-End (E2E) slicing architecture and interfaces must support network slicing, including creation, modification, and deletion of a network slice subnet, where an instance of slice network function may be associated with one or more slices.
#FReq-B02	E2E slicing architecture shall support differentiated handling of traffic for different network slice subnets, i.e., shall support means by which the operator can differentiate policy control, functionality and performance provided in different network slice subnets (e.g., through slice aware resource management strategies such as admission control, congestion control, handover preparation, etc.), thus enabling the support for QoS differentiation within a slice
#FReq-B03	E2E slicing architecture shall support resource isolation between slices, i.e., shall enable traffic and services in one network slice subnet without impacting traffic and services in other network slice subnets in the same network
#FReq-B04	E2E slicing architecture shall enable mechanisms to avoid shortage of shared resources in one slice breaking the service level agreement for another slice
#FReq-B05	E2E slicing architecture shall enable defining a priority order between different network slice subnets in case multiple slices compete for resources on the same network
#FReq-B06	Redundant transmission support for URLLC must be supported in the 5G core network
#FReq-B06	URLLC supported in RAN
#FReq-B07	Handover procedures supported in RAN

3.2.9.2 Non-Functional

The non-functional requirements are summarized in Table 13.

Table 13: Non-Functional requirements of Autonomous Driving use case

Requirement ID	Requirement Description
#NFReq-B01	OSS should have dynamic management capabilities to respond to real-time changes in the service

#NFReq-B02	Applications that are sensitive to the distance between the application and the user may use a "follow-me" functionality during the trip

4. Validation Framework

This section provides an initial draft of the validation framework, towards the deployment of the diverse components of the OREOS platform, including RAN, edge and core. This section also provides a set of KPIs that will be employed to assess the performance of OREOS in the use cases.

4.1 Software validation

Once the software is successfully installed, in order to make the tests one must access the UE container and verify the connectivity with the Internet:

```
ping google.com -I <<ip-address-user-equipment-interface>>
```

Afterwards, a more elaborated validation shall take place for the four main blocks (RAN, CN, MEC and ONAP).

4.1.1 RAN & CN

In order to validate the software implementation of RAN & CN, a Wireshark trace (or equivalent) of the following workflow must be assessed, in particular to assess the exchange of messages between the diverse components as depicted in Figure 51.

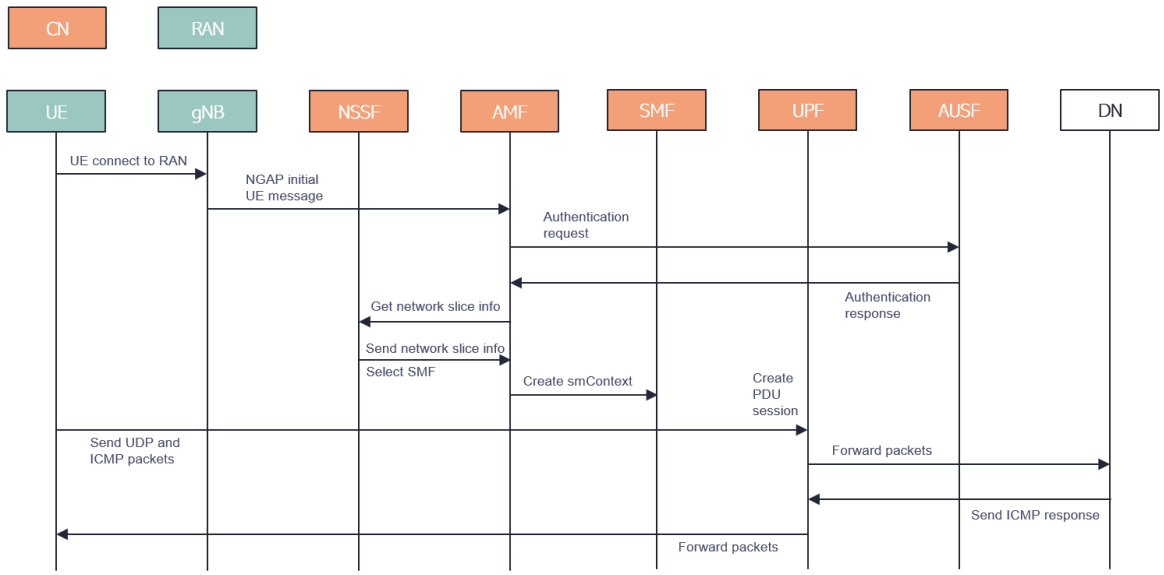

Figure 51: Sequence diagram of RAN and CN

4.1.1.1 Protocol stack

The trace analysis (performed via Wireshark tool or similar one) should be able to identify the following protocols, both for data plane, as illustrated in Figure 52.

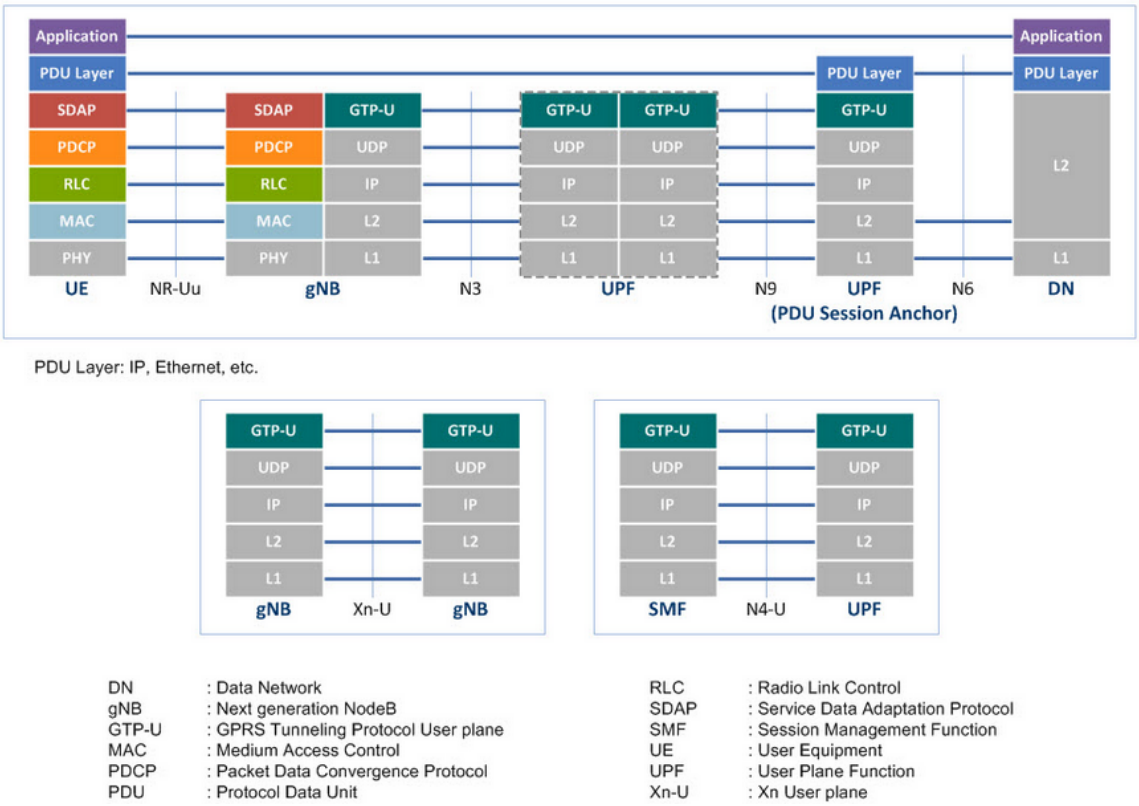

Figure 52: User plane protocol stack

And also for control plane, as illustrated Figure 53.

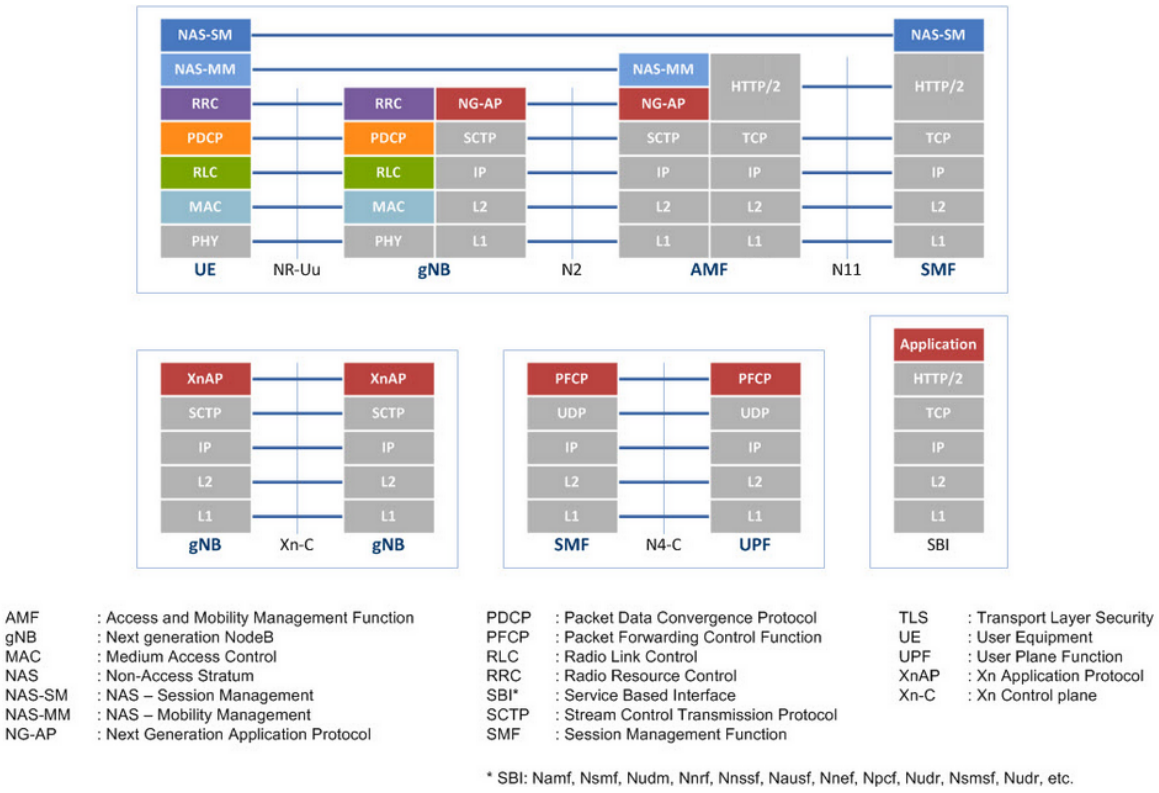

Figure 53: Control plane protocol stack

4.1.2 RAN deployment

The process of deploying OpenAirInterface's RAN, in RF SIM mode, starts by pulling the right Docker images from Docker-Hub, and re-tag them accordingly with the docker-compose YAML file present in the 5g_rfsimulator directory part of the OAI's source code (develop branch), as shown in Figure 54.

```
$ docker pull mysql:5.7
$ docker pull rdefosseoai/oai-amf:latest
$ docker pull rdefosseoai/oai-nrf:latest
$ docker pull rdefosseoai/oai-smf:latest
$ docker pull rdefosseoai/oai-spgwu-tiny:latest

$ docker pull rdefosseoai/oai-gnb:develop
$ docker pull rdefosseoai/oai-nr-ue:develop
$ docker image tag rdefosseoai/oai-amf:latest oai-amf:latest
$ docker image tag rdefosseoai/oai-nrf:latest oai-nrf:latest
$ docker image tag rdefosseoai/oai-smf:latest oai-smf:latest
$ docker image tag rdefosseoai/oai-spgwu-tiny:latest oai-spgwu-tiny:latest

$ docker image tag rdefosseoai/oai-gnb:develop oai-gnb:develop
$ docker image tag rdefosseoai/oai-nr-ue:develop oai-nr-ue:develop
```

Figure 54: Docker images for OAI-RAN deployment

Note that in Figure 54 AMF, NRF, SMF, SPGWU docker images have been pulled as well, so that we have a running CN on top of RAN SIM before deploying the RAN itself. Hence representing a mandatory step, followed by the proper deployment of the 5G CN as shown in Figure 55.

```
$ cd ci-scripts/yaml_files/5g_rfsimulator
$ docker-compose up -d mysql oai-nrf oai-amf oai-smf oai-spgwu oai-ext-dn
Creating network "rfsim5g-oai-public-net" with driver "bridge"
Creating network "rfsim5g-oai-traffic_net-net" with driver "bridge"
Creating rfsim5g-oai-nrf ... done
Creating rfsim5g-mysql      ... done
Creating rfsim5g-oai-spgwu ... done
Creating rfsim5g-oai-amf   ... done
Creating rfsim5g-oai-smf   ... done
Creating rfsim5g-oai-ext-dn ... done
```

Figure 55: Deployment of OAI CN

With regards to the necessary configurations to the docker-compose YAML file, the reader must follow the topology presented in the Figure 56, especially the DNS IP address has to match the one used by the host machine where CN is being deployed.

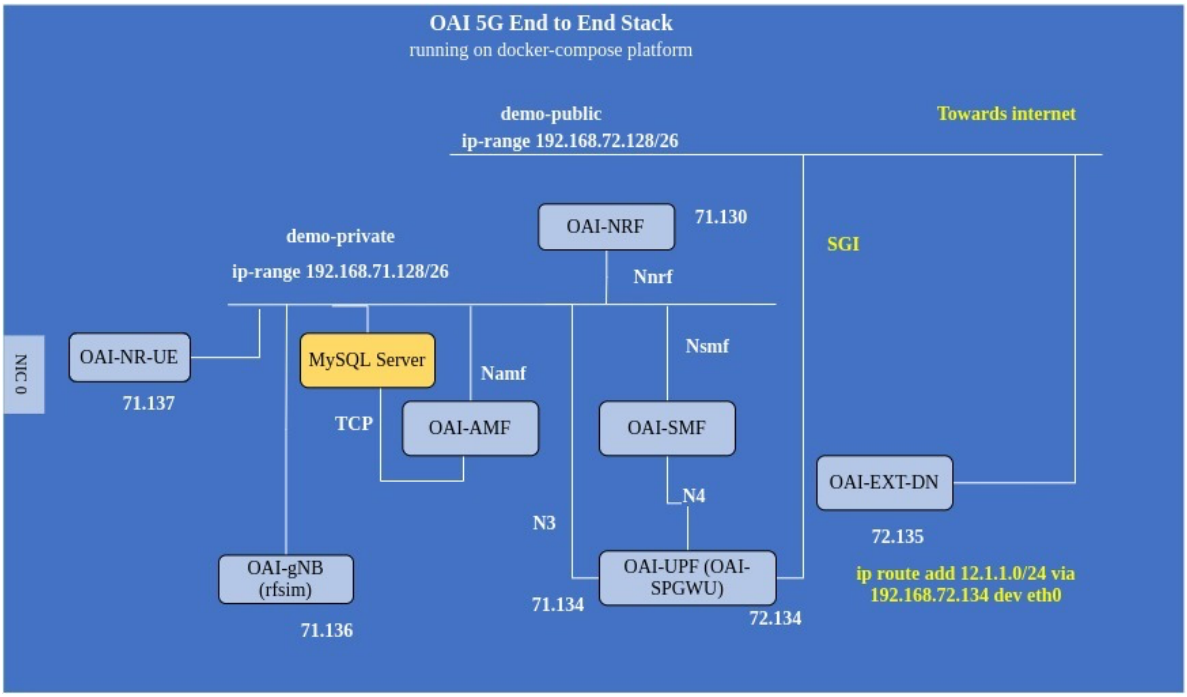

Figure 56: RAN deployment

Besides, the PLMN identification parameters have to match in all three entities: gNB, AMF, User Plane of the Packet Data Network Gateway (SPGW-U). As an example, if AMF establishes that MCC, MNC and TAC are 268

(Portuguese Mobile Country Code), 06 (Mobile Network Code) and 1 (Tracking Area Code) then both gNB and AMF must have this same codes. Finally, the reader can now deploy the gNB as shown in Figure 57.

```
$ docker-compose up -d oai-gnb
rfsim5g-oai-nrf is up-to-date
rfsim5g-oai-spgwu is up-to-date
rfsim5g-oai-ext-dn is up-to-date
Creating rfsim5g-oai-gnb ... done
```

Figure 57: OAI-gNB deployment

Before checking the status of the connection between RAN and CN, it is first necessary to check the deployed containers, as they must all be in a healthy state, as shown in Figure 58.

```
$ docker-compose ps -a
       Name                     Command                  State                 Ports
-------------------------------------------------------------------------------------------------
rfsim5g-mysql        docker-entrypoint.sh mysqld      Up (healthy)   3306/tcp, 33060/tcp
rfsim5g-oai-amf      /bin/bash /openair-amf/bin ...   Up (healthy)   38412/sctp, 80/tcp, 9090/tcp
rfsim5g-oai-ext-dn   /bin/bash -c  apt update;  ...   Up (healthy)
rfsim5g-oai-gnb      /opt/oai-gnb/bin/entrypoin ...   Up (healthy)
rfsim5g-oai-nrf      /bin/bash /openair-nrf/bin ...   Up (healthy)   80/tcp, 9090/tcp
rfsim5g-oai-smf      /bin/bash -c /openair-smf/ ...   Up (healthy)   80/tcp, 8805/udp, 9090/tcp
rfsim5g-oai-spgwu    /openair-spgwu-tiny/bin/en ...   Up (healthy)   2152/udp, 8805/udp
```

Figure 58: Status of components

Last but not least, in order to check the gNB deployment status, one way is to access the AMF's container logs using the command "docker logs rfsim5g-oai-amf", and in fact see if the CN has any gNB registered and connected to it, as shown in Figure 59.

```
$ docker logs rfsim5g-oai-amf
...
[AMF] [amf_app] [info ] |----------------------------------------------------gNBs' information------------------
[AMF] [amf_app] [info ] |    Index    |      Status      |       Global ID       |       gNB Name       |
[AMF] [amf_app] [info ] |      1      |    Connected     |         0x0       |         gnb-rfsim        |
[AMF] [amf_app] [info ] |-------------------------------------------------------------------------------------------
...
```

Figure 59: Logs of OAI AMF

4.1.3 ONAP

The following subsections describe the software validation in ONAP.

4.1.3.1 5GC deployment

ONAP needs to be tested against basic NFs of the CN (AMF/SMF/UPF) via Helm Charts related to Docker images (with config info about the DCAE Collector IP and port). The core NSSMF shall be tested for instantiation of a 5G core service comprising the above-mentioned elements, namely AMF, SMF and UPF CNFs. The configurations applied to these CNFs during instantiation and modification configurations comprise of S-NSSAI and the configuration mechanism to be config-map type of k8s.

This was successfully tested by Orange with CBA[6] package for ONAP[7] and helm charts for dummy CNFs[8]. For OREOS this test must be repeated, but for the Helm charts of our chosen open-source CN. Procedure is as follows:

1. Once a 5G slice order request is received to be processed, the SO creates the service instance and updates service instance information to A&AI. SO further triggers SDN-C to create service instance info in MD-SAL. SDN-C further triggers CDS to execute resource assignment workflow;
2. SO creates VNF and updates VNF information to A&AI. SO further triggers SDN-C to create VNF info in MD-SAL. SDN-C further triggers CDS to execute resource assignment workflow;
3. SO creates vf-module and updates vf-module information to A&AI. SO further triggers SDN-C to create VNF info in MD-SAL. SDN-C further triggers CDS to execute resource-assignment workflow. The workflow triggers script *ProfileUpload.kt* (from CBA package) which updates profile artifact with S-NSSAI parameters and create profile in K8splugin;
4. SO triggers config-assign workflow in CDS. CDS processes config-assign workflow;
5. SO invokes infra-workload API of MultiCloud which further invokes K8splugin. Before processing of instantiating helm charts, the K8splugin retrieves profile artifact and updates the ConfigMap with S-NSSAI information from *override.yaml* (from CBA package) of profile artifact;
6. SO updates A&AI with vf-module status as Activated;
7. SO triggers config-deploy workflow of CDS. CDS processes the config-deploy workflow. The workflow would execute script *DayOneConfig.kt* (from CBA package). The script performs the following activities in sequence. Steps from second to last bullet point will be executed for all the vf-modules except base:
    - Script invokes A&AI and fetch vf-modules details;
    - From each vf-module response it retrieves heat-stack-id which is actually k8splugin instance id;
    - With instance-id, invokes K8splugin instance API to fetch ConfigMap name information;
    - Updates the ConfigMap name in config-template artifact;

[6] Controller Blueprint Archive (CBA) is a service design tool, fully model-driven, intent based package for provisioning and configuration automation.
[7] https://gerrit.onap.org/r/c/ccsdk/cds/+/113518
[8] https://wiki.onap.org/download/attachments/84671993/5G_Core_Helm_package.zip

- Executes Create template API of K8splugin;
- Uploads the config-template artifact by invoking Upload config template content API of K8splugin;

8. SO updates A&AI with VNF status as Active;
9. SO updates A&AI with Service Instance status as Active.

Figure 60 illustrates steps 1-4.

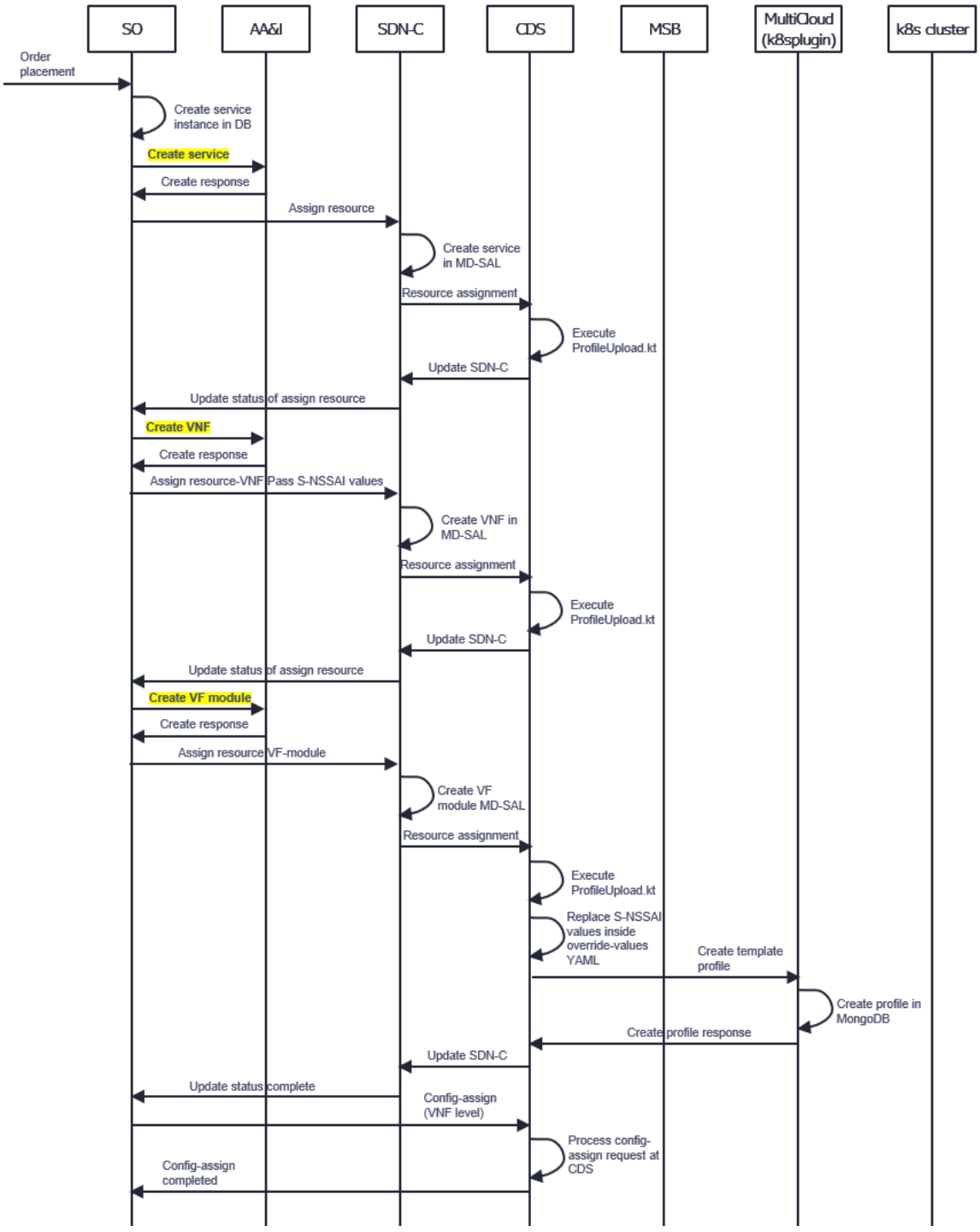

Figure 60: 5GC deployment – part 1

While Figure 61 illustrates the remaining steps.

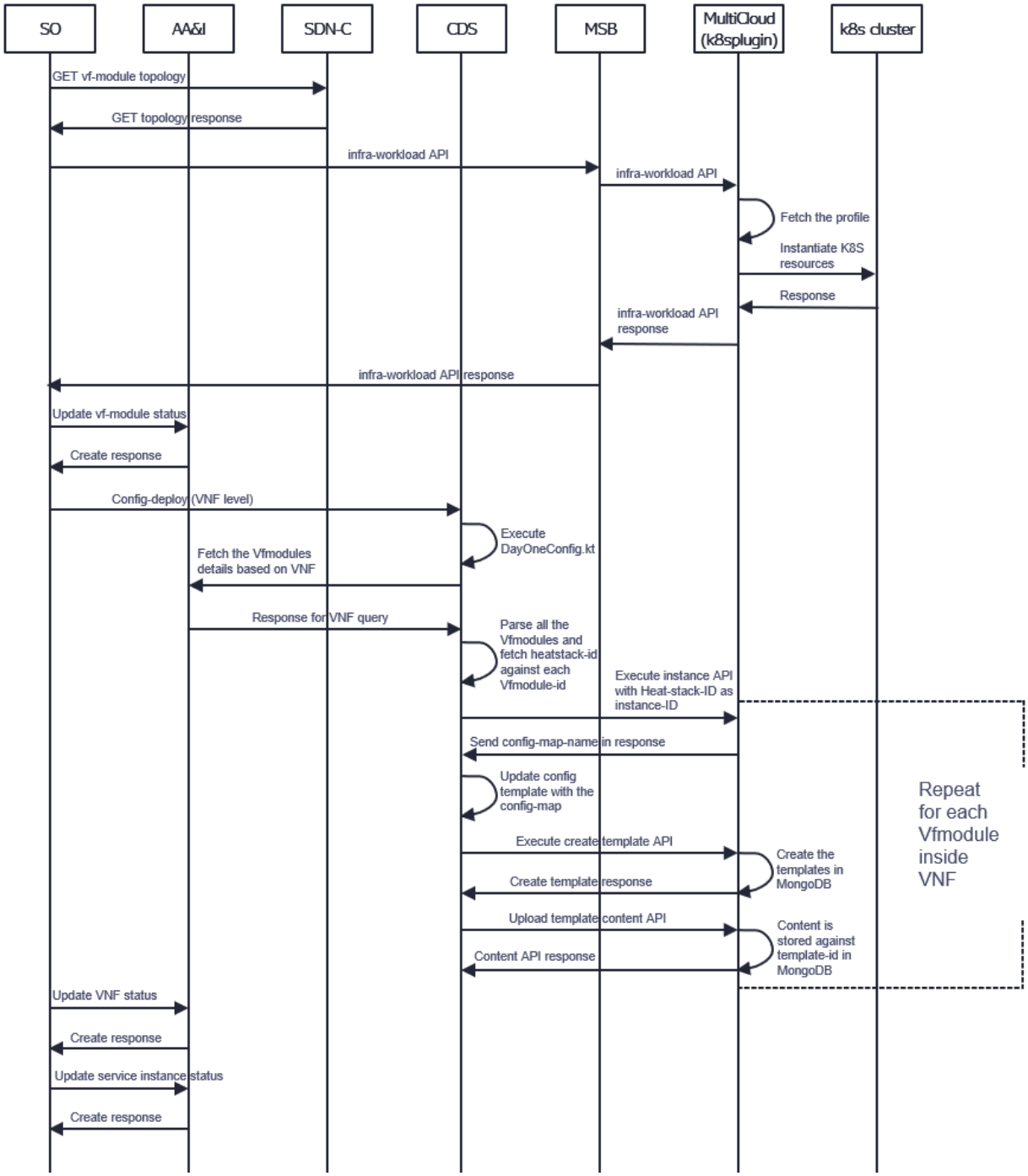

Figure 61: 5GC deployment – part 2

4.1.3.2 Slice instantiation

The process of slice ordering is illustrated in the workflow of Figure 62, as per references (ONAP-white-paper, 2020) – (Huang & Min).

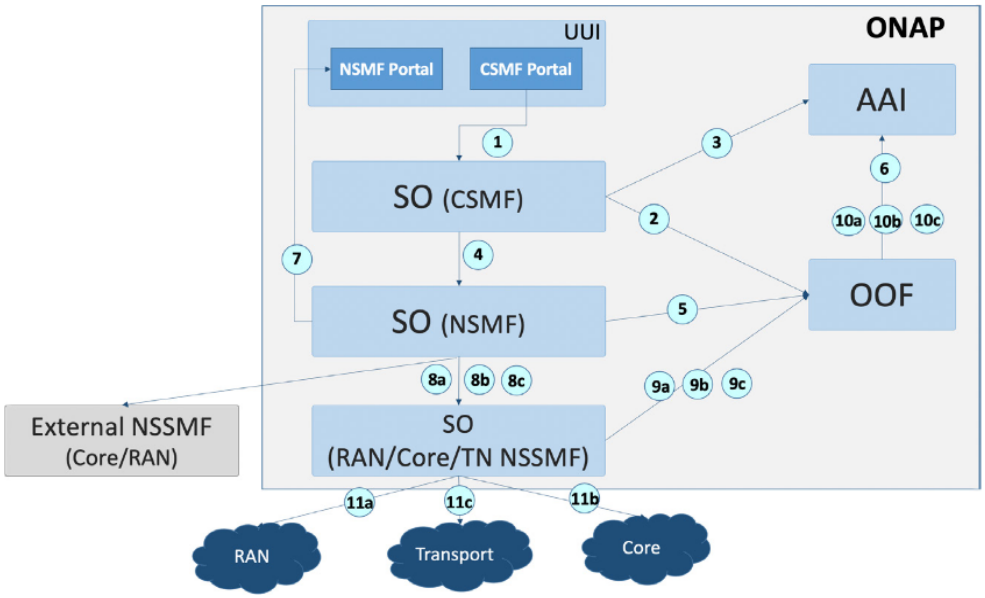

Figure 62: Workflow of service ordering

The steps in this figure are explained below:

1) The slice consumer requests a communication service with relevant inputs on the service's characteristics and performance requirements via the CSMF portal in UUI. The CSMF portal sends a request to SO acting as CSMF denoted as SO (CSMF – Communication Service Management Function);

2) The SO of CSMF converts the communication service request into a Service Profile and triggers OOF for providing a suitable Network Slice Template (NST);

3) The SO of CSMF updates AAI with communication service details. OOF provides the NST back to SO;

4) The SO of CSMF then triggers SO (NSMF) for allocating a suitable NSI;

5) The SO of NSMF triggers OOF for selecting a suitable NSI (if allowed to be shared and available, or Slice Profiles of the constituent NSSIs for a new NSI to be created);

6) OOF checks Policy inputs and existing inventory (AAI), and then selects the suitable NSI and provides it to the SO of NSMF. In case no suitable NSI exists or if the service request required a non-shared service to be instantiated, OOF provides suitable RAN/Core/Transport Network (TN) Slice Profiles to create a new NSI;

7) The SO of NSMF shares details of the selection made by OOF with NSMF Portal in UUI for the operator to check and make any modifications if needed;

8) The SO of NSMF triggers RAN/Core/TN NSSMFs (in Steps 8a/8b/8c respectively) for allocating/updating the RAN/Core/TN NSSIs. Note that the RAN/Core NSSMFs could be within or external to ONAP;

9) SO (RAN/Core/TN NSSMF) triggers OOF for selecting a suitable NSSI with the corresponding Slice Profile;

10) OOF checks Policy inputs and existing inventory of NSSIs. If sharing is allowed and an existing NSSI matches the Slice Profile, then OOF returns the existing NSSI that can be reused (in case of RAN and Core), otherwise it informs SO (NSSMF) to create a new RAN/Core/TN NSSI;

11) SO (RAN/Core/TN NSSMF) then creates/updates RAN/Core/TN NSSI and its constituents.

The inventory (AAI) is also updated with the Service Profile, NSI, NSSIs, Slice Profiles and S-NSSAI, and appropriate links are created (not shown in above sequence diagram). This procedure is more detailed in Figure 63 illustrating this workflow in the form of a call-flow, for a successful ONAP deployment.

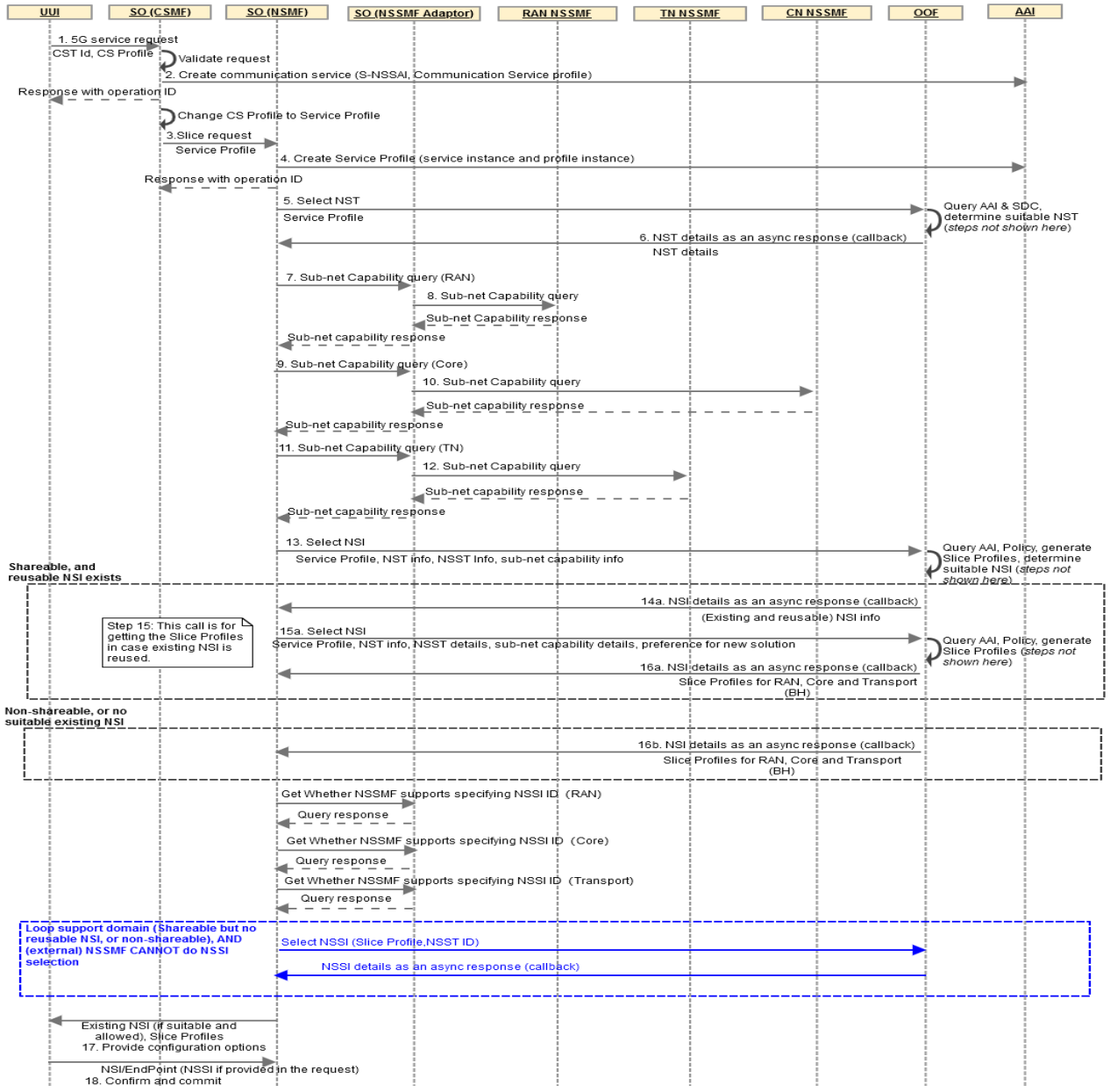

Figure 63: Slice instantiation sequence diagram – part 1

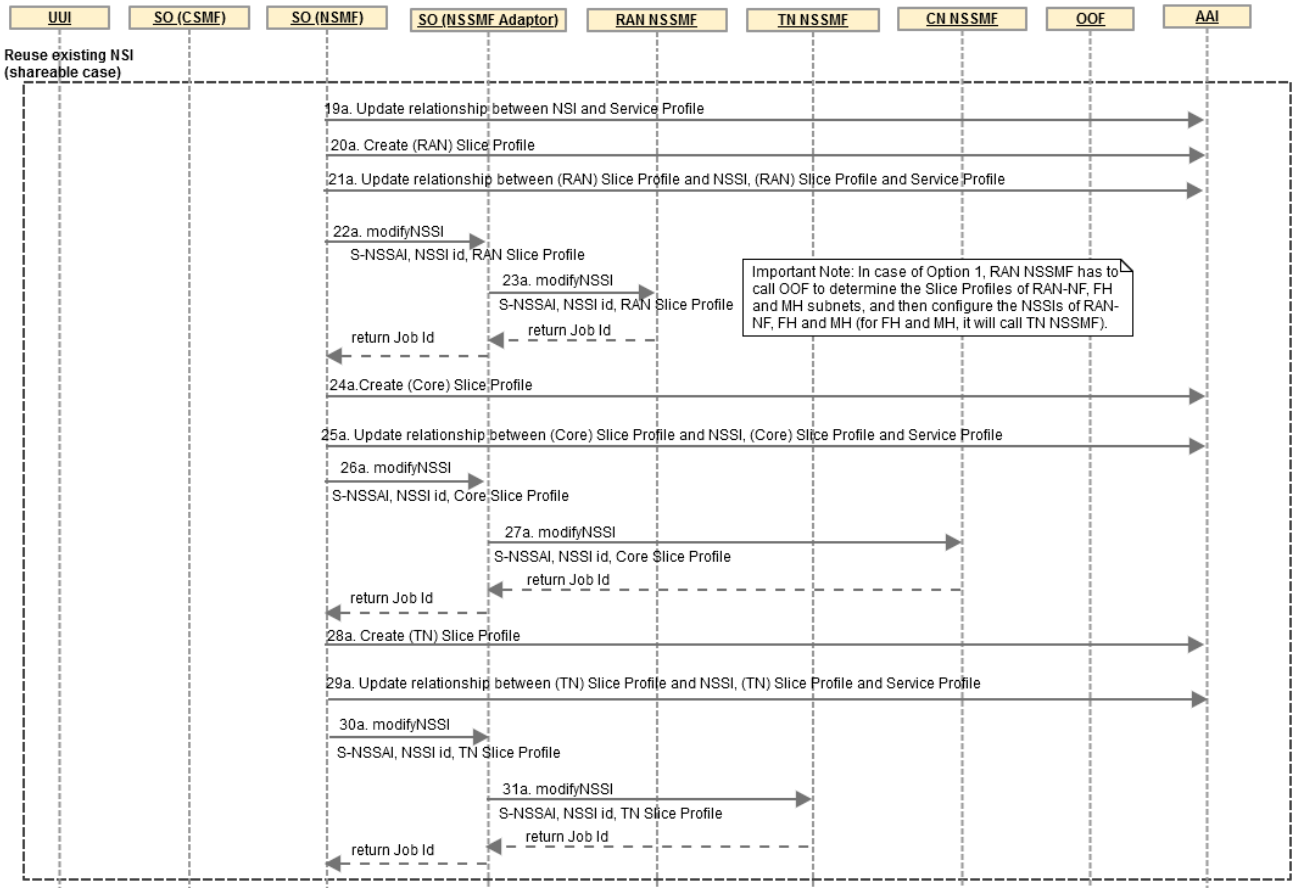

Figure 64: Slice instantiation sequence diagram – part 2

The sequence diagram in Figure 64 shows the NSMF and NSSMF interactions for RAN Slicing Option 1[9], and when all NSSMFs are within ONAP. Assumptions and notes:

- RAN Option 1 or Option 2 shall be determined by NSMF by examining the NST contents;
- NSMF shall determine if it needs NSSI selection (in case of external NSSMF) for a domain by checking the vendor in the NSST;
- In case of internal NSSMF, the "<Domain> NSSMF" (domain = AN/TN/CN) shown in below figure shall be SO. SO (NSSMF) will further interact with other components (e.g. SDN-C/SDN-R/CDS) for domain-specific NSSMF actions;
- Steps in Blue are applicable only in case of external NSSMF;
- Step 29b: NSSI id shall be passed from NSMF to NSSMF in case of (external) NSSMF that cannot do NSSI selection, as illustrated in Figure 65.

[9] **Option 1** = RAN NSSI shall comprise of RAN NF NSSI, Front-Haul (FH) NSSI and Mid-Haul (MH) NSSI. RAN NSSMF shall be responsible for RAN NSSI and RAN NF NSSI orchestration, and shall trigger Transport NSSMF for FH and MH NSSI orchestration. NSI shall comprise of RAN NSSI, TN Back-Haul (BH) NSSI and Core NSSI.
Option 2 = External RAN NSSMF will be invoked with RAN Slice profile and FH and MH are abstracted out.

Figure 65: Slice instantiation sequence diagram – part 3

The NSI/NSSI deallocation must be assessed as well, according to Figure 66 and Figure 67.

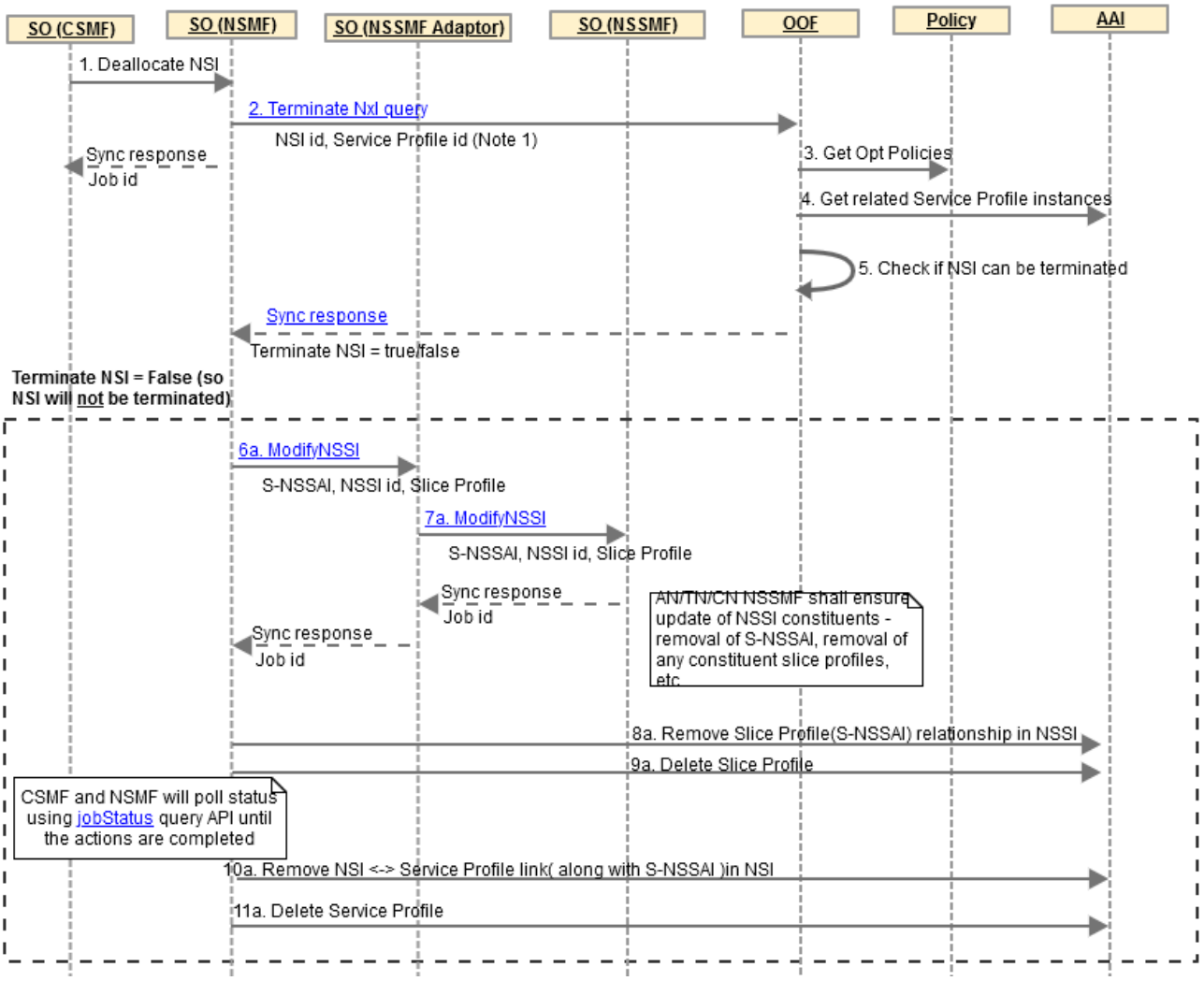

Figure 66: Slice instantiation sequence diagram – part 4

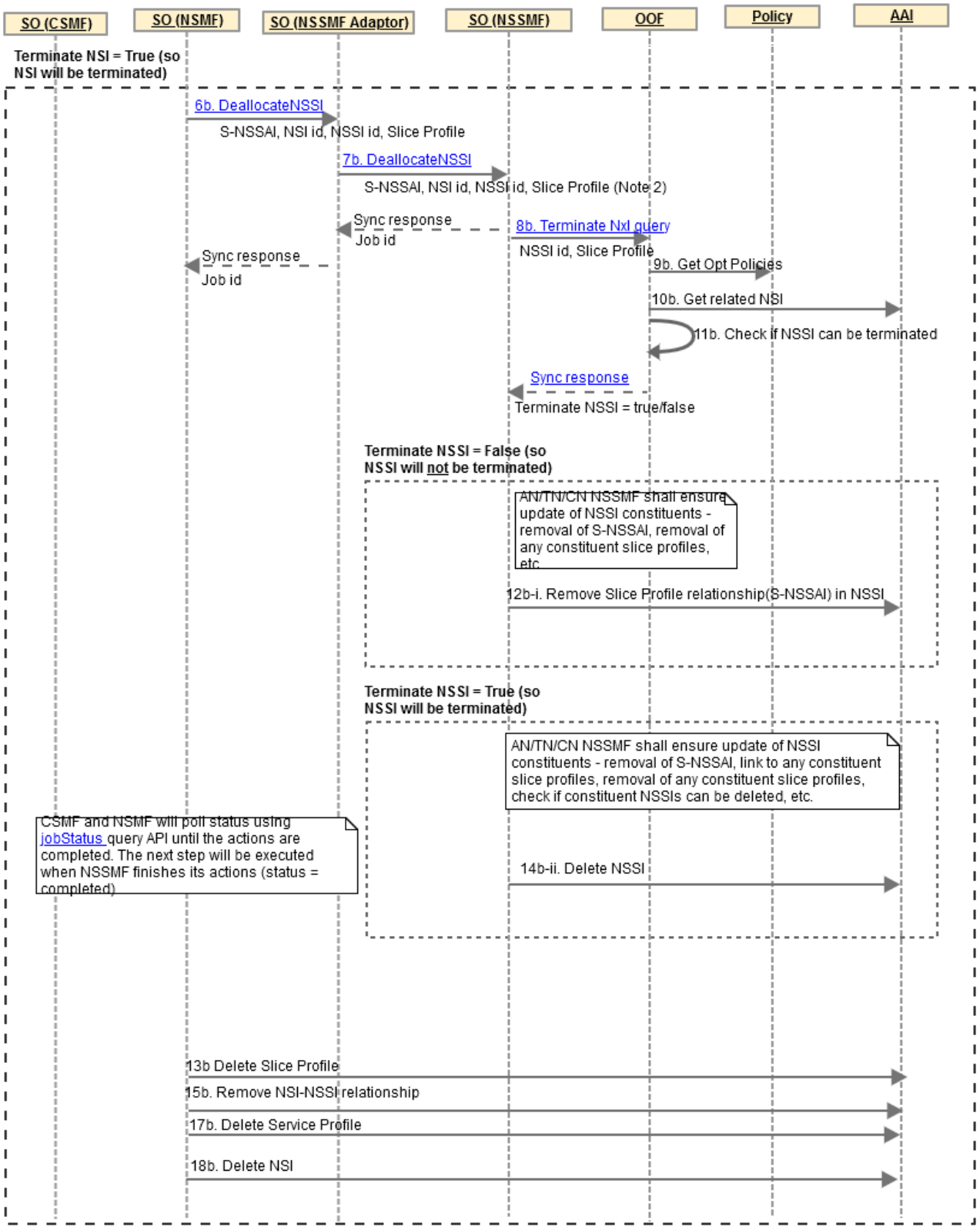

Figure 67: Slice instantiation sequence diagram – part 5

4.1.3.3 Slice optimization

Figure 68 illustrates the slicing optimization sequence diagram, that must be replicated when performing the software validation.

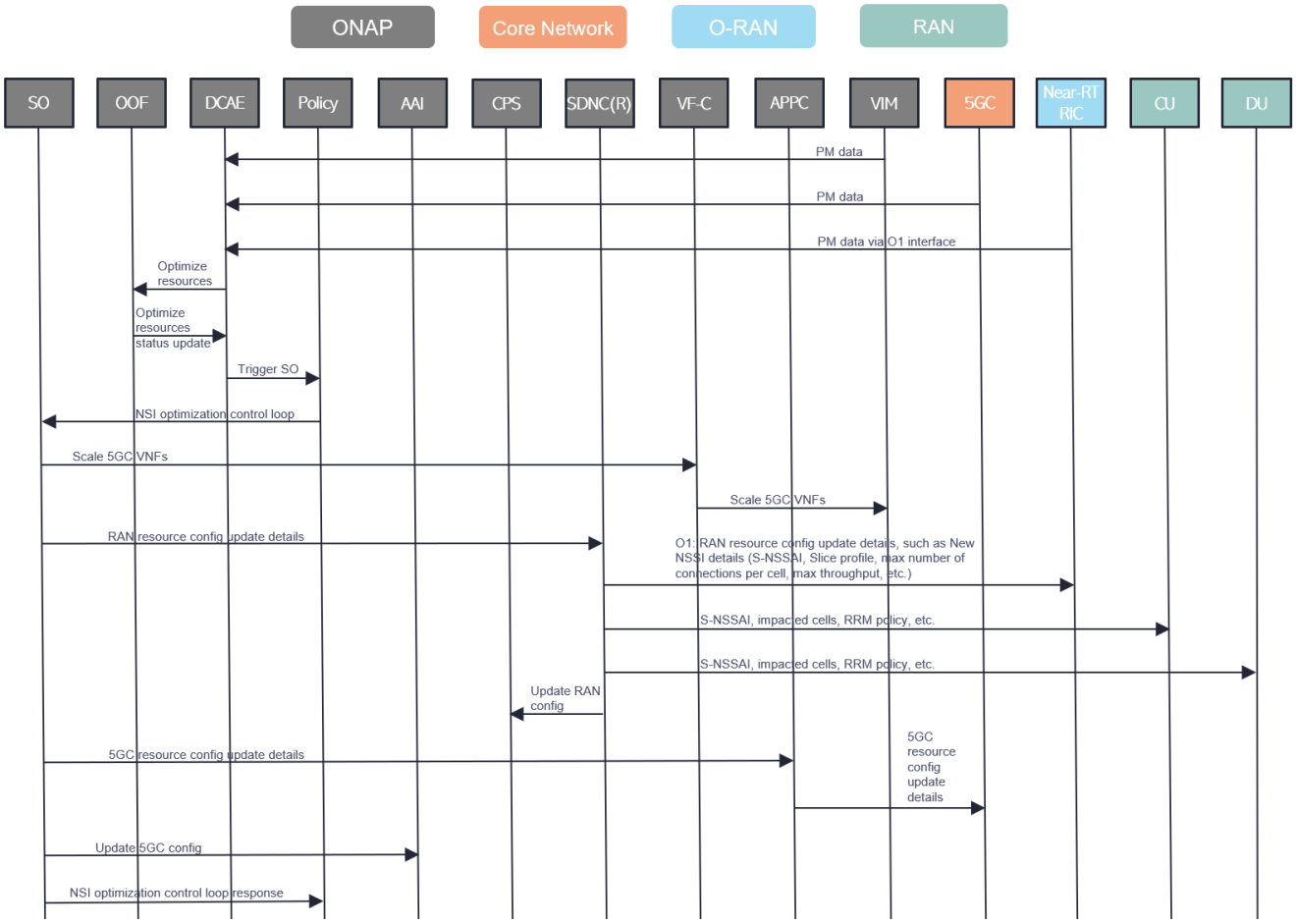

Figure 68: Slice optimization sequence diagram

4.1.4 Multi-access Edge Computing

In order to validate the software implementation of MEC, a Wireshark trace (or equivalent) of the workflow[10] depicted in Figure 69 must be assessed:

[10] The workflow either uses hops N33+N30 or just N5, depending on whether the AF is trusted or not.

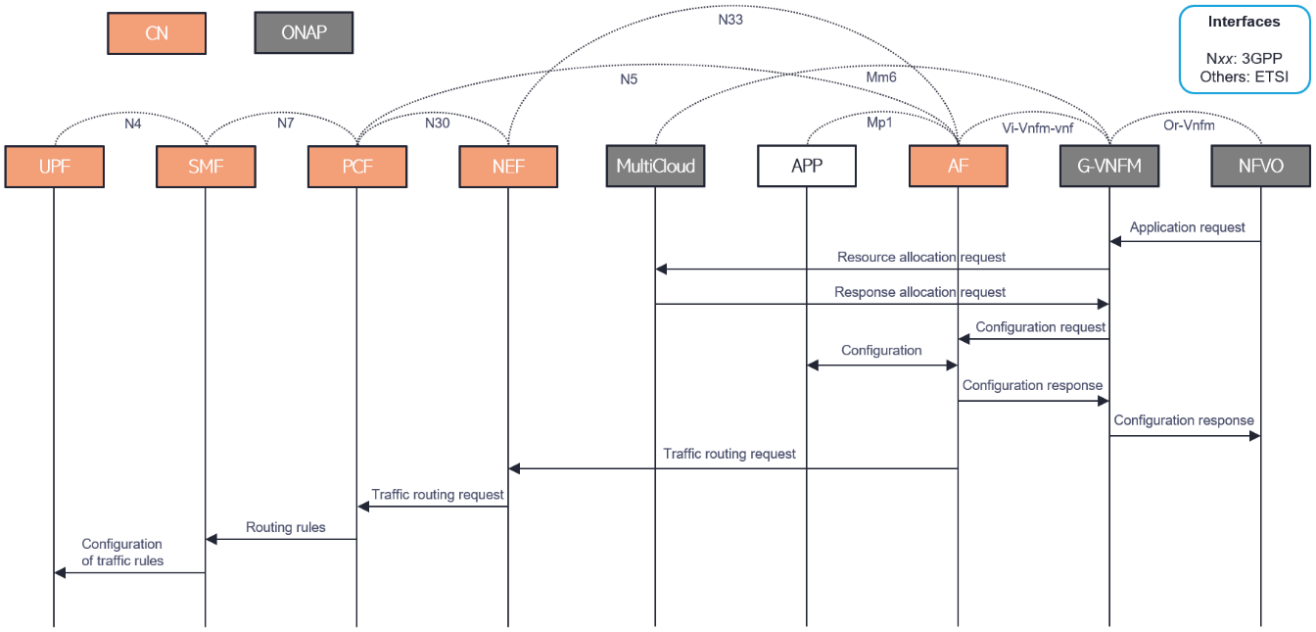

Figure 69: APP instantiation in Edge Computing sequence diagram

1. NFVO checks the APP configuration data and selects the appropriate MEC host by sending a request to the G-VNFM (APP provides information such as *max latency*);
2. G-VNFM, acting as the MEPM, initiates a resource allocation request to the virtualization infrastructure manager specifying the requested resource;
3. The MultiCloud, acting as a Mm6 interface to VIM Manager, allocates the resources according and sends the response back to the G-VNFM;
4. G-VNFM provides the AF (which acts as a MEP) with the configuration including the traffic rules to be configured, the required and optional services;
5. AF configures the traffic rules for the application instance and sends a configuration response to the G-VNFM;
6. This response is communicated to the MEAO, i.e., to the NFVO;
7. MEP, i.e., the AF, interacts with the PCF to request traffic routing by sending information that identifies the traffic to be routed. This can be done directly if the MEP is considered a trustworthy AF, or via NEF if it is not;
8. PCF, based on the received information, transforms the request into policies and provides the routing rules to the appropriate SMF;
9. SMF identifies the target UPF and initiates configuration of the traffic rules.

When a MEC application is instantiated, no traffic is routed to the application until the application is ready to receive traffic and the underlying data plane is configured to route the traffic towards it. This configuration is done by the MEC platform. When deployed in a 5G network, a MEC platform, is in the role of a 5G AF towards the 5G core network. It interacts with the PCF to request traffic steering by sending information that identifies the traffic to be steered. The PCF will transform the request into policies that apply to targeted PDU session(s) and provides the routing rules to the appropriate Session Management Function (SMF). Based on the received information, the SMF identifies the target UPF, if it exists, and initiates configuration of the traffic rules there. If no applicable UPF exists, the SMF can insert one or more UPFs in the data path of the PDU session.

The discussion above assumes that the MEC system, with the relevant Functional Entities supported there, are trusted by the 3GPP network and that policies allow direct access from AFs to the 5G Core Network Functions. There are cases however, where a MEC FE needs to request services from the Network Exposure Function (NEF), for example when MEC is not considered trusted, and the policy does not allow direct interaction with the 5G Core NFs. Also, whenever the request targets, or may target multiple PCFs, it should go via the NEF.

4.2 Use cases validation

This section presents the validation of the use cases, considering KPIs regarding the performance of the involved services, and others like the confusion matrix to assess the accuracy of ML models.

4.2.1 Smart City

Table 14 presents the KPIs for the different functionalities described in the Smart City (J. Navarro-Ortiz, 2020) use case (United for Sustainable Cities (U4SSC) initiative, 2019) and on a recent survey (Sewalkar & Seitz, 2019).

Table 14: KPIs for the Smart City Use Case – Pedestrian safety and Crime prevention

KPI Name	Unit of Measurement	Description
Availability	Percentage	The communication infrastructure shall be available to support the different services in the use case
Video sharing rate	Mbps	Minimum data rate for video sharing between a UE supporting V2X application and a V2X application server
Video sharing latency	Milliseconds	Maximum tolerated latency for video sharing between UEs supporting V2X application
Video sharing reliability	Percentage	Video must be shared between UEs supporting V2X application and other components with minimum reliability levels
Connection density under traffic	Vehicles/Km^2	Maximum amount of vehicles able to exchange/receive data per Km^2
IoT sharing rate	Mbps	Minimum data rate for information sharing between IoT devices and other components of the infrastructure
IoT sharing latency	Milliseconds	Maximum tolerated latency for information exchange between IoT devices and other components of the infrastructure
IoT data aggregation	Percentage	Certain IoT data flows can be aggregated at the intermediate layers (edge)
Number of IoT devices	Number	Number of IoT devices supported by an edge node
Accuracy of ML models	%	Confusion matrix of ML algorithms (True positives, True negatives, False positives, flase negatives)

The air quality use case also has a subset of metrics that are specified by the U4SSC initiative (United for Smart Sustainable Cities (U4SSC) initiative, 2019). The document (ITU, 2021) specifies a set of KPIs that can be implemented to assess how sustainable a city is. The following table summarizes a set of initial KPIs that are relevant for the Smart City use case validation.

Table 15: KPIs for the Smart City Use Case – Air quality use case

KPI Name	Unit of Measurement	Description
Air Pollution	Mass of pollutants collected (ug) per volume of air (m^3)	Air quality index based on particulate matter, Nitrogen dioxide (NO_2), Sulphur Dioxide (SO_2) and Ozone (O_3)
Noise exposure	Percentage	Number of inhabitants exposed to city levels over 55dB over the total number of city inhabitants.
Electricity Consumption	KWh / year	Measured considering the devices that are connected to the infrastructure.

Other KPIs and related metrics will be researched during the implementation activities (See Section 4.3 for further details).

4.2.2 Autonomous Driving

The Autonomous Driving use case will be validated considering KPIs specified by related standardization bodies such as ETSI working group focused on Intelligent Transport Systems (ITS) (ETSI, ETSI ITS committee, 2021). The following table summarizes some of the KPIs that are interesting to validate this use case. Such KPIs rely on the validation of ITS in 5G networks (5G Car consortium, 2019; ITU, 2021).

Table 16: KPIs for the Autonomous Driving scenario

KPI Name	Unit of Measurement	Description
Accuracy of Localization	Meters	The report of location of vehicles should be accurate and provided in quasi real time fashion
Maneuvre completion time	seconds	Time to complete a certain manoeuvre (e.g. parking, change of lane)
Accuracy of Mobility	Km/h	Mobility, speed of vehicle should be accurately reported

Availability	percentage	The communication infrastructure shall be available to support the different services in the use case
Data Rate	Mbps	The communication infrastructure shall support the minimum data rates to allow the exchange of messages in V2X contexts
Latency	seconds	The exchange of information must occur bellow certain delay
Reliability	percentage	All the messages must be communicated with reliable mechanisms
Power Consumption	Watts	The energy for communication must be within certain thresholds

Other KPIs and related metrics will be researched and defined during the implementation activities.

4.3 OREOS platform deployment and testing time-frame

The OREOS platform will be developed in a stepwise approach, according to the next deliverables time-frame, as summarized in Table 17.

Table 17: Software deployment and testing time-frame

Deliverable	Name	Month	Testing	Components deployed
E4.1	Platform components (preliminary)	M15	Installation of OpenStack Installation of Kubernetes Deployment of components UE-gNB traffic generation UE-UPF traffic generation DCAE KPI collection	ONAP OAI-RAN OAI-CN (Except NWDAF)
E5.2	Prototype (preliminary)	M17	Instantiation of a network slice Slice optimization via OOF Wireshark trace of UE ⟷ UPF-1 and UE ⟷ UPF-2 slices	
E5.3	Tests of preliminary prototype (preliminary)	M21	Evaluation of preliminary prototype. Traces, logs of components	

E4.2	Platform components (final)	M24	Deployment of components KPI collection via O1 interface NWDAF-ONAP communication MEC APP instantiation via ONAP APP transfer to different DC	NWDAF FlexRIC FlexCN Intel Open Edge
E5.4	Prototype (final)	M27	Use-cases end-to-end	
E5.5	Final tests of prototype	M29	All KPIs	

Note: Month M01 corresponds to January 2021.

5. Conclusion

The Smart City and Autonomous Driving scenarios in OREOS impose different requirements in terms of reliability, latency, the number of communicating devices, which can be provided by leveraging the support of 5G for Ultra Reliable Low Latency Communications, for mobile broadband services and for Massive Internet of Things communication. The orchestration of resources in such scenarios needs to optimize the infrastructure, by considering the specificities of each service. For instance, the detection of pedestrians in cross-roads requires analysis of video images in real time and announcing the location of the pedestrian to vehicles near that location, for safety purposes.

The OREOS platform relies on ONAP to manage all the services at the infrastructure and service levels, including also the connection with the radio access components through Open Air Interface. Key functionalities include slice management at the Radio Access Network and 5G core levels. At the RAN level, the optimization support with the *RAN Intelligent Controller* is relevant to enable efficient network slicing, as required by the OREOS use cases. ONAP also comprises support for closed loop automation, where it is possible to automate Virtual Network Function management, including scaling according to certain thresholds, or based on AI mechanisms that consider *Muti-access Edge Computing* specificities. This document provides an extensive description of the solutions to support *Muti-access Edge Computing*, by placing components at the edge and cloud, and by leveraging on the advances of open-source projects to deploy RAN components in a standard and scalable approach.

The scenarios in OREOS have been grouped into two main use cases, namely, Smart City and Autonomous Driving, detailing the required workflows to support the services associated with each use case. This document provides information on the feasibility of the OREOS project to support such scenarios, including a preliminary set of KPIs. These will drive the performance assessment of the OREOS platform, and they will also show how OREOS contributes to the innovation in the domains of Smart Cities and Intelligent Transport Systems. The use cases have been described in detailed workflows, where the different interactions between the components/tools of the OREOS platform have been identified.

This document acts as the baseline to deploy the OREOS platform, by providing step-by-step instructions to validate the components and use cases, as well as by delineating the planned activities for a successful validation of the OREOS platform.

6. Bibliography

Fortescue, P., Stark, J., & Swinerd, G. (2003). *Spacecraft Systems Engineering.* Wiley.

Wakker, K. F. (2015). *Fundamentals of Astrodynamics.* Institutional Repository Library - Delft University of Technology.

ETSI. (2017). *TS 22.185 - Service requirements for V2X services.* ETSI.

Cisco. (February de 2021). *Intent-Based Network Security At-a-Glance*. (Cisco) Obtido em June de 2021, de https://www.cisco.com/c/en/us/solutions/collateral/enterprise-networks/enterprise-network-security/intent-based-network-security-aag.html

ONAP. (January de 2017). *Security Framework*. (ONAP) Obtido de https://wiki.onap.org/display/DW/Security+Framework

ONAP. (Accessed: June 2021). *ONAP Use Cases & Requirements Portal.* URL: https://wiki.onap.org/pages/viewpage.action?pageId=99681185.

Cisco. (Accessed: June 2021). *Intent-Based Networking: Building the bridge between business and IT.* URL: https://www.cisco.com/c/en/us/solutions/enterprise-networks.

ONAP. (Accessed: June 2021). *SON use case workflow.* URL: https://wiki.onap.org/display/DW/R7+OOF+SON+Use+Case.

ONAP. (Accessed: June 2021). *IBN workflow.* URL: https://wiki.onap.org/display/DW/Intent-Based+Network.

World-Health-Organization. (2015). Global status report on road safety. Geneva, Switzerland.

OREOS. (2021). *Deliverable 2.1.*

ONAP-white-paper. (2020). *ONAP E2E Network SlicingTechnical Overview: Providing End-to-end 5G Network Slicing Capability.*

Chih-Lin, Kuklinskí, S., & Chen, T. (2020). A Perspective of O-RAN Integration with MEC, SON, and Network Slicing in the 5G Era. *IEEE Network, 34(6)*.

O-RAN. (2021). *Study on O-RAN Slicing, O-RAN Technical Report v02.00.*

O-RAN. (2020). *Study on O-RAN Slicing Architecture, O-RAN Technical Report v04.00.*

O-RAN. (2020). *Study on O-RAN Slicing Architecture, O-RAN Technical Report v04.00.*

ONAP. (12 de Dec de 2019). *Edge Automation through ONAP*. Obtido de ONAP Wiki: https://wiki.onap.org/display/DW/Edge+Automation+through+ONAP

3GPP. (2021). *TS 28.552 - Management and orchestration; 5G performance measurements.* 3GPP.

3GPP. (2021). *TS 38.300 - NR; NR and NG-RAN Overall description; Stage-2.* 3GPP.

3GPP. (2021). *TS 22.261 - Service requirements for the 5G system.* 3GPP.

3GPP. (2021). *TS 28.554 - Management and orchestration; 5G end to end Key Performance Indicators.* 3GPP.

3GPP. (2021). *TS 28.531 - Management and orchestration; Provisioning.* 3GPP.

3GPP. (2021). *TS 28.541 - Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3.* 3GPP.

Aliu, O. G., Imran, A., Imran, M. A., & Evans, B. (2013). A Survey of Self Organisation in Future Cellular Networks. *IEEE Communications Surveys & Tutorials , 15*(1), 336-361.

Zakrzewska, A., Ruepp, S., & Berger, M. S. (2014). Towards converged 5G mobile networks-challenges and current trends. *ITU kaleidoscope academic conference: Living in a converged world - Impossible without standards?* St. Petersburg, Russia.

Imran, A., Zoha, A., & Abu-Dayya, A. (2014). Challenges in 5G: how to empower SON with big data for enabling 5G. *IEEE Network , 28*(6), 27-33.

Moysen, J., & Giupponi, L. (2018). From 4G to 5G: Self-organized network management meets machine learning. *Computer Communication, 129*(1), 248-268.

Zeydan, E., & Turk, Y. (2020). Recent Advances in Intent-Based Networking: A Survey. *2020 IEEE 91st Vehicular Technology Conference.* Antwerp, Belgium.

Tsuzaki, Y., & Okabe, Y. (2017). Reactive configuration updating for Intent-Based Networking. *IEEE International Conference on Information Networking (ICOIN).* Da Nang, Vietnam.

Ujcich, B. E., Bates, A., & Sanders, W. H. (2020). Provenance for Intent-Based Networking. *6th IEEE Conference on Network Softwarization (NetSoft).* Ghent, Belgium: IEEE.

Campanella, A. (2019). Intent Based Network Operations. *IEEE Optical Fiber Communications Conference and Exhibition (OFC)* (pp. 3-7). San Diego, California, USA: IEEE.

Wei, Y., Peng, M., & Liu, Y. (2020). Intent-based networks for 6G: Insights and challenges. *Digital Communications and Networks, 6*(3), 270-288.

Organization, W. H. (2015). Global status report on road safety. Geneva, Switzerland.

Sewalkar, P., & Seitz, J. (2019). Vehicle-to-Pedestrian Communication for Vulnerable Road Users: Survey, Design Considerations, and Challenges. *Sensors, 19(2)*.

Linget, T. (2020). C-V2X Use Cases Volume II: Examples and Service Level Requirements. *5GAA Automotive Association, White paper*.

Sohaib, R. (2021). Network Slicing for Beyond 5G Systems: An Overview of theSmart Port Use Case. *Electronics 10(1090)*.

Meng, L. (2020). *E2E network slicing use case overview, technical presentation.*

Guo, C. (2021). *Overall E2E Network Slicing Modeling Design for G&H, technical presentation, LF Networking.*

Niknam, S. (2021). Intelligent O-RAN for Beyond 5G and 6G Wireless Networks. *https://arxiv.org/pdf/2005.08374.pdf*.

Shankaranayaranan, N., Swaminathan, S., & Moor, K. (2019). *5G Self-Organizing Network (SON) using ONAP Optimization Framework (OOF), Guilin Demo & Roadmap.* LF-Networking.

Yoonsu Shin, S. K. (2017). Virtualized ANR to Manage Resources for Optimization of Neighbour Cell Lists in 5G Mobile Wireless Networks. *Mobile Information Systems*.

Naqqash Dilshad, J. H. (2020). Applications and Challenges in Video Surveillance via Drone: A Brief Survey. *International Conference on Information and Communication Technology Convergence (ICTC).* Korea: IEEE.

U4SSC, I.-T. a. (2021). *United 4 Smart Sustainable Cities*. Obtido de https://www.itu.int/en/ITU-T/ssc/united/Pages/default.aspx

ITU. (2021). *ITU's implementation of the U4SSC KPIs on Smart Sustainable Cities*. Obtido de https://www.itu.int/en/ITU-T/ssc/Pages/KPIs-on-SSC.aspx

Melodia, L. B. (2020). Open, Programmable, and Virtualized 5G Networks: State-of-the-Art and the Road Ahead. *Computer Networks, 182*, 107516.

E. Zhukov, M. N. (2019). *Edge Automation –Potential Strategies for Deploying ONAP at Edge.* Obtido em December de 2021, de https://wiki.onap.org/download/attachments/28379482/Edge%20Automation%20%20Potential%20Strategies%20for%20Deploying%20ONAP_v1.8.pdf

Hairuman, A. (2019). MEC Deployment with Distributed Cloud in 4G Network for 5G Success. *6th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE).*

ETSI. (2020). *ETSI NFV&MEC Plugtests report.* Obtido em December de 2021, de https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/ETSI_NFV&MEC_2020_Plugtests_Report_v1_0_0.pdf

García-Rois, J. (2021). Evaluating management and orchestration impact on closed-loop orchestration delay. *Journal of software: practice and experience, 51*, 193-212.

Americas, 5. (2019). *A 5G Americas White Paper on management orchestration & automation.* Obtido em December de 2021, de https://www.5gamericas.org/wp-content/uploads/2019/11/Management-Orchestration-and-Automation_clean.pdf

Bhattacharjee, S., Katsalis, K., Arouk, O., Schmidt, R., Wang, T., An, X., . . . Nikaein, N. (2021). Network Slicing for TSN-Based Transport Networks. *IEEE Access, 9*(9).

Samsung. (April de 2020). *Network Slicing.* Obtido de https://images.samsung.com/is/content/samsung/p5/global/business/networks/insights/white-paper/network-slicing/200420_Samsung_Network_Slicing_Final.pdf

Huawei. (March de 2020). *Categories and Service Levels of Network Slicing.* Obtido de https://www.huawei.com/en/news/2020/3/huawei-release-network-slice-white-paper

Frangoudis, A. K. (2020). Toward Slicing-Enabled Multi-Access Edge Computing in 5G. *IEEE Network, 34*.

OREOS. (2021). *Deliverable E3.1.*

T. Taleb, P. H. (2013). Follow-Me Cloud: An OpenFlow-Based Implementation. *IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing.*

Ksentini, T. T. (2013). Follow me cloud: interworking federated clouds and distributed mobile networks. *IEEE Network, 27*(5).

D. Lake, N. W. (2021). Softwarization of 5G Networks–Implications to Open Platforms and Standardizations. *IEEE Access, 9*.

Sabella, D. (2021). Multi-access Edge Computing: Software Development at the Network Edge. *Springer*.

United for Smart Sustainable Cities (U4SSC) initiative. (2019). *Air quality management in Southern California, USA: Case study of the U4SSC City Science Application Framework.* California, USA: ITU.

5G Car consortium. (2019). *Deliverable D2.1 5GCar Scenarios, Use Cases and KPIs.* 5GCar.

ITU. (2021). *United 4 Smart Sustainable Cities*. Obtido de https://www.itu.int/en/ITU-T/ssc/united/Pages/default.aspx

United for Sustainable Cities (U4SSC) initiative. (2019). *Crime prediction for more agile policing in cities - Rio de Janeiro, Brazil: Case study of the U4SSC City Science Application Framework.* Geneva, Switzerland: ITU.

Filali, A. e. (2020). Multi-access edge computing: A survey. *IEEE Access 8*.

Tampe, A. F. (2020). Multi-access edge computing in cellular networks. *CSI Transactions on ICT, 8*, 85-92.

Jiang, J. (2020). *Optimizing Edge Computing in 5G Networks, Master thesis at Delft University of Technology in the Netherlands.*

al, V. R. (2019). Automating the deployment of 5G Network Slices using ONAP. *10th International Conference on Networks of the Future (NoF).* Rome, Italy .

OpenAirInterface. (2021). Obtido em December de 2021, de https://openairinterface.org/

UERANSIM. (2021). *UERANSIM*. Obtido de https://github.com/aligungr/UERANSIM

Alliance, O.-R. (2021). *O-RAN*. Obtido de https://www.o-ran.org/

Free5G. (2021). *Free5GC*. Obtido de https://www.free5gc.org

Intel. (2021). *OpenNess*. Obtido de https://www.openness.org/

EURECOM. (2021). *OAI CN 5G*. Obtido de https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-amf/-/wikis/home

EURECOM. (2021). Obtido de https://gitlab.eurecom.fr/oai/openairinterface5g/blob/develop/doc/FEATURE_SET.md

Foundation, O. (2020). Obtido de https://opennetworking.org/wp-content/uploads/2020/09/Jyh-Cheng-Chen-Final-Slides.pdf

ETSI. (2021). Obtido de https://www.etsi.org/technologies/multi-access-edge-computing

Silverman, B. a. (2018). *OpenStack for Architects: Design production-ready private cloud infrastructure.* Packt Publishing Ltd.

Diouf, G. E. (2020). On Byzantine fault tolerance in multi-master Kubernetes clusters. *Future Generation Computer Systems*, 109, pp.407-419.

Seittenranta, R. (2018). *Modernizing Proprietary E-commerce Platform Infrastructure.*

Pervaiz, A. (2021). *New Features in CMSWEB Kubernetes Cluster at CERN (No. CERN-STUDENTS-Note-2021-245).*

Qi, S. K. (2020). Assessing container network interface plugins: Functionality, performance, and scalability. *IEEE Transactions on Network and Service Management*, 18(1), pp.656-671.

Nascimento, M. R. (2011). Virtual routers as a service: the routeflow approach leveraging software-defined networks. *In Proceedings of the 6th International Conference on Future Internet Technologies*, (pp. 34-37).

Hochstein, L. a. (2017). *Ansible: Up and Running: Automating configuration management and deployment the easy way.* O'Reilly Media, Inc.

J. Navarro-Ortiz, P. R.-D.-M.-S. (2020). A Survey on 5G Usage Scenarios and Traffic Models. *IEEE Communications Surveys & Tutorials, vol. 22, no. 2*, pp. 905-929.

A. Appendix – Network Functions

The following descriptions were retrieved from .

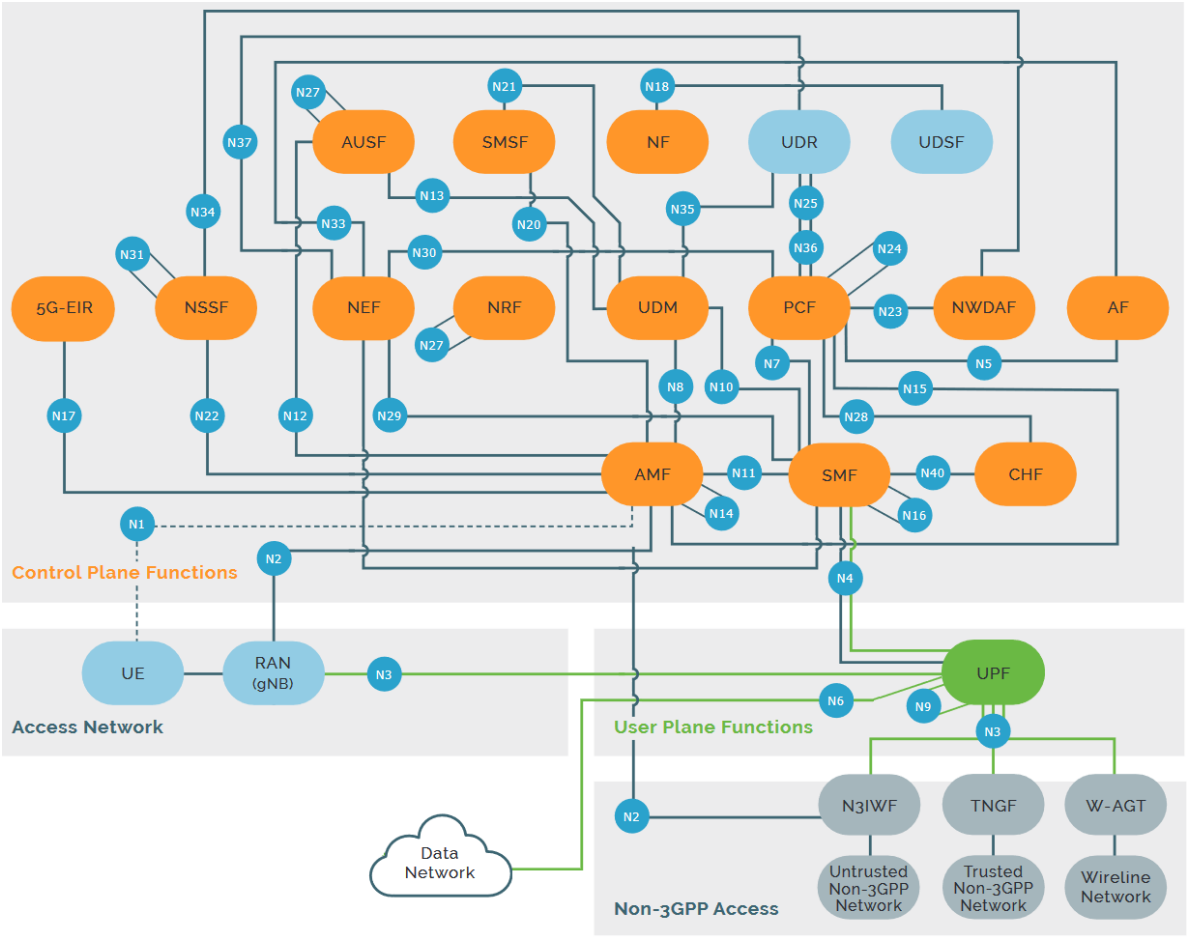

Figure 70: 5GC Network Functions

Table 18: Protocol stack of Non-3GPP interfaces

Interface	Between		3GPP Spec	Protocol Stack
MWu	UE	N3IWF	TS 23.501	CP: IP|IPsec(tunnel)|Inner IP|GRE UP: IP | IKEv2
Y1	UE	Non-3GPP Network	N/A	Not specified by 3GPP
Y2	Non-3GPP Network	N3IWF	N/A	Not specified by 3GPP

Table 19: Protocol stack of 3GPP interfaces

Interface	Between		3GPP Spec	Protocol Stack
N1	UE	AMF	TS 23.501	IP | SCTP | NG-AP | NAS
N2	(R)AN	AMF	TS 23.501	IP | SCTP | NG-AP
N3	(R)AN	UPF	TS 23.501	IP | UDP | GTP-U
N4	UPF	SMF	TS 23.501	CP: IP | UDP | PFCP UP: IP | UDP | GTP-U
N5	PCF	AF	TS 23.501	IP | TCP | HTTP2 | JSON
N6	UPF	DN	TS 23.501	IP | UDP | GTP-U
N7	SMF	PCF	TS 23.501	IP | TCP | HTTP2 | JSON
N8	AMF	UDM	TS 23.501	IP | TCP | HTTP2 | JSON
N9	UPF	UPF	TS 23.501	IP | UDP | GTP-U
N10	SMF	UDM	TS 23.501	IP | TCP | HTTP2 | JSON
N11	AMF	SMF	TS 23.501	IP | TCP | HTTP2 | JSON
N12	AMF	AUSF	TS 23.501	IP | TCP | HTTP2 | JSON
N13	AUSF	UDM	TS 23.501	IP | TCP | HTTP2 | JSON
N14	AMF	AMF	TS 23.501	IP | TCP | HTTP2 | JSON
N15	AMF	PCF	TS 23.501	IP | TCP | HTTP2 | JSON
N16	SMF	SMF	TS 23.501	IP | TCP | HTTP2 | JSON
N17	AMF	5G-EIR	TS 23.501	IP | TCP | HTTP2 | JSON
N18	NF	UDSF	TS 23.501	IP | TCP | HTTP2 | JSON
N19	PSA	PSA	TS 23.501	IP | TCP | HTTP2 | JSON
N20	AMF	SMSF	TS 23.501	IP | TCP | HTTP2 | JSON
N21	SMSF	UDM	TS 23.501	IP | TCP | HTTP2 | JSON
N22	AMF	NSSF	TS 23.501	IP | TCP | HTTP2 | JSON
N23	PCF	NWDAF	TS 23.501	IP | TCP | HTTP2 | JSON
N24	PCF	PCF	TS 23.501	IP | TCP | HTTP2 | JSON
N25	UDR	PCF	TS 29.513	IP | TCP | HTTP2 | JSON
N26	AMF	MME	TS 23.502	IP | UDP | GTP-C
N27	NRF	NRF	TS 23.501	IP | TCP | HTTP2 | JSON
N28	PCF	CHF	TS 32.240	IP | TCP | HTTP2 | JSON
N29	NEF	SMF	Depreciated IP	IP | TCP | HTTP2 | JSON
N30	NEF	PCF	TS 29.554	IP | TCP | HTTP2 | JSON
N31	NSSF	NSSF	TS 23.501	IP | TCP | HTTP2 | JSON
N32	SEPP	SEPP	TS 23.501	IP | TCP | HTTP2 | JSON

N33	NEF	AF	TS 23.501	IP | TCP | HTTP2 | JSON
N34	NSSF	NWDAF	TS 23.501	IP | TCP | HTTP2 | JSON
N35	UDM	UDR	TS 23.501	IP | TCP | HTTP2 | JSON
N36	PCF	UDR	TS 23.501	IP | TCP | HTTP2 | JSON
N37	NEF	UDR	TS 23.501	IP | TCP | HTTP2 | JSON
N40	SMF	CHF	TS 23.501	IP | TCP | HTTP2 | JSON
N50	AMF	CBCF	TS 23.501	IP | TCP | HTTP2 | JSON
N51	AMF	NEF	TS 23.501	IP | TCP | HTTP2 | JSON
N52	NEF	UDM	TS 23.501	IP | TCP | HTTP2 | JSON
N53	I-NEF	NEF	TS 23.501	IP | TCP | HTTP2 | JSON
N55	AMF	UCMF	TS 23.501	IP | TCP | HTTP2 | JSON
N56	NEF	UCMF	TS 23.501	IP | TCP | HTTP2 | JSON
N57	AF	UCMF	TS 23.501	IP | TCP | HTTP2 | JSON
N58	AF	NEF	TS 23.501	IP | TCP | HTTP2 | JSON

A.1 Access and Mobility Management Function (AMF)

Mobility can be considered the essence of a 5G system. Mobility management starts when a new connection is established between a UE and the core of the network. This action triggers a series of procedures to identify the UE, providing a security framework, in order to provide a channel for the transport of messages. The main objective of the AMF component is to ensure that the communication process takes place in a cohesive and transparent way, considering user mobility as a key factor. Through the functions implemented in the AMF, the network can, for example, reach a specific user to notify him of any messages or calls received. Furthermore, the AMF component can allow, for example, that a certain UE starts a communication process with other UEs also connected in the RAN or that have access to the Internet. Another important feature of the AMF is to ensure connectivity and that any existing sessions can be maintained as the UE moves between different access points.

In 5G networks, there is a need to provide flexible support for a wide range of new users. Many of these users have specific needs regarding mobility. For example, a certain UE used in a factory normally does not move, while the UE in an autonomous or remotely controlled vehicle may be highly mobile. To better support these different needs, the specification in Release 15 divided the mobility procedures into three categories for the AMF component:

- Common Procedures can be characterized as a set of steps that will be performed when any UE requests a connection with the core. Among these steps, the security process stands out, consisting of primary authentication, management of access keys, identification and basic configuration of the UE;
- Specific Procedures have the function of managing the registration and periodic updating of the mobility of a given UE in the AMF. Furthermore, this procedure controls the termination of registration of a UE, given a scope of different access technologies;
- Connection Management procedures are used to establish a secure communication process between a given UE and the core. Furthermore, this procedure is used when a certain UE needs to perform a network resource reservation process for sending data.

Each of these categories of procedures aims to provide functionalities that allow UEs to establish connections with the core of the network, using services associated with mobility.

A.2 Session Management Function (SMF)

The SMF component is responsible for managing the UEs sessions, that is, the sessions that represent the connected users. The main responsibilities of the SMF are associated with the activities of establishing, modifying and releasing the individual sessions of the UEs, as well as allocating the IP addresses for each connected UE. However, communication between UEs and SMF is carried out indirectly through the AMF component. It is up to the AMF to forward the messages associated with the session of a given UE and the functions of the SMF component.

The internal functions of the SMF component interact with the other VNFs of the other components through the producer/consumer model, defined according to the Service Based Architecture (SBA). For example, the SMF has the responsibility for controlling different functions associated with the UPF component. This control includes the ability of SMF to configure the direction of data traffic associated with a UPF, for a given UE session (i.e., the SMF establishes a Protocol Data Unit (PDU) session between the UE and the UPF). In addition, the SMF must perform monitoring and control actions on the UPF. The SMF component also interacts with functions associated with the PCF, with the objective of executing the policy of the sessions of the connected UEs. It is this action that determines the guidelines for providing data connectivity between a UE and the data network (DN - Data Network): the establishment of the PDU session, as well as its characteristics, depends on the UE requirements, the information from the UDM/UDR databases, and the service and QoS policies configured in the PCF.

A.3 User Plane Function (UPF)

3GPP Release 15 characterizes the UPF component as a key function within the new SBA architecture. The UPF can be seen as part of the process of separating Control Plan and Data Plan, initially introduced in Release 14 with CUPS. The decoupling of data and control allows the SBA architecture to further decentralize its

components. For example, it is possible to direct activities such as packet processing to be positioned closer to the edge of the network, increasing the QoS for the user and reducing network traffic.

The functionalities implemented in the UPF component are controlled by SMF. The main function of the UPF is associated with the routing and processing of data from UEs. This component is also responsible for generating notifications associated with data traffic and for the packet inspection process. The UPF also works as a stable anchoring point between the core and any external networks. The UPF allows communication to happen in a transparent way, hiding aspects of complexity associated with mobility. IP packets destined for a certain UE are forwarded (from the Internet) to the respective UPF, which is serving that UE, even when the UE is in mobility. Generally speaking, the UPF component is responsible for:

- Play the role of anchoring between the core and the external networks;
- Act as an external access point for PDUs (Protocol Data Unit), interconnecting different data networks;
- Perform packet routing/forwarding, in addition to inspecting packets in order to detect application characteristics;
- Apply the definitions associated with the management of the user's data plan, as well as providing information on data traffic.

3GPP TS 23.501 states parameters that may be considered by the SMF for the UPF selection.

A.4 Authentication Server Function (AUSF)

AUSF is responsible for the service that performs authentication of UEs through the access credentials provided by UDM. In addition, AUSF provides services associated with encryption to enable secure information traffic and allow the execution of displacement information update processes (roaming), as well as other parameters associated with the UE.

In general, the services provided by the AUSF component are consumed by the AMF functions, which request resources associated with the authentication process. Requests for the functions of the AMF component are processed internally by the AUSF and, later, delegated to services provided by the UDM component, for the execution of registration procedures in the data repository.

A.5 Unified Data Management (UDM)

The UDM is the component responsible for managing the data of users on the network in a single centralized element. The UDM is equivalent to the HSS of the EPC/4G core. Through the UDM, several VNFs of the SBA architecture are able to perform different actions, such as: registration and authentication of UEs, identification of users, application of access and authorization policies, among others. The UDM component interacts directly with the AMF that forwards requests from the other components. Furthermore, in scenarios where there is more than one instance of the AMF component in the network, the UDM must control which instances will be responsible for servicing a specific UE. Among the features of UDM, the following stand out:

- Generation of 3GPP AKA authentication credentials (Authentication and Key Agreement);
- User identification treatment;
- Privacy-protected signature identifier hiding support;
- Authorization of access based on subscription data (e.g., restrictions associated with mobility);
- Subscription management;
- SMS management.

The UDM component acts as the front end for the user's subscription data that is stored in the UDR (Unified Data Repository). The UDM uses this signature data to execute various application logic, such as access authorization, record management, and accessibility for event termination. UDR is a database where various types of data are stored and whose access is offered as a service to other components such as UDM, PCF and NEF. There is also an optional storage component called UDSF (Unstructured data storage function), which allows other components (or functions) to store dynamic context data outside of the function (or component) itself. In the 3GPP context, unstructured data refers to data whose structure is not defined in the specifications, allowing each vendor to use a UDSF and choose their own specific structure for storage. There is no requirement for any compatibility of accessing or storing UDSF data from different vendors.

A.6 Unified Data Repository (UDR)

The UDR stores and provides access to enrollment data for the UDM, policy data for the PCF, and structured data for exposure to the NEF.

A.7 Unstructured Data Storage Function (UDSF)

In present broadband cellular network, all NFs are stateful, in which data related to user/connection/association is stored inside network elements itself. However, stateful NFs cause multiple connection related problems in case of its failure and it needs a separate standby network to continue the process, which increases maintenance cost and reduces the reliability of network system. UDSF was introduced by 3GPP to resolve data storage issue with stateful Network Functions, by being implemented with stateless principle to create a distributed system that can store structured and unstructured data related to NF on UDSF instead of in Network Functions, and by this mean, improving the reliability of the network system.

A.8 5G-Equipment Identity Register (5G-EIR)

5G-EIR is an optional network function that supports the functionality of checking the status of Equipment's identity (e.g., to check that it has not been blacklisted).

A.9 Policy Control Function (PCF)

This component performs the same function as the EPC PCRF in the 4G system. The PCF is responsible for controlling the behavior of the network, applying security and control rules related to session management, especially those features associated with user mobility. It interacts directly with the AMF, aiming to provide an access and mobility policy that may include, among other things, the management of access restrictions to services in a given area, as well as the management of issues associated with the priority of access to the media to certain UEs to the detriment of others (RFSP - Radio Frequency Selection Priority).

In the context of session management, PCF can interact with application functions and with SMF. The main objective is to provide metrics associated with quality of service, as well as information regarding data flow, which are obtained by regularly monitoring events associated with the PDU session. PCF also provides security policy information for UEs. These policies can be associated with network selection resources and rules for resource slicing selection. The PCF can be activated, for example, to provide information when a certain device performs an access selection (UE access selection) or when a PDU session is established. The interaction between the PCF and the other application functions are implemented through the exposure of six services, namely:

- PCF-AM-PolicyControl – provides information related to access control policies, network selection, mobility management and guidelines that can be applied in the selection of routes between UEs and AMFs;
- PCF-PolicyAuthorization – provides authorization and provides access control policies to a request from an AF element, related to the session PDU to which the AF is linked;
- PCF-SM-PolicyControl – provides an SMF component with access policies related to the session PDU to which the component is bound;
- PCF-BDT-PolicyControl – provides the NEF element with a set of guidelines that can be used by applications to perform data transfer in the background;
- PCF-UE-PolicyControl – provides control guidelines that can be used to manage the communication process between UEs and other network functions;
- PCF-Event-Exposure – allows other network functions to subscribe to be notified when a certain event happens.

Decisions about the application of the monitoring and control policies made by the PCF are informed, in part, by analytical information provided by other network functions, such as the NWADF. PCF is also a critically important component in a scenario where an AF needs to perform a specific activity, for example, data transfer in the background. In this case, the AF can contact the PCF in order to infer the best time interval for carrying out the activity. This enables the system operator to make information available to application providers about the most appropriate time to transfer data in the background.

A.10 Network Repository Function (NRF)

The NRF component is the repository where all functions available for a given network are listed. The purpose of this component is to allow other VNFs to find the proper function to meet their requirements. The NRF has the responsibility to select the most suitable service provider component based on the performance criteria provided. In this way, the NRF component is updated whenever a new VNF instance is deployed or modified. In addition, the NRF holds information about the other VNFs, such as type, capacity, address, among others. Within the SBA scope, the NRF component plays a fundamental role in the functioning of the other VNFs. This component provides a central mechanism that is capable of automating the entire configuration process necessary for the other VNFs to be able to discover and connect to other specialized services.

A.11 Network Slice Selection Function (NSSF)

The 5G core architecture defines the NSSF component as being responsible for managing the available network slice instances. It is responsible for selecting the instances of network slices, along with the set of AMFs available for a given UE (AMF can be a component dedicated to a specific slice or serve a set of network slice instances). The role of NSSF is to assist AMF in choosing the available network slices and redirecting traffic between the controlled network slices. NSSF can be seen as an orchestrator, capable of influencing the way network traffic is routed. It produces two services, a selection service which produces information about the selected network slice and another availability service which produces information about the available resource slices.

A.12 Network Exposure Function (NEF)

The NEF component is responsible for exposing some internal events, related to UEs and SBA architecture. The exhibition of these events aims to meet the demand of specific applications and VNFs of other components. For example, these demands may need to have access to the location of a certain UE or be notified about the interruption of connectivity of a certain equipment. In addition, this information can allow the AMF component to adjust the system according to the behaviour of a group of users.

The possibility of exposing internal events through an NEF access interface opens up new business opportunities for service providers, allowing in some cases more advanced services to be offered by third parties. For example, an application can use the services exposed by the NEF component to find out whether or not a particular UE is reachable, in addition to determining the geographic location of that UE or knowing if it is in motion. The NEF component anticipates the requests of the different VNFs through regular interactions between the UDM and AMF components.

A.13 Network Data Analytics Function **(NWDAF)**

NWDAF is an optional component in the core architecture and is responsible for collecting different types of information from the network and its respective users. This information is then organized and analyzed in order to provide the inferred results for other network functions.

The data collected by NWDAF comes from several other network functions that make up the core. Data collection is performed through the service layer that interconnects the core components via a writing service. This service can be triggered by internal events triggered by each component. The NWDAF also collects information about system health as well as information log data in the UDR component.

As per the specification, the services provided by NWDAF can be consumed by any other core component. External access is also possible through the use of the NEF component mechanism. The analysis performed by the NWDAF on data collected over time can be used as a historical/statistical resource to predict future values (Machine Learning) in order to apply certain actions in the context of the network.

A.14 Application Function **(AF)**

AF is a generic component that represents a possible application, internal or external to the operator's network, which interacts with the SBA architecture.

An important factor that must be evaluated by the system operator is the degree of confidence that an AF can have to interact directly with certain VNFs. For example, an AF with higher reliability (trusted) could directly access VNFs from all components of the SBA architecture[11], while a less reliable AF (untrusted[12]) would first have to interact with the NEF component before having access to more sensitive network functions.

Probably the best example of the usefulness of the AF is regarding the MEC, in which the MEC platform acts as an Application Function (AF) for 5G (Tampe, 2020). If the AF is trusted, it interacts with PCF by sending MEC application details like its routing information and data network name. PCF converts AF requests into policies and store it into the UDM repository. PCF notifies the policy creation event to SMF. SMF will use this information to add new traffic steering rules in selected available UPF. If UPF is not available at a particular location, based on QoS and location constraints, SMF creates a new instance of UPF and adds new traffic rules so that the UE traffic route to MEC application locally:

[11] A trusted AF can interact directly with the PCF component, influencing QoS aspects and, consequently, charging policies.
[12] For example, if deployed by a third-party application provider.

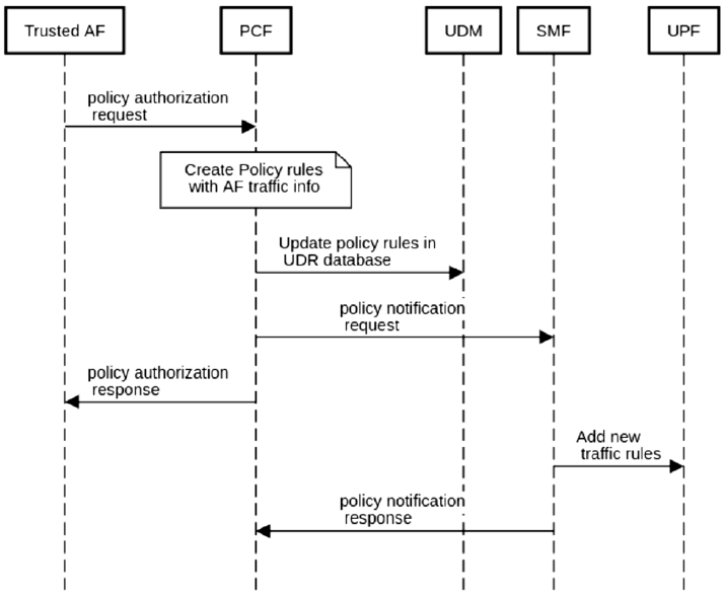

Figure 71: Sequence diagram of a trusted AF

If AF is un-trusted, then it interacts with NEF for traffic influence. After authorization of AF, NEF sends received traffic influence details to UDR. If SMF is subscribed to UDR notifications, then SMF gets a new traffic update from UDR. As per data received, it adds new traffic rules in the available UPF or newly created UPF. During the PDU session establishment, SMF provides the IP address of respective UPFs for routing traffic locally to the MEC applications:

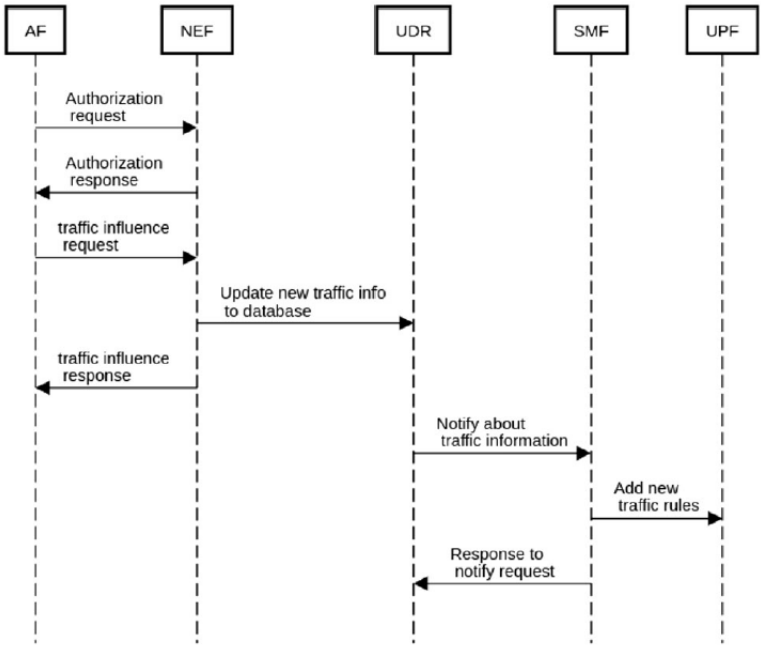

Figure 72: Sequence diagram of an untrusted AF

A.15 Non-3GPP InterWorking Function (N3IWF)

The N3IWF component is used to integrate non-3GPP accesses with the 5G core. Wi-Fi (IEEE 802.11) and DOCSIS (Data Over Cable Service Interface Specification) are examples of non-3GPP access technologies provided for integration by the standard. Conventional 3GPP access uses a base station, for example, eNB (4G) or gNB (5G). However, non-3GPP access starts on a different device, for example, a Wi-Fi access point or an HFC (Hybrid Fiber-Coaxial) modem. This device uses the N3IWF component to access the 3GPP network, including the other 5G core components.

All traffic from the N3IWF component is transmitted over secure channels and is isolated from all other 3GPP traffic. Isolation is maintained not only for data traffic (as is common for 3GPP communications), but also for control traffic, including what occurs before the authentication process. The connection between the UE and the CN is made through the establishment of IKEv2 (Internet Key Exchange) and IPsec (IP security protocol) tunnels.

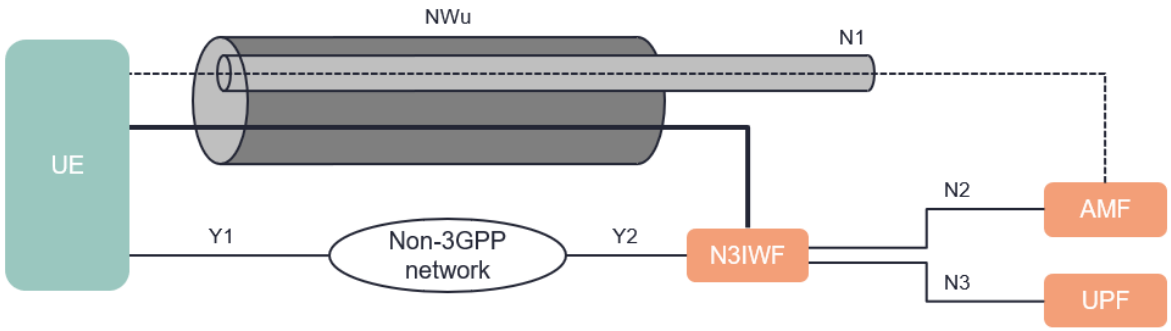

Figure 73: N3IWF

B. Appendix – Multi-Edge networking

When the UE moves out of the serving area of its current DC or there is a change of UPF, a relocation be required in order to continue the MEC services.

Two different types of MEC applications exist:

- Dedicated MEC application: like the name implies, the app instance is dedicated to a specific UE;
- Shared MEC application: in this case, the app instance serves multiple UEs.

These apps can be further divided into two types:

- Stateless: this type of app does not record data about UE for use in the next service session;
- Stateful: this type of app stores information which can be used to facilitate service continuity during the session transition.

Depending on the scenario, if the UE changes its serving DC, the app instance may be relocated to the new DC and likewise the user context:

Table 20: Multi-Edge possible application configurations

App scope	State	Events
Dedicated	Stateless	App instance relocation
	Stateful	User context transfer and App instance relocation
Shared	Stateless	App instance relocation (conditional)
	Stateful	User context transfer

The following figure illustrates a sequence diagram of the events for the share-stateful scenario (which is the same of the Autonomous Driving use-case):

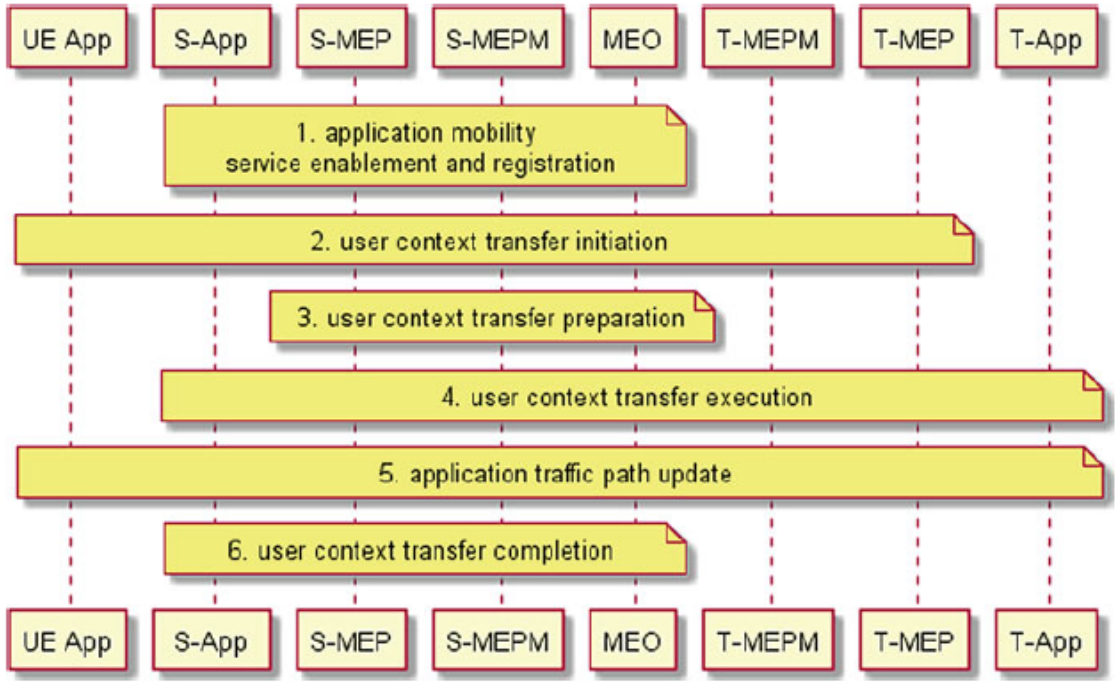

Figure 74: Multi-Edge APP mobility

The following figure illustrates the MEC relocation procedure (Jiang, 2020):

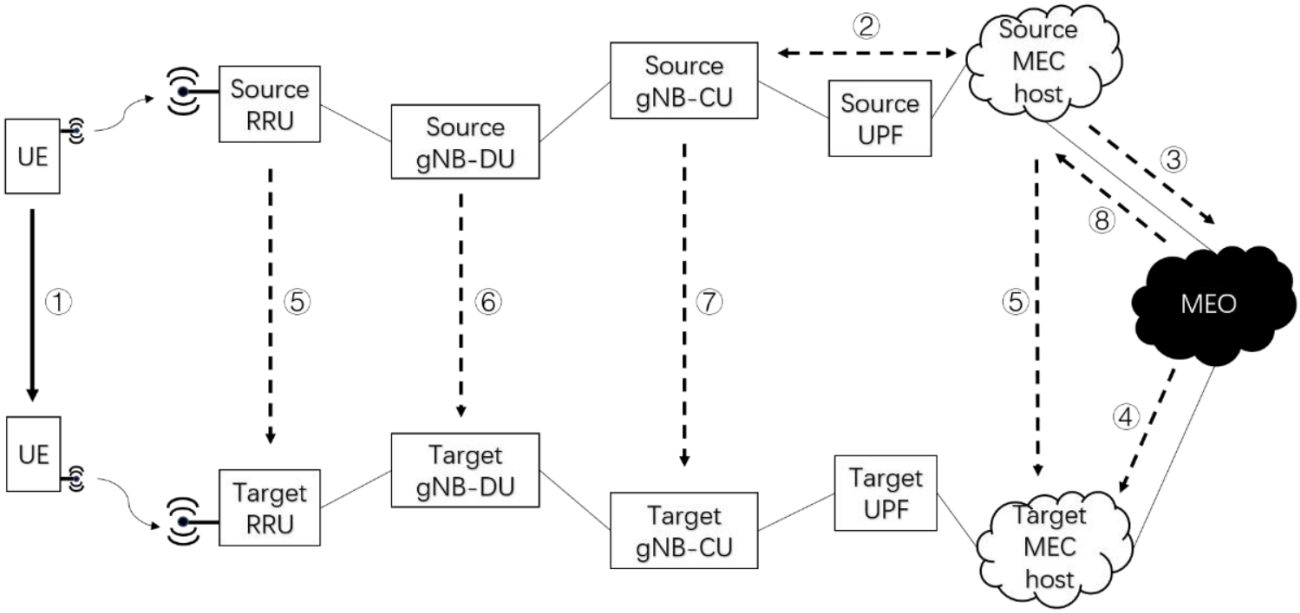

Figure 75: MEC relocation procedure

1. A UE moves to the coverage of a new site (antenna & RRU) and this new site starts to serve this UE;

2. The current MEC host (source MEC host) of this UE detects this change[13];
3. The source MEC host saves the current state for the UE if necessary, and sends a relocation request to the MEO (ONAP's NFVO);
4. The MEO selects a MEC host for this UE and instructs the selected MEC host to instantiate a new MEC application instance if necessary;
5. The source MEC host sends the user context to the target MEC host; the source gNodeB sends all access signalling, session management signalling and payload signalling to the target gNodeB; the source RRU or the target gNB-DU sends all access signalling, session management signalling and payload signalling to the target RRU;
6. If the source RRU and the target RRU are not connected to the same gNB-DU, then there will be a change in gNB-DU;
7. If the source gNB-DU and the target gNB-DU are not connected to the same gNB-CU, then there will be a change in gNB-CU;
8. The MEO instructs the source MEC host to terminate the relevant MEC application instance if necessary.

[13] To detect the UE mobility, the AF subscribes to relevant event notifications provided by the SMF, via NEF.

C. Appendix – Slice Instances & Templates

The key concepts of slice terminology as defined by 3GPP in TS 28.530 & TS 28.531 is illustrated in the following figure:

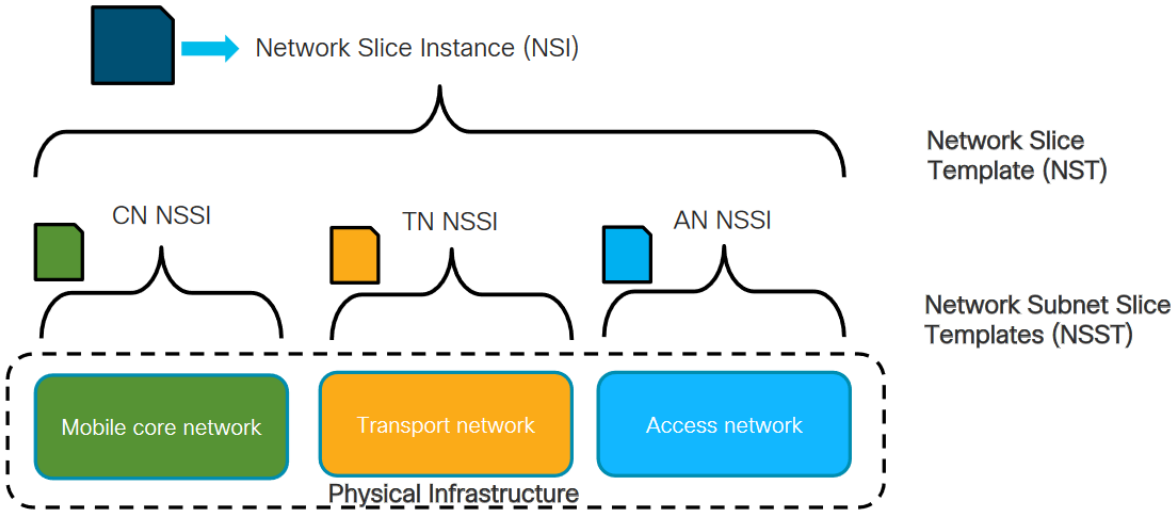

Figure 76: Network slice instances and templates

Note that a slice can be of type:

1:1:1 (AN/CN/TN)

1:n:m (AN/CN/TN)

Which means that it is possible to have Multiple NSSIs per NSI, as per illustrated in the following figure (al V. R., 2019):

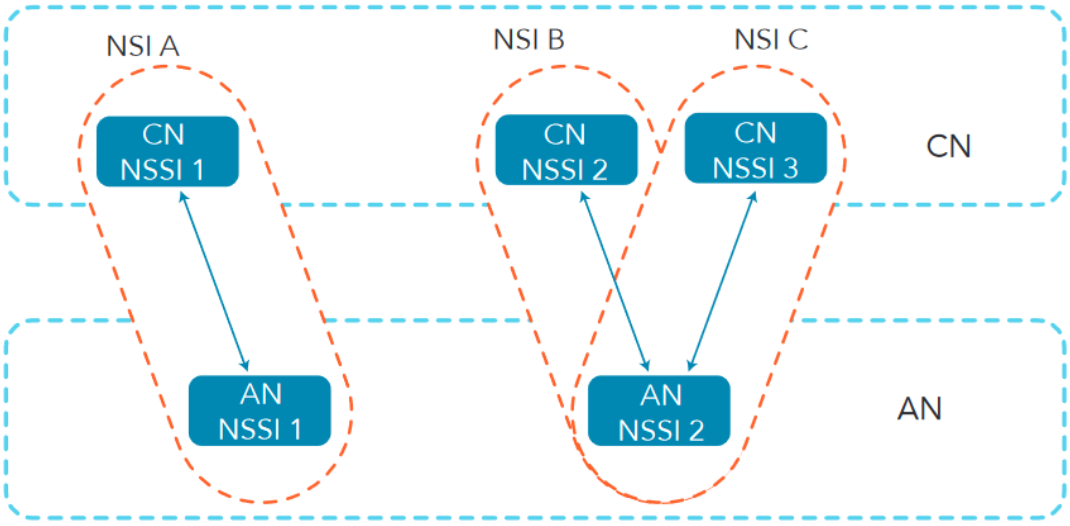

Figure 77: Network slice instances in CN and CN

The next subsections will describe the following four main components:

- Network Slice Template (NST);
- Network Slice Subnet Template (NSST);
- Network Slice Instance (NSI);
- Network Slice Subnet Instance (NSSI).

Their relationships to CSMF, NSMF and NSSMF are as follows:

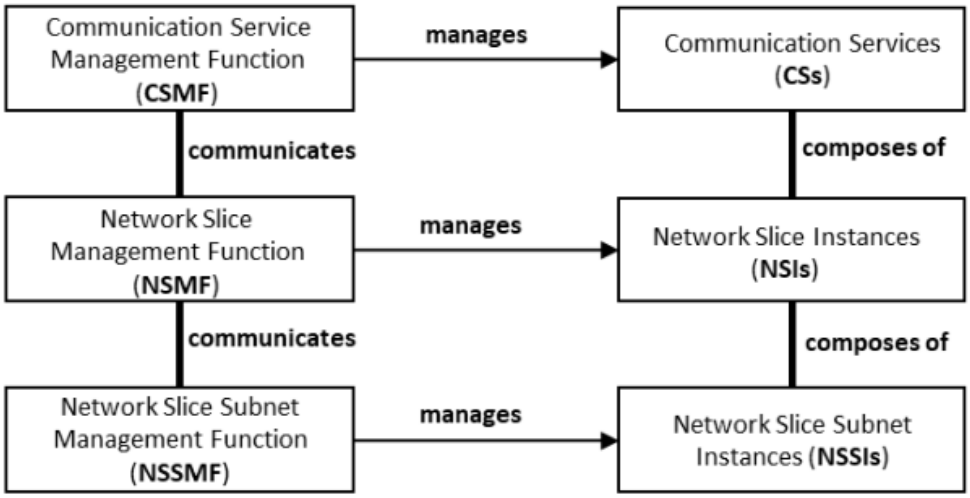

Figure 78: Relationship of network slice instances to management functions

C.1 Network Slice Template

The NST is a network slice descriptor with set of attributes that characterize the network slice. It is used by the Slice Orchestrator as the baseline to manage the lifecycle of the NSIs created starting from it.

The NST includes:

- Service Profile CBA and required policies;
- The network slices that can created from the NST, via the SST value (the Slice Service Type, as defined by 3GPP, when requesting the creation of the NSI);
- The list of NSSTs associated to the NST;
- The list of slice access points for the NST, i.e., the logical endpoints that allow to access or interconnect the network slices created from the NST.

C.2 Network Slice Subnet Template

The NSST details the capabilities and requirements of the individual components of the network slices. It is by the Slice Orchestrator as the baseline to manage the lifecycle of the NSSIs created starting from it.

The set of NSSTs associated to an NST, regulates the logic for the management of network slices. Network Slice Subnets (NSS) can be implemented as NFV network services when the related network functions are provisioned in the virtualized infrastructure. For this, the NSST can include information about Network Service Descriptors (NSD)[14] to be used to implement the given network slice subnet.

The NSST includes:

- Slice Profile CBA;
- Required policies;
- Identification of the type of NSS that can be created from the NSST, e.g., RAN slice subnet, core slice subnet, etc.;
- The list of logical endpoints associated to the network slices created from the NST;
- A description of the maximum performance capabilities that NSS created from the NSST can support. It is a list of objects that depends on the "sst" attribute[15] of the related NST. In particular:
    - If SST is eMBB, it is a list of eMBBPerfReq objects (ref. 3GPP 28.541), which include among the other attributes maximum uplink and downlink data rate, maximum user density;
    - If SST is uRLLC, it is a list of uRLLCPerfReq objects (ref. 3GPP 28.541), which include among the other attributes maximum end-to-end latency, maximum jitter, maximum expected data rate;
    - If SST is mMTC, it is a list of mMTCPerfReq objects;
- Identification of the maximum number of UEs that simultaneously can access network slice subnets created from the NSST;
- Information on whether the resources of network slice subnets created from the NSST can be shared with other subnets;
- A summary[16] of the NFV NSD composed by:
    - A Network Service Descriptor identifier;
    - A Network Service Descriptor name;
    - A Network Service Descriptor type that identifies the type of service implemented through the Network Service (e.g., vEPC, eHealth, etc.);
- List of QoS and application metrics that the network slice subnets created from the NSST can provide through monitoring.

[14] The Network Service Descriptor would include the Transport Networks and/or VNFs/CNFs/PNFs
[15] This attribute is inherited from the "perfReq" attribute defined in the 3GPP TS 28.541
[16] Only if the network slice subnet is related to a NFV Network Service

C.3 Network Slice Instance

The NSI represents a provisioned network slice in a given domain that has been instantiated based on an existing NST. The NSI includes:

- An operational state that indicates whether the slice resources are actually provisioned and working;
- An indication of the permission to use the NSI;
- A list of ServiceProfile objects which identify the NSI requirements and runtime attributes;
- The identifier of the NSSI associated to this NSI;
- The identifier of the NST used to provision the NSI;
- The unique identifier of the Service Profile object describing the NSI requirements;
- A description of the maximum performance capabilities that NSS created from the NSST can support. It is a list of objects that depends on the "sst" attribute of the related NST. In particular:
    - If SST is eMBB, it is a list of eMBBPerfReq objects (ref. 3GPP 28.541), which include among the other attributes maximum uplink and downlink data rate, maximum user density;
    - If SST is uRLLC, it is a list of uRLLCPerfReq objects (ref. 3GPP 28.541), which include among the other attributes maximum end-to-end latency, maximum jitter, maximum expected data rate;
    - If SST is mMTC, it is a list of mMTCPerfReq objects;
- Identification of the maximum number of UEs that simultaneously can access network slice subnets created from the NSST;
- Identification of the packet transmission latency (in millisecond) through the RAN, core, and backhaul segments of the NSI;
- Identification of the the mobility level of UEs accessing the NSI. It can be: stationary, nomadic, restricted mobility, fully mobility;
- Identification of whether the NSI resources can be shared with other NSIs. It can be: shared, not-shared;
- Identification of the availability requirement for the NSI, expressed as a percentage;
- The NSSAI list.

C.4 Network Slice Subnet Instance

The NSSI represents a constituent logical component of an NSI. Each NSI has at least one NSSI that actually implements the network slice itself.

An NSSI is composed by a set of Managed Functions (which can be VNFs and PNFs) and optionally an NFV Network Service, and it is characterized by a Slice Profile (which mostly provides the reference to the performance requirements).

The NSSI includes:

- A list of Managed Functions (i.e., VNFs and PNFs) instances identifiers that are associated with the NSSI. Some of them may have been reused from other NSSIs;
- A list of identifiers of NSSIs which are associated to this NSSI. It is used when an NSI is composed by multiple NSSIs. If an NSI is built by a single NSSI this list is empty;
- An operational state that indicates whether the slice subnet resources are actually provisioned and working;
- An indication of the permission to use the NSSI;
- A list of network and application KPIs that are monitored for the NSSI;
- A description of the maximum performance capabilities that NSS created from the NSST can support. It is a list of objects that depends on the "sst" attribute of the related NST. In particular:
    - If SST is eMBB, it is a list of eMBBPerfReq objects (ref. 3GPP 28.541), which include among the other attributes maximum uplink and downlink data rate, maximum user density;
    - If SST is uRLLC, it is a list of uRLLCPerfReq objects (ref. 3GPP 28.541), which include among the other attributes maximum end-to-end latency, maximum jitter, maximum expected data rate;
    - If SST is mMTC, it is a list of mMTCPerfReq objects;
- Identification of the maximum number of UEs that simultaneously can access network slice subnets created from the NSST;
- Identification of the packet transmission latency (in millisecond) through the RAN, core, and backhaul segments of the NSI;
- Identification of the the mobility level of UEs accessing the NSI. It can be: stationary, nomadic, restricted mobility, fully mobility.

C.5 Networking slicing in DCAE, OOF components

C.5.1 DCAE

The Slice Analysis MC component is introduced in ONAP for three main objectives: 1) analysing FM/PM data and KPI data of the different slices instances and the services catered by them; 2) determining and triggering the proper control loop actions; and 3) receiving recommendations for closed loop actions from analytics engines (e.g., performing validity checks) (ONAP, ONAP Documentation - Slice Analysis MS, (ONAP, ONAP Documentation - Slice Analysis MS, . The internal architecture of the Slice Analysis MS module is depicted in

Figure 79, taken from (ONAP, ONAP Documentation - Slice Analysis MS, (ONAP, ONAP Documentation - Slice Analysis MS, (ONAP, ONAP Documentation - Slice Analysis MS, . There is a DMaaP interface that communicates the Slice Analysis MS module with the Policy and VES-Collector, and a REST interface for the Config DB. An additional DMaaP interface is used to receive recommendations for closed loop updates. As depicted in Figure 79 , the DMaaP creates a call to the DMaaP client (1) which creates a thread pool, each thread is related to a DMaaP topic consumer and polls the topic periodically, when a message is received it is stored in the Postgres DB (2). The PM thread (a) reads the events from the DB (3) is put in the internal queue (b) in the needed format for further processing. The Consumer Thread (c) consumes PM samples from the internal queue (b) and makes any necessary call to the Config DB, performs required analysis, and puts the message in the proper DMaaP topic.

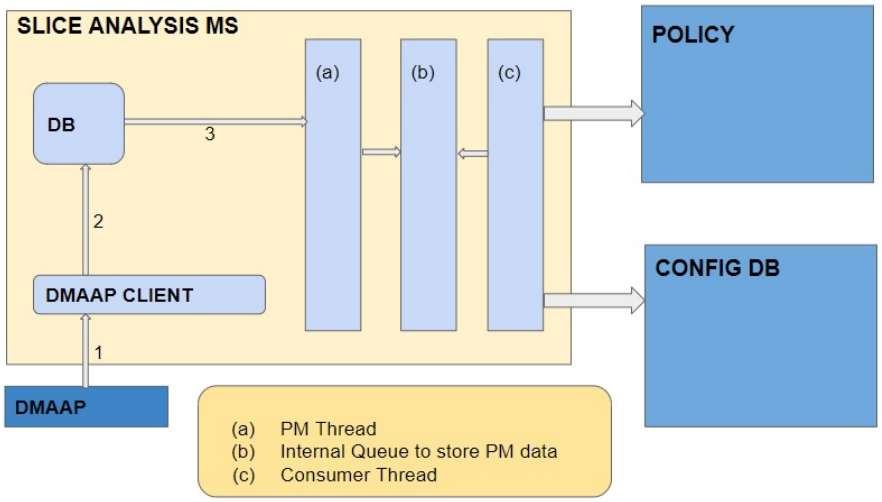

Figure 79: Slice Analysis MS architecture

The Slice Analysis MS performs two main functions, the first one is depicted in Figure 80 (ONAP-white-paper, 2020). The End-to-End slicing is performed within the following sequence:

1- The PM Mapper receives data from the RAN components. This data includes Downlink and Uplink PRBs for the data traffic in each S-NSSAI;

2- The Slice Analysis MS analyses the data received and determines if updates are required to the RAN components;

3,4,5- The control loop is activated by interacting with the DMaaP for policy updates;

6- The interface with the SDN-C (SDN-R) is used to apply the configurations at the RAN components;

7- The SDN-C is also responsible to synchronize in the Database the policies corresponding to the updates;

8- The policies are applied in the RAN components.

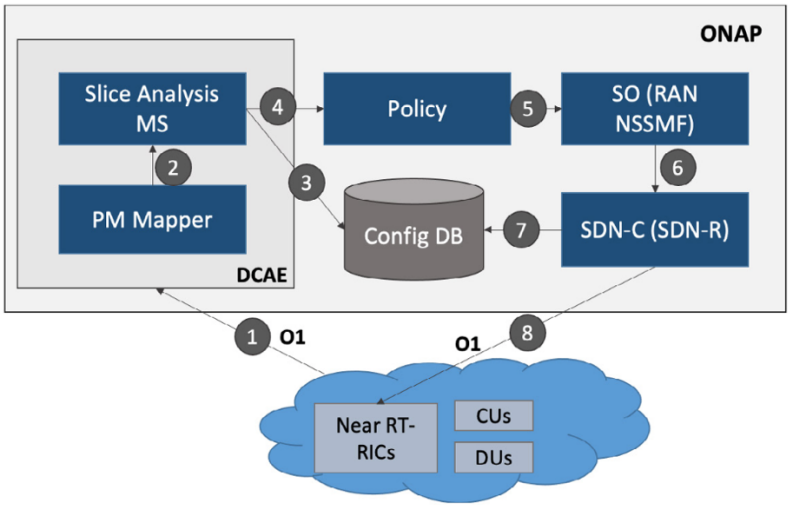

Figure 80: DCAE Closed loop flow

The second function of the Slice Analysis MS, which differs slightly from the previous one, incorporates ML models inside the DCAE module. This is described in Figure 81, taken from (ONAP-white-paper, 2020).

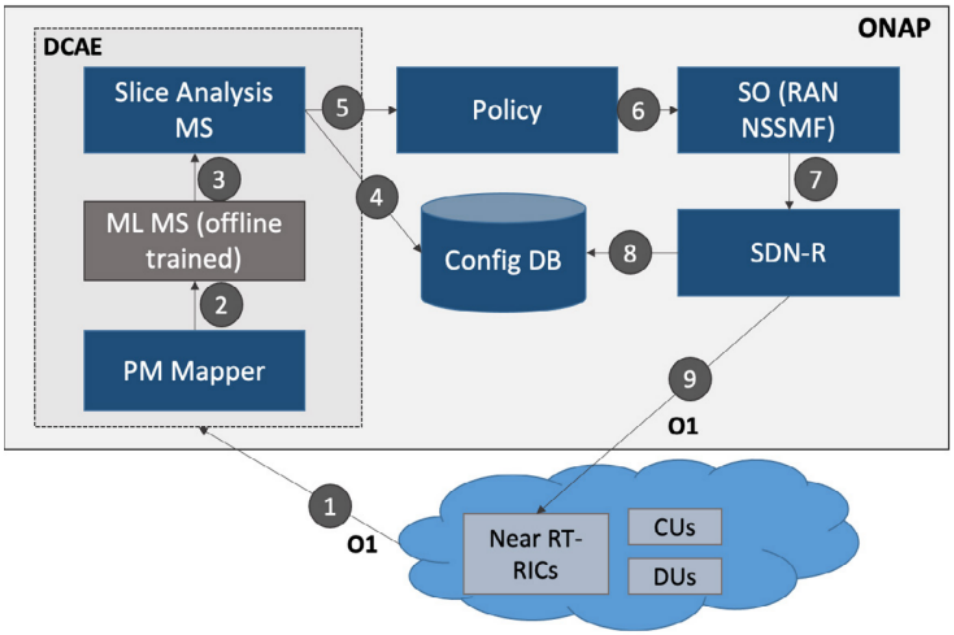

Figure 81: DCAE steps in ML-based Closed loop flow

The End-to-End slicing is performed following the next sequence:

1 - The PM Mapper receives data from the RAN components. This data includes Downlink and Uplink PRBs for the data traffic in each S-NSSAI;

2 - An offline trained ML model, included as a ML micro-service, generates configuration updates for the maximum number of connections to be supported by each cell for every S-NSSAI;

3 - The Slice Analysis MS analyses the data received and determines if updates are required to the RAN components;

4, 5, 6 - A DMaaP message with configuration updates is sent to Policy to activate the control loop;

7 - The interface with the SDN-C (SDN-R) is used to apply the configurations at the RAN components;

8 - The SDN-R synchronizes the policies corresponding to the updates in the Database;

9 - The policies are applied in the RAN components.

Figure 82 (adapted from (ONAP, ONAP Documentation - Slice Analysis MS,) shows the message flow between the different ONAP components. An AAI event triggers the creation of an NSI message to the Slice Analysis MS (SAMS), that waits for the PM data. The *Files Ready* event is published in DMaaP by the VES Collector and consumed by the DFC. Later on, the data corresponding to the event is published in the DMaaP Data Router by DFC and consumed by the PM Mapper, which publishes the PM data to be consumed by the Slice Analysis MS (SAMS), that in turn analyses the data and triggers the control loop in the Policy module. Finally, the new RAN configuration is uploaded to the simulator (ONAP, ONAP Documentation - Slice Analysis MS, .

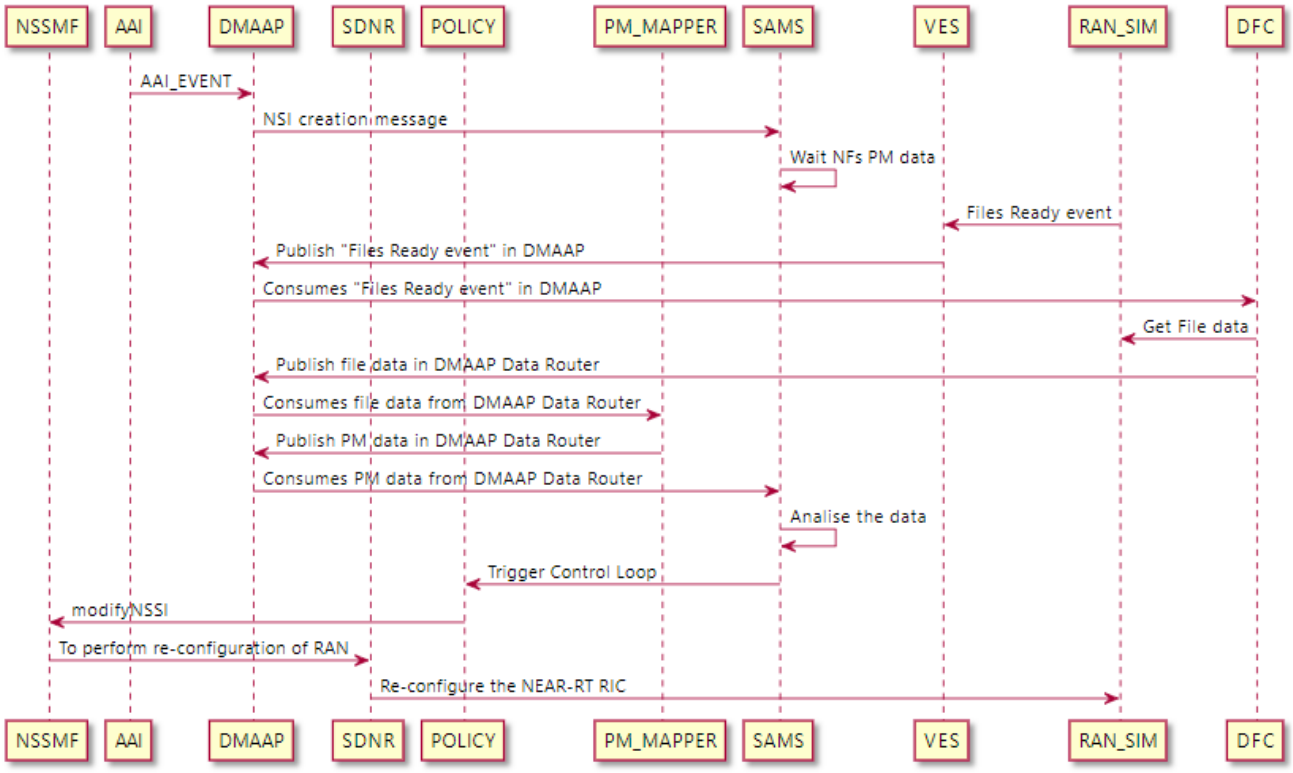

Figure 82: DCAE Slicing Closed Loop Message Flow

C.5.2 SO

Figure depicts the run-time activities for instantiating a new service that operates on a network slice. There are different Business Process Models and Notation processes for Communication Service Management Function (CSMF), Network Slice Management Function (NSMF), and multiple Network Slice Sub-net Management Functions (NSSMFs) in SO. The CSMF workflow takes the service request that arrives through the CSMF portal (UUI) and stores the order information into an AAI communication service instance. The CSMF process sends a network slice request to the NSMF workflow, producing the service profile, Network Slice Instances NSI, and NSSI. A service profile is a logical notion that exists exclusively in AAI—it comprises two AAI instances, one of which is a profile instance that will retain the slice parameters and the other of which will be used to arrange the NSI. In AAI, NSI is also a service instance that will be utilized to organize NSSI. NSSI is the real entity that will be generated by NSSMF, as well as an AAI service instance to represent NSSI in an ONAP environment. Both NSI and NSSI can be shared.

SO queries OOF to pick a slice template, followed by NSI and NSSI selection. OOF may return an existing slice instance or propose that SO construct a new slice instance in response to a slice instance selection question. A new procedure called Orchestration Task is established in UUI to manage NSI&NSSI selection recalibration with manual involvement by the operator via the NSMF Portal. For NSSI orchestration, an NSSMF adaptor in SO interfaces with internal/external NSSMFs.

The NSSMF functionality includes a common component that responds to SO subnet capability queries (NSMF) and executes domain (RAN/Core/TN) specific NSSMF functions. The domain-specific NSSMF processes carry out the appropriate steps for creating/updating the NSSI based on OOF feedback (for NSSI creation/reuse).

SO also offers service activation, deactivation, and termination functionality, which is handled by the SO (CSMF), SO (NSMF), and SO (NSSMF) (ONAP, ONAP template design for E2E slicing, .

C.5.3 OOF

According to the previous section, SO calls OOF to choose the Network Slice Template (NST) and the Network Slice Instances (NSI)/ Network Slice Subnet Instance (NSSI). In the event of NSI selection, as previously mentioned, OOF may return:

- Existing NSI, if the service request indicates it is shared, and if a suitable NSI exists;
- Slice Profiles if the request specifies that it is not shareable or no appropriate NSI exists. The Slice Profiles for the RAN, Core, and TN subnets must be produced in line with the Service Profile and the capabilities of the relevant subnet;

In the event of NSSI selection, OOF may return:

- Existing NSSI, if the service request indicates that it is shareable and if a suitable NSSI exists;
- Basic Slice Profiles or no solution.

Furthermore, OOF is utilized to evaluate if an NSI/NSSI must be terminated when deallocated for a service. When a new slice is created, SO needs to know the network slice template by which the slice needs to be created. SO will request OOF to suggest an appropriate Network Slice Template (NST) for the requirement which is provided by the end user which is a part SO request input. Figure 83 represents OOF message flows (ONAP-white-paper, 2020):

1. SO sends to OOF a select NST request;
2. The NST model information is returned to SO via OOF;
3. If the slice creation request being processed may share an existing NSI, SO sends a request to OOF to recommend one;
4. If an NSI ID exists, OOF will answer with it. Otherwise, an empty string is returned;
5. If OOF does not offer an appropriate NSI, SO should construct the NSI on its own and then ask OOF to recommend an NSSI that may be shared (ONAP, ONAP template design for E2E slicing, .

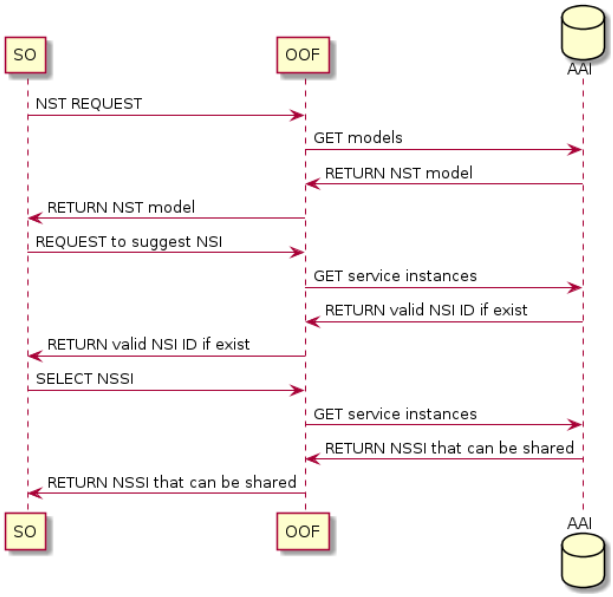

Figure 83: OOF message flows

D. Appendix – Smart cities initiatives and related standardization activities

This appendix provides a general overview of diverse initiatives regarding the smart cities initiatives and related standardization activities.

D.1 Smart Sustainable Cities (SSC)

The Smart Sustainable Cities (SSC) is an initiative that is aligned with the SDG 11: Sustainable Cities and Communities [4], aiming to contribute to the following:

- The massive amount of data in the smart urban ecosystem must be collected, analysed and shared. The shared and exchanged data can rely on blockchain mechanisms. Services can rely on AI;
- Smart Architectures, addressing interoperability, scalability, flexibility, fault tolerance, availability, manageability, resilience, vendor independence. Reference architecture oneM2M;
- Implement and monitor the security of IoT and smart applications, considering emerging technologies like AI, edge computing, autonomous systems.

D.2 CEN-CENELEC-ETSI

ETSI and other standardization bodies have united efforts to promote a forum on smart and sustainable cities and communities[17]. Diverse workgroups have been defined in order to address European needs in a consistent approach with ISO TC 268 'Sustainable Cities and Communities', and also to enable development and implementation of a holistic and integrated approach to the achievement of sustainable development and sustainability.

This forum is providing contributions in diverse areas, such as:

- Machine-to-Machine (M2M) communications, with smart devices in smart cities. Relevant standards include ETSI TR 103 290 – IoT in Smart Cities, M2M Service layer with diverse hardware and software components;
- Green Smart cities, with relevant committees, groups:
  - TC ATTM Sustainable Digital Multiservice Communities towards the creation of digital services in cities and communities. Example ICT waste management in sustainable communities;
  - ISG OEU Industry Specification Group on Operational energy Efficiency for Users, for KPIs management of green smart cities with residential and office environments;
- Context information Management with the work of:
  - ISG CIM Industry Specification Group on cross-sector Context Information Management.

D.3 ITU Standardization group (SC 20)

The ITU SC 20 group has a set of recommendations, where the most relevant for smart cities are:

[17] https://standards4citizens.etsi.org/Links/

- Y.4122: Requirements and capability framework of the edge-computing-enabled gateway in the Internet of things;
- Y.4200: Requirements for the interoperability of smart city platforms;
- Y.4201: High-level requirements and reference framework of smart city platforms;
- Y.4202: Framework of wireless power transmission application service;
- Y.4203: Requirements of things description in the Internet of things;
- Y.4204: Accessibility requirements for the Internet of things applications and services;
- Y.4207: Requirements and capability framework of smart environmental monitoring;
- Y.4208: Internet of things requirements for support of edge computing;
- Y.4210: Requirements and use cases for universal communication module of mobile IoT devices;
- Y.4211: Accessibility requirements for smart public transport services;
- Y.4413: Requirements and reference architecture of the machine-to-machine service layer;
- Y.4417: Framework of self-organization network in Internet of things environments;
- Y.4421: Functional architecture for unmanned aerial vehicles and unmanned aerial vehicle controllers using IMT-2020 net;
- Y.4453: Adaptive software framework for Internet of things devices;
- Y.4455: Reference architecture for Internet of things network service capability exposure;
- Y.4456: Requirements and functional architecture for smart parking lots in smart cities;
- Y.4457: Architectural framework for transportation safety services;
- Y.4458: Requirements and functional architecture of a smart street light service;
- Y.4459: Digital entity architecture framework for Internet of things interoperability;
- Y.4462: Requirements and functional architecture of open IoT identity correlation service;
- Y.4466: Framework of smart greenhouse service;
- Y.4467: Minimum set of data structure for automotive emergency response system;
- Y.4468: Minimum set of data transfer protocol for automotive emergency response system;
- Y.4500.1: oneM2M – Functional architecture.

ITU also coordinates the United for Smart Sustainable Cities (U4SSC) (ITU, 2021). As per the U4SSC report the following use cases are pertinent:

D.3.1 Use Case 1 – Air quality management (Example of California, USA)

Goal: clean the air and protect the health of residents.

“The Envirosuite´s platform combine multiple data sources data such as air quality data, weather data, emission rates, weather forecasts and altitude to provide a very accurate baseline about air pollution. Significant innovation in design makes complex data sets simple for non-subject matter experts to understand and use in real-time decision making and response to incidents.” (ITU, 2021)

Needs:

- Real-time data;
- Quasi Real-time processing (at edge and cloud levels).

"Two unique approaches are used during the implementation phase. 1) Sensor data are integrated with real-time information on wind speed, wind direction and the variability in wind direction to provide a real-time designation of the likely area that a measured value has originated from. With clever design, users of the system can immediately see which industry or other source is the likely cause of air quality incident. 2) Three-dimensional, non-steady state meteorological modelling techniques (commonly used in advanced dispersion modelling) are re-engineered and reprocessed and displayed in a way to provide an immediate reverse trajectory display of the likely source of elevated sensor value or complaint in the community."

D.3.2 Use Case 2- Crime prediction for more agile policing in cities (Rio de Janeiro, Brazil)

Goal: Enable predictive policing and provide crime data accessible to everyone.

"Predictive policing is also evolving. They are benefiting from advances in machine learning, coupled with more affordable computational power. When compared to traditional hot-spots mapping approaches using retrospective data, predictive analytics can process more granular data at a more rapid pace, generating predictions associated not only to a location, but also to a crime type, and to specific times of the day and days of the week" (United for Sustainable Cities (U4SSC) initiative, 2019).

"CrimeRadar is a digital platform that forecasts the probability of crime. It runs on smartphones and desktop browsers. The software uses advanced data analytics to show real and relative crime rates and risks for different neighborhoods at different times in the Rio de Janeiro municipality" (United for Sustainable Cities (U4SSC) initiative, 2019).

"CrimeRadar makes crime data accessible, interactive and useful" (United for Sustainable Cities (U4SSC) initiative, 2019).

E. Appendix – European Innovation Partnership on Smart Cities & Communities (EIP-SCC)

This appendix represents the main points presented on the initiative: Smart cities and communities Standardization to meet citizen and consumer requirements. Standards are perplexing for cities, and citizen concerns like usability, accessibility, and data security are frequently overlooked. The Human Factors Technical Committee of ETSI is creating a Technical Report that will provide an overview on which standards are relevant when considering the needs of residents/visitors regarding smart cities or communities. In the following sections, we briefly describe the main projects/initiatives that are being developed.

More information available at: https://standards4citizens.etsi.org/Links/

E.1 Action Clusters

An Action Cluster is a group of partners that have agreed to cooperate on specific smart city challenges by sharing their knowledge and skills with their peers and identifying gaps that need to be filled at the European level. In this context, there are six main action clusters which will be described in the following sections:

1. Sustainable Urban Mobility;
2. Sustainable Built Environment;
3. Integrated Planning, Policy and Regulations;
4. Integrated Infrastructures and Processes;
5. Business Models & Finance;
6. Citizen Focus.

E.1.1 Sustainable Urban Mobility

Sustainable Urban Mobility (SUM) is a priority in all major Smart City initiatives. As a consequence, addressing Climate Change Goals, improving air quality, realizing seamless inter-modal mobility, using renewable energy for transportation, and shifting mobility away from individual motorized vehicles and toward sharing and public transportation will necessitate more data-driven mobility services than ever before. Cities and communities need mobility data for new services such as AI-based analytics, traffic flow analysis, park search traffic analysis, environmental-sensitive traffic management, or to visualize it using Digital Twins.

The Action Cluster includes three active initiatives:

1. EV4SCC (Electric Vehicle for Smart Cities & Communities);
2. UAM (Urban Air Mobility);
3. NMS (New Mobility Services);

E.1.1.1 Electric Vehicles for Smart Cities & Communities

Main Goal: To become the largest platform of Electric Mobility in the World.

Key findings:

- Intelligent management of public and private fleets of electric vehicles;
- Smart urban logistics;
- Intelligent electrification of public transport;
- Autonomous vehicles;
- Innovative integrated infrastructure;
- Smart electro-mobility solutions that serve multi-modal mobility services.

E.1.1.2 Urban Air Mobility (UAM)

Main Goal: The mission is to drive the sustainable and responsible transition of urban mobility to the vertical (third) dimension.

Key findings:

- Sustainable solutions for passenger transport and freight that foster accessible, safe, and affordable mobility, while aligning with European Green Deal emissions reduction objectives;
- Holistic planning approach that encompasses not only the integration of UAM, along with its support infrastructure on the ground into the transportation system, but also the urban infrastructure and overall city liveability.

E.1.1.3 New Mobility Services (NMS)

Main Goal: The primary objective is to deliver New Mobility Services on a large scale using user-centric design approaches. Due to the complexity of the shift, all stakeholders must be brought together in a learning-by-doing multi-stakeholder ecosystem.

Key findings:

- Interoperability and developing a long-term business model involving the proper industry partners for whom international standards are critical;
- Implementations tailored to both local conditions and agreed design (think globally, act locally);
- Adoption of user-centric design paradigms for the 'learning by doing' principles. Among other things, to determine whether or not assumptions about people's behavior are correct;
- The roles of private and public responsibility will shift;
- The role of private and public duties in the realm of mobility will alter.

E.1.2 Sustainable Built Environment

The key problems are to decrease energy use, environmental effect, and carbon footprint. The most challenging difficulty in this sector is scaling up (new) solutions and materials. To recognize that each city has a unique environment, it is critical to integrate the following requirements:

- Provide stakeholders (industry, cities, operators, etc.) with the tools they need to make appropriate systemic or individual decisions and facilitate scaling up solutions by enabling industries to provide solutions that are fit for purpose while also reasonably priced high quality;
- Offer a large-scale launching pad for new concepts to test and release into the market and test and execute new financial products and models.

Two active initiatives are included in the Action Cluster:

1. Positive Energy Blocks;
2. Deep Retrofitting.

E.1.2.1 Positive Energy Blocks

Main goal: To deploy 100 Positive Energy Blocks (PEBs) throughout the EU and neighbouring countries by 2020, with at least one PEB deployed in each Member State. 50 percent of which should be in cities with a population of 100,000 people or more. A PEB is a collection of at least three connected neighbouring buildings that produce more primary energy than they use annually. To use complementary energy consumption curves and optimize local renewable energy generation, consumption, and storage, these buildings must serve multiple functions (housing, workplaces, commercial spaces, etc.). Another significant benefit of the concept is to contribute to urban regeneration by generating a functional and social mix.

Key findings:

- Cities will lead PEB Initiatives, backed by industry and investors. Ideally, they will be implemented in city centers, using an integrated strategy centered on mixed-use new construction and retrofit buildings, emphasizing ICT technologies;
- The relevance of business concepts and financial structuring cannot be overstated. Financing may come from public and private sources, and long-term development will undoubtedly need business model innovation;
- The effective evolution of these twenty-first-century sustainable districts must also consider other critical city systems such as water, waste, urban biodiversity, and social sciences.

E.1.2.2 Deep Retrofitting

Main Goal: Developing new sustainable ways to build retrofitting based on thermal load electrification and the deployment of novel economic models: converting thermal load to electric for energy self-consumption.

Key findings:

- The obsolescence of existing building stock at European level is a challenge for energy savings and climate targets, a great opportunity for local economic development;
- A Deep Retrofit Site (DRS) is composed of one or more buildings that will undergo transformation, allowing either the reduction to a third of their energy consumption/CO_2 emissions;
- Drive smart technologies in buildings, mobilization of investments in building renovation, tackling energy poverty.

E.1.3 Integrated Planning, Policy and Regulations

This action cluster integrated planning, policy, and regulations focus on novel forms of governance, policy development, and appropriate regulatory frameworks required to support and encourage large-scale implementations and roll-outs of climate-neutral and smart city solutions. Cities must have an acceptable set of framework conditions regarding rules, regulations, and monitoring to smarten up.

Initiatives that are active within their scope:

1. Tools for decision-making, management, and benchmarking.

E.1.3.1 Tools for Decision-Making, Management, and Benchmarking

Main Goal: Assist in developing a measuring framework for policymakers and city authorities to monitor and report smart city projects.

Key findings:

- To identify the needs of decision-makers when launching a smart and sustainable program for the city or community;
- Collaborate to analyse existing KPIs and monitor the demand for additional KPIs;
- Create a uniform standard across the EU. In this regard, they collaborate on all standardization-related activities, both at the EU level (CEN, CENELEC, ETSI, EU projects, etc.) and worldwide (ISO, IEC, ITU), contributing to the coordination of efforts and the convergence of standards. To monitor gaps using available technologies and develop relevant KPIs and their relationship to standardization concerns.

E.1.4 Integrated Infrastructures and Processes

Sharing data among municipal agencies and stakeholders and repurposing public assets to accomplish desired objectives is very modern, necessary, and frustratingly difficult. These possibilities and difficulties will persist as technology continues to evolve at a rapid pace.

With this in mind, the Action Cluster consists of four initiatives:

1. Urban Data Platform;
2. Geospatial Cities;
3. Small Giants;
4. Humble Lamppost.

E.1.4.1 Urban Data Platform

Main Goal: Accelerate the adoption of shared open urban data platforms and ensure that by 2025, cities with competent urban data platforms will serve 300 million European inhabitants, in total.

Key findings:

- The EU-wide study offers a strong platform for ensuring that the UDP initiative's actions satisfy market demands;
- Alignment across the several other EC-supported programs related to digital transformation is critical to successfully provide a clear message and shared assets to the (cities) market.

E.1.4.2 Geospatial Cities

Main Goal: Raise awareness of the game-changing potential that geospatial data and services offer to cities, and advocate for the continued development of services tailored to the needs of cities and to the operational processes of public administrations, SMEs, and NGOs.

Key findings:

- There is a lack of market awareness of what geospatial data can achieve in cities. Nonetheless, most cities are already active users, for example, in sectors such as transportation. It is critical to raise awareness of the value that geospatial data can provide cities efficiently to draw attention, support the case, and motivate action;
- This plan includes gathering a robust evidence-based portfolio of case studies, refining our understanding of local service requirements via conversation, and proving and sharing the outcomes.

E.1.4.3 Small Giants

Main Goal: Provide a gateway for smaller cities to action-focused theme networks through which extremely practical solutions adapted to the common needs of smaller cities may be produced.

Key findings:

- Create a 'View from City Hall' White Paper and communications pieces for tiny giants based on an officer and (Department) Mayor interaction via surveys and interviews, in order to generate both interest and action;
- Examine the shared vision and objective, to create a feasible financial model;
- Establish "Small Giants" as a "Smart Brand";
- Establish a virtual Small Giants project secretariat by identifying financing resources to provide administrative assistance and early-stage cooperation;
- Formally confirm the existing 20-40 candidate cities, with continuous commitment;
- Create a portfolio with brief profiles from each verified city within a consistent/standardized way;
- Support the Mayors' Covenant (which have multiple smaller cities as signatories);
- Create a hub-and-spoke model in a single focal point, with two objectives in mind: to drive communications and activities in each nation and subject, and to serve as the primary link to the leading city and secretariat;
- Fine-tune the technique for "selling" the Small Giants concept to additional locations to expand the network;
- Create a website for information, sign-up, and so forth, or use the redesigned SCM site;
- Continue to advertise through event coverage;
- Locate and collect case studies of successful EU initiatives with candidate Small Giants for inclusion on the website.

E.1.4.4 Humble Lamppost

Main goal: Install 10 million smart lampposts in European towns.

Key findings:

- Upgrades to smart lampposts are an apparent *Quick Win* for any community;
- City-wide LED luminaire replacement (if not previously completed), optimizing lighting settings to improve energy efficiency while guaranteeing the safety and quality of place, and identifying the numerous additional services that may be deployed in specific areas to multi-purpose a city's lamppost assets;

- The initiative takes in consideration the solution using an open component-based approach to support its aims, concentrating on what is functionally similar for all communities.

E.1.5 Business Models & Finance

This Business Models and Finance (BM&F) Action Cluster was reintroduced in October 2019 under new leadership. It will carry on the excellent work that has been done in the past and endeavour to satisfy the changing demands of the city market. The Action Cluster strives to be the preferred forum for stakeholders discussion and to identify and remove barriers that block the faster growth of the smart cities market. The goal is to expedite the development of intelligent city markets through information exchange, innovation, and experience in business models, financing, funding, and procurement.

E.1.6 Citizen Focus

In an era of urban development and smart city digitalization in response to climate change and population issues, there is a risk that inhabitants would receive insufficient attention. According to the Citizen Focus Action Cluster, citizens are key players in the regeneration and growth of intelligent communities. The advocacy strategy is founded on civic involvement, empowerment, participation, and co-creation: they recognize that people's voices may be critical in putting pressure on government, service providers, and organizations to encourage full responsiveness of urban innovations to citizen demands and objectives. Co-creation of services and solutions with citizens' informal networks and grassroots civil society groups has also been shown to be effective in design/creativity and execution. Citizens may assist local governments in prioritizing and responding to smarter cities by continually proposing inclusive 'smart' solutions sensitive to differential disparities and existing divides.

E.2 Other EU Initiatives

E.2.1 Bridge

BRIDGE is a European Commission program that brings together Horizon 2020 Smart Grid, Energy Storage, Islands, and Digitalisation Projects to offer a systematic picture of cross-cutting difficulties faced in demonstration projects that may impede innovation. More information available at: https://www.h2020-bridge.eu/

E.2.2 BUILD-UP: The European platform for energy efficiency in buildings

BUILD UP is a space for building professionals, local governments, and building tenants. BUILD UP encourages the exchange of best practices for the deployment of energy-saving solutions in buildings across Europe. More information available at: http://www.buildup.eu/en

E.2.3 Covenant of Mayors for Climate and Energy

The EU Covenant of Mayors on Climate and Energy brings together hundreds of municipal governments voluntarily committed to fulfilling EU climate and energy goals. More information available at: https://www.eumayors.eu/en/

E.2.4 EIT Urban Mobility

EIT Urban Mobility is a European Institute of Innovation and Technology program (EIT). They have been striving to inspire positive changes in the way people travel around cities since January 2019 to make them more liveable places. They hope to become Europe's greatest urban mobility transformation effort. The EIT, a body of the European Union, will provide up to € 400 million in co-funding (2020-2026). More information available at: https://www.eiturbanmobility.eu/

E.2.5 Eltis

Eltis supports the sharing of information, expertise, and experience in Europe in sustainable urban transportation. Eltis also includes a platform with useful resources to model the traffic, to create data visualizations (e.g., PoliVisu) for policy works. Besides the tools, constant resources in the form of Webniars are provided with detailed in formation. More information available at: https://www.eltis.org/

E.2.6 ESPRESSO

The ESPRESSO (systEmic Standardization apPRoach to Empower Smart citieS and cOmmunities) project, financed by Horizon 2020, intends to secure the interoperability of Smart City technologies. This will assist cities in avoiding entry barriers or vendor lock-in by encouraging the use of standard meta-data formats and interoperable (open) interfaces rather than proprietary ones. The ESPRESSO project will concentrate on creating a conceptual Smart City Information Framework based on open standards. This framework will include a Smart City platform and a variety of data provision and management tools. More information available at: http://espresso-project.eu/

E.2.7 EU City Facility

Municipalities and local governments are the driving force behind Europe's sustainable energy transformation. They can construct comprehensive sustainable energy investment programs and combine smaller initiatives into more significant investment portfolios and mobilize considerable financial resources for the energy transition. The EUCF, established under the European Union's Horizon 2020 Framework Programme for Research and Innovation, will unlock this local potential by providing tailor-made, fast, and simplified financial support (in the form of EUR 60,000 lump sums) and related services to enable municipalities in Europe to develop relevant investment concepts related to the implementation of actions identified in their climate and energy action plans. More information available at: https://www.eucityfacility.eu/home.html

E.2.8 European Capital of Innovation (iCapital) Award

This is a monetary award given annually to the European city that best demonstrates its capacity to use innovation to better the lives of its residents. The contests run under the Horizon Europe and mainly aim to address challenges related with city sustainability. More information available at: https://eic.ec.europa.eu/eic-funding-opportunities/eic-prizes/european-capital-innovation-awards_en

E.2.9 European Energy Research Alliance (EERA) Joint Programme Smart Cities

The Joint Programme on Smart Towns intends to provide new scientific methodologies, concepts, and technologies to help European cities evolve into smart cities. The primary focus is on large-scale renewable energy integration and increased energy efficiency, facilitated through intelligent energy management at the municipal level. More information available at: https://www.eera-set.eu/

E.2.10 European Green Capital Award

The European Green Capital Award was established on 15 May 2006 in Tallinn, Estonia, by 15 European cities and the Association of Estonian cities. Their green vision was transformed into a Memorandum of Understanding that established an award to recognize communities pioneering ecologically friendly urban life. The European Commission initiated the program in 2008. It is critical to recognize and reward cities working to enhance the urban environment and progress toward healthier and more sustainable living environments. Although progress is its reward, the joy of receiving a prominent European award encourages towns to engage in additional efforts and raises awareness inside the city and other cities. The prize allows cities to encourage one another and share examples of best practices in action. The underlying message that the award program seeks to express to local communities is that Europeans have the right to live in healthy urban settings. Cities

should thus seek to enhance their inhabitants' quality of life while reducing their influence on the global environment. The Award's motto "Green cities — fit for life" encapsulates this idea.
More information available at: https://ec.europa.eu/environment/europeangreencapital/about-the-award/

E.2.11 Green Digital Charter

The Green Digital Charter is a proclamation that commits towns to collaborate to meet the EU's climate goals via information and communication technologies (ICT). As a result, it encourages progress in combating climate change through the innovative use of digital technology in cities. More information available at: http://www.greendigitalcharter.eu/

E.2.12 Horizon 2020

Horizon 2020 is the largest EU Research and Innovation initiative ever, with approximately €80 billion in financing available over seven years (2014—2020) – in addition to the private investment that this money will attract. Bringing amazing ideas from the lab to the market promises additional breakthroughs, discoveries, and world-firsts. More information available at: https://ec.europa.eu/programmes/horizon2020/what-horizon-2020

E.2.13 Intelligent Cities Challenge

The Intelligent Cities Challenge (ICC) is a European Commission program that assists 136 cities in leading the intelligent, ecological, and socially responsible recovery by utilizing cutting-edge technology. The ICC cities and their local ecosystems will be engines for local economic revival, creating new employment and increasing citizen involvement and well-being. More information available at: https://www.intelligentcitieschallenge.eu/

E.2.14 JPI Urban Europe

JPI Urban Europe was established in 2010 to address today's global urban concerns, develop a European research and innovation center on urban issues, and develop European solutions through coordinated research. JPI Urban Europe invites anybody with a desire to enhance urban living in the twenty-first century. Its mission is to bring together government, civic society, scientists, innovators, businesses, and industry to create a new environment for research and innovation. They provide a wide range of experimental zones and long-term research infrastructures. Its mission is to develop knowledge, tools, and platforms for dialogue on urban transitions. More information available at: https://jpi-urbaneurope.eu/

E.2.15 Living-IN.EU

In an era when cities and communities are turning to digital solutions to address a rising number of linked difficulties, they must channel their efforts via a 'European Way,' in which digital solutions aid in the creation of places where people want to live and work. Smart urban transportation, energy efficiency, sustainable housing, digital public services, and civic-led governance are all examples of digital solutions. The widespread adoption and expansion of these solutions are critical to assisting our cities and communities in meeting their climate targets and reducing their environmental footprint while encouraging citizen participation and bringing prosperity to all types of businesses, including SMEs and start-ups. They want to deliver this transition's economic and social advantages to all local communities through co-creation with citizens and establish an inclusive digital Europe with powerful digital services, technologies, infrastructures, and skills. More information available at: https://living-in.eu/

E.2.16 POLIS network

POLIS is the main network of European cities and regions working together to create innovative local transportation technology and regulations. Since 1989, European municipal and regional governments have collaborated within POLIS to improve sustainable mobility via innovative transportation solutions. More information available at: https://www.polisnetwork.eu/who-we-are/about-polis/

E.2.17 SETIS – Strategic Energy Technologies Information System

SETIS is critical to the successful implementation of the SET-Plan by identifying energy technology and RD&D objectives, building consensus around the SET-Plan program, identifying new opportunities, and assessing the SET-effectiveness Plan's and efficiency in delivering energy and climate change policy goals. The biggest challenge and opportunity of our time is to become the world's first climate-neutral continent by 2050. To that end, the European Commission unveiled the European Green Deal, the most ambitious set of policies aimed at enabling European individuals and companies to benefit from a sustainable green transition. Combating climate change is central to the Green Deal. To live up to the Paris Agreement, Europe should commit to achieving carbon neutrality by 2050. The Energy Union plan is based on achieving a significant reform of Europe's energy system while cost-effective.

More information available at: https://setis.ec.europa.eu/about-setis

E.2.18 URBACT - Driving change for better cities

URBACT's mission is to enable cities to work together and develop integrated solutions to common urban challenges by networking, learning from one another's experiences, drawing lessons, and identifying good practices to improve urban policies. More information available at: https://urbact.eu