Demonstration of **5G** solutions for
SMART energy **GRID**s of the future

Deliverable 3.2

Final report for the development of the 5G network facility

Version 1.0 - Date 23/12/2022

The 5G Infrastructure Public Private Partnership

This project has received funding from the European Union's *Horizon 2020 research and innovation programme* under grant agreement n° 101016912

D3.2 – Final report for the development of the 5G network facility

Document Information

Programme	Horizon 2020 Framework Programme – Information and Communication Technologies
Project acronym	Smart5Grid
Grant agreement number	101016912
Number of the Deliverable	D3.2
WP/Task related	WP3
Type (distribution level)	PU Public
Date of delivery	23-12-2022
Status and Version	Version 1.0
Number of pages	56 pages
Document Responsible	Antonello Corsi – ENG
Author(s)	Andrés Cárdenas – i2CAT August Betzler – i2CAT Borja Otura – ATOS Dimitris Brodimas – IPTO Gaetano Scibilia – W3 Gianluca Rizzi – W3 Ioannis Chochliouros – OTE Irina Ciornei – UCY Lenos Hadjidemetriou – UCY Rita Santiago – UW Nicola Cadenelli – NBC

	Nicola di Pietro – ATH
	Paula Encinar – ATOS
	Sonia Castro – ATOS
	Yanos Angelopoulos – AXON
	Michalis Rantopoulos – OTE
Reviewers	Antonello Corsi – ENG
	Angelos Antonopoulos – NBC
	Daniele Ronzani – ATH

Revision History

Version	Date	Author/Reviewer	Notes
A	03/10/2022	NBC	Initial table of content for feedback
B	25/10/2022	ATH, ATOS, I2CAT, ENG, NBC	First round of contribution from many partners
C	5/12/2022	ATH, ATOS, I2CAT, ENG, NBC, ENEL, EE, OTE, IPTO	Complete draft with all contributions
D	6/12/2022	ATH, ATOS, I2CAT, ENG, NBC, ENEL, EE, OTE, IPTO	Version for the Internal Review
Revised	16/12/2022	ENG, NBC, ATH	Revised Version
Final	23/12/2022	ENG, NBC	Submitted Version

Executive summary

The objective of this deliverable is to present an update of the work done in WP3, both in terms of the development of the 5G network infrastructure for all four use cases, as well as the implementation of the Smart5Grid architecture. In particular, as key contributions, it completes the previous deliverable D3.1 by providing the details of the final network facilities and outlines the work done to develop, integrate, test, and deploy the Smart5Grid architecture and its component in each use case.

Regarding the 5G network of the use cases, this deliverable provides an update from D3.1 and the final view of the infrastructure of each use case, located in Italy, Spain, Bulgaria, and Greece. Regarding the components of the Smart5Grid architecture, this deliverable focuses on the Network Application Controller & MEC Orchestrator (NAC) and its integration with the other components of the architecture. The other two architectural components, i.e., the OSR and the V&V framework, will be described in more detail in their dedicated deliverables – D3.3 and D4.2, respectively.

The provided document will support all the work of WP4, WP5, and WP6 until the end of the project. In these WPs, the focus will be on the deployment and testing of the Network Applications in the actual UCs and only minor adjustments are to be expected on the network and Smart5Grid architecture.

Table of Contents

List of Figures

List of Tables

1. Introduction

This second deliverable of WP3 is one of three deliverables that provides an update of the previously presented D3.1 and reports the final state of the work done in WP3. While D3.3 focuses on one Smart5Grid's Open-Source Repository and D3.4 on the pre-piloting preparatory testing activities, D3.2 focuses on the development of the 5G facilities used in all four use cases of Smart5Grid.

1.1. Scope of the document

This document reports the work performed by all partners and the results obtained in Task *3.1: Smart5Grid network access/core platform building and orchestration* and in Task *3.3: Platform integration and interfacing between 5G and energy infrastructure*.

The main scope of this document is twofold:

- Complement D3.1 [1] with an update and final description of the 5G network architectures and their components for each use case. These architectures are those that will support the work in WP5 and WP6.
- Outline the work done to develop, integrate, and deploy the Smart5Grid architecture and its component in each use case.

1.2. Document Structure

The deliverable is organized in the following manner:

Section 2 provides a brief overview regarding the main concepts and components of the Smart5Grid architecture presented in the previous deliverables.

Section 3 describes in detail the network deployment of all four use cases, ranging from the UE to the deployed RAN, UPF and 5G Core, while we also discuss the challenges faced during these deployments, including the serious security concerns of deploying services in production environments of national Telco operators, the availability of stand-alone 5G cores, and the challenges of using a cross-border infrastructure with Telcos from different countries, among others.

Section 4 outlines the integration and interfaces between the services and the 5G infrastructure done during T3.1 and T3.3. Hence, this section will mainly focus on the Network Application Controller and MEC Orchestrator, which is the component that orchestrates the developed smart energy services and guarantees their smooth integration with the 5G infrastructure. The other components of the infrastructure are to be covered in other deliverables (i.e., OSR in D3.3 and V&V in D4.1) and, thus, they are out of the scope of this document. Nonetheless, Section 4 will describe the integration tests carried out by the partners to validate the integration with architectural components (including OSR and V&V).

Finally, Section 5 reports the conclusions and Section 6 lists the references.

1.2.1. Notations, Abbreviations, and Acronyms

Table 1: Acronyms list

Item	Description
3GPP	3rd Generation Partnership Project
5G	5th Generation (of mobile telecommunication networks)
5G PPP	5G Infrastructure Public Private Partnership
A&A	Authentication and Authorization
AAA	Authentication, Authorization and Accounting
AB	Advisory Board
AF	Application Function
AKA	Authentication and Key Agreement
AMF	Access Management Functions
AMI	Advanced Metering Infrastructure
API	Application Programming Interface
APN	Access Point Name
BBU	Base Band Unit
BIOS	Basic Input/Output System
BRP	Balancing Responsible Party
BSP	Balancing Service Provider
BSS	Battery Storage System
CA	Consortium Agreement
CFS	Customer Facing Service
CIR	Container Image Registry
CISM	Container Infrastructure Service Management
CLI	Command Line Interface
CM	Configuration Management
CN	Core Network
CNF	Containerized Network Function
CoE	Centre of Excellence
CP	Connection Point
CPD	Connection Point Descriptor
CPE	Customer Premise Equipment
CPF	Control Plane Function
CRAN	Cloud RAN
CRUD	Create, Read, Update, Delete
CSMF	Communication Service Management Function
CT	Current Transformer
CUPS	Control and User Plane Separation
DCN	Data Communication Network
DER	Distributed Energy Resources
DevOps	Development and Operations
DHCP	Dynamic Host Configuration Protocol
DLT	Distributed Ledger Technology

DNAI	Data Network Access Identifier
DNN	Data Network Name
DNS	Domain Name System
DoW	Description of Work
DRES	Distributed renewable energy sources
DSO	Distribution System Operator
EDSO	European Distribution System Operators for Smart Grids
EEGI	European Electricity Grid Initiative
ENTSO-E	European Network of Transmission System Operators for Electricity
EPC	Evolved Packet Core
EPIA	European Photovoltaic Industry Association
ETSI	European Telecommunications Standards Institute
EU	European Union
EWEA	European Wind Energy Association
FM	Fault Management
FP7	Seventh Framework Program
GA	Grant Agreement
GDS	Global Digital Services
GGSN	Gateway GPRS Support Node
GIT	Global Information Tracker
GOOSE	Generic Object-Oriented Substation Even
GPIO	General Purpose Input Output
GPS	Global Positioning System
gRPC	Google Remote Procedure Calls
GUI	Graphical User Interface
HLR	Home Location Register
HSS	Home Subscriber Server
HTTP(S)	Hypertext Transfer Protocol (Secure)
IAP	Identity and Access Proxy
ICT	Information and Communications Technologies
IMSI	International Mobile Subscriber Identity
IMT	Information Model Translation
IPMI	Intelligent Platform Management Interface
iPXE	Preboot eXecution Environment
ISO	Optical Disc Image
IT	Information Technology
KPI	Key Performance Indicator
LAN	Local Area Network
LBO	Local Break-Out
LCM	LifeCycle Management
LI	Lawful Interception
LTE	Long-Term Evolution
LV	Low Voltage

M&O	Management and Orchestration[1]
MANO (NFV)	Management and Orchestration[1]
MCIO	Managed Container Infrastructure Object
MCIOP	Managed Container Infrastructure Object Package
MEC	Multi-access Edge Computing
MECO	Multi-access Edge Computing Orchestrator
MIMO	Multiple-Input Multiple-Output
MME	Mobility Management Entity
MMS	Manufacturing Message Specification
MNO	Mobile Network Operator
MPLS	Multi-Protocol Label Switching
MSA	Micro-Service Architecture
MV	Medium Voltage
NAC	Network Application Controller & MEC Orchestrator
NBI	Northbound Interface
NEF	Network Exposure Function
NFMF	Network Function Management Function
NFVI	Network Function Virtualization Infrastructure
NFVIaaS	NFV Infrastructure-as-a-Service
NFVO	Network Function Virtualization Orchestrator
NGMN	Next Generation Mobile Networks
NR	New Radio
NRF	Network Resource Function
NS	Network Service
NSaaS	Network Slice-as-a-Service
NSD	Network Service Descriptor
NSFVal	Network Services and Functions Validator
NSI	Network Slice Instance
NSMF	Network Slice Management Function
NSSF	Network Slice Selection Function
NSSI	Network Slice Subnet Instances
NSSMF	Network Slice Subnet Management Function
nZTP	near Zero-Touch Provisioning
OAM	Operations, Administration, and Management
ONAP	Open Network Automation Platform
OPC	Open Platform Communications
OPC-UA	OPC Unified Architecture
OSM	Open-Source Mano

1 Through this deliverable, "Management & Orchestration" is abbreviated as MANO (or NFV MANO) when referring to the Management and Orchestration as defined by ETSI NFV and M&O when referring to the Management and Orchestration framework of the NFV/TELCO layer of the Smart5Grid Architecture.

OSR	Open Service Repository
OSS/BSS	Operational Support Systems / Business Support Systems
PAS IEC	Publicly Available Specification
PC	Personal Computer
PCF	Policy Control Function
PDC	Phasor Data Concentrator
PFD	Packet Flow Description
PGW	Packet Data Network Gateway
PM	Performance Management
PMU	Phasor Measurement Unit
PoP	Points-of-Presence
PPDR	Public Protection and Disaster Relief
PST	Power System Testbed
PV	Photovoltaics
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RES	Renewable Energy Sources
REST	REpresentational State Transfer
ROCOF	Rate of Change of Frequency
ROS	Robotics Operating System
RRU	Remote Radio Unit
RSC	Regional Security Coordinator
RSU	RoadSide Units
RTD	Research and Technology Development.
RT-HIL	Real-Time Hardware-In-the Loop
RTS	Real-Time Simulator
RTU	Remote Terminal Unit
SAP	Service Access Point
SBA	Service-Based Architecture
SBI	Southbound Interface
SCADA	Supervisory Control and Data Acquisition
SDK	Service Development Kit
SEAF	Security Anchor Functionality
S-Gateway	Serving Gateway
SLI	Service Level Indicator
SLO	Service Level Objective
SIM	Subscriber Identity Modules
SME	Small and Medium Enterprise
SMF	Session Management Function
SOA	Service-Oriented Architecture
SP	Service Providers
SSH	Secure Shell
T&D	Transmission and Distribution
TCP	Transmission Control Protocol

TLS	Transport Layer Security
TPM	Trusted Platform Module
TSO	Transmission System Operator
UC	Use Case
UCY	University of Cyprus
UDM	Unified Data Management
UE	User Equipment
UL	Up-link
UPF	User Plane Function
UWB	Ultra-Wideband
V&V	Verification and Validation
VDU	Virtual Deployment Unit
VDU	Virtual Deployment Unit
VIM	Virtual Infrastructure Manager
VLD	Virtual Link Descriptor
VNF	Virtual Network Function
VNFaaS	Virtual Network Function-as-a-Service
VNFC	Virtual Network Function Component
VNFM	Virtual Network Function Manager
vPDC	virtual Phasor Data Concentrator
VPN	Virtual Private Network
VSB	Vertical Service Blueprint
VSD	Vertical Service Descriptor
VT	Voltage Transformer
WAM	Wide Area Monitoring
WP	Work Package

2. Smart5Grid Architecture: Brief Review

The main objective of the Smart5Grid Architecture is to reduce barriers in the market for energy applications that aim to provide solutions for energy grid operators through a common platform for application developers and consumers. The Smart5Grid Architecture, which can be found further defined in deliverables D2.1 [2], D2.2 [3], and D3.1 [1], is composed of three main layers as shown in Figure 2-1 and explained below.

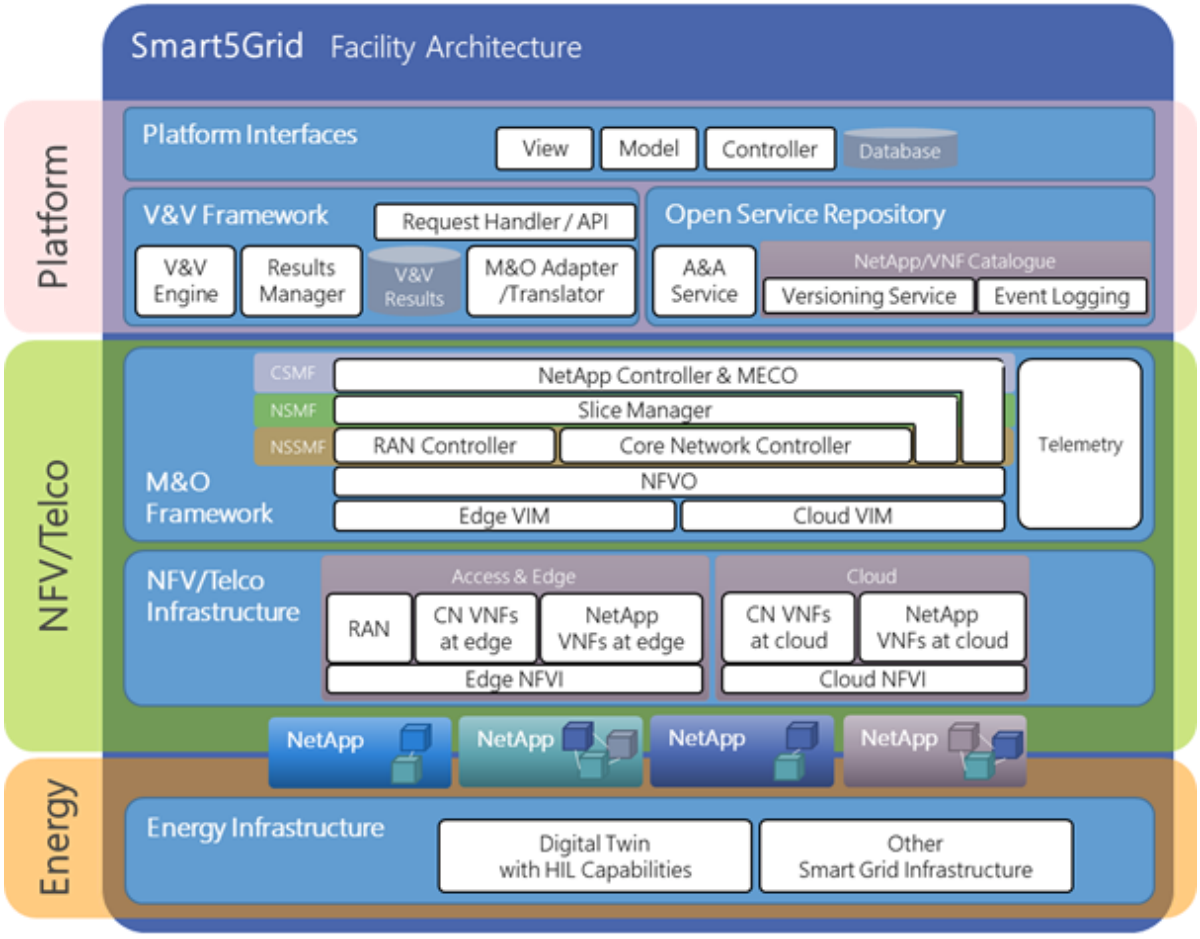

Figure 2-1: Smart5Grid functional architecture.

1. The platform layer is the top layer of the Smart5grid architecture, being the entry point for users and where it opens to third parties. This layer is composed of three main components:
    - User Interface (UI) provides the services exposed through the Open Service Repository (OSR) and Verification Framework V&V component APIs handling the Authorization and Authentication (A&A) of users
    - Open Service Repository (OSR) is a key component, as it allows developers to register their Network Applications and their components and make them accessible to

consumers. The OSR is responsible for keeping records of the actions performed on Network Applications.

- Verification and Validation (V&V) Framework provides automated testing of Network Applications and their components. This component is composed of two sub-modules, the Verification engine, which verifies the syntax, topology, and integrity of a Network Application package; and the second sub-module, the Validation engine, which performs onboarding tests on Network Application as well as on Network Application instances to guarantee the required performance levels.

2. The second layer consists of the virtualization and telecommunications infrastructure on which the Network Applications will be deployed, with their associated management and orchestration functions. This layer contains the components necessary to manage the end-to-end lifecycle of a Network Application implementation, which will be discussed below:
    - Network Application Controller & MEC Orchestrator (NAC) manages Network Application and MEC server lifecycles by performing NAC-specific functions, such as Network Application onboarding and Network Application lifecycle management, as well as resource management including node and network slice management.
    - Slice Manager (SM) ensures the management and orchestration of infrastructure resources as well as the deployment of Network Application services, managing the lifecycle of network slices.
    - NFV Framework manages Network Applications and their components, through two components of the NFV framework. The NFVO, which oversees the orchestration of the services, and the VIM, which oversees providing the computing, storage, and network resources for these services.
    - 5G Core Network (CN) Controller is responsible for the management and orchestration of CN resources, acting at the request of the SM which asks for CN resources to be reserved and allocated to a specific slice.
    - RAN Controller is responsible for the management and configuration of the 5G network radio devices, providing the necessary abstraction to the SM.
    - Telemetry provides monitoring and analysis of functionalities that are responsible for supervising the resources used within a given infrastructure, thus guaranteeing service performance.

3. Finally, the energy infrastructure layer where the energy infrastructure devices are located, which through the telco network will be connected to the services offered by Network Applications. Within this layer, one component stands out: the Real-Time Hardware-In-the Loop (RT-HIL). It is a real-time test infrastructure that allows the configuration of a digital twin of a power system and integrates it with real devices. This component is the focus of D3.4.

3. 5G Network Deployments

One of the objectives of T3.1 was to build and setup the (public or private) 5G networks that support the project's use cases and the Network Application operations. In this section, we complement the initial information about such networks provided in D3.1 [1], and we complete the description of the deployed hardware and software equipment. These components define the 5G network used to support the Smart5Grid infrastructure in each UC.

The Smart5Grid project consists of four use cases, located in four EU countries, and it involves four energy operators, three telco operators, and one private 5G deployment. Before to delve in the 5G network details of each UC, we provide a short summary of the use cases:

- UC 1: Located in Italy, this UC involves the production network infrastructure of WI3.
- UC 2: Located in Spain, this UC involves a private 5G network infrastructure hosted in a substation of ENEL.
- UC 3: Located in Bulgaria, this UC involves the production network infrastructure of VIVACOM.
- UC 4: As a cross-border UC between Greece and Bulgaria, this UC involves the production infrastructure of OTE (Greece) and VIVACOM (Bulgaria).

We highlight that, thanks to the use of a private network in UC2, we could test there and showcase the Slice Manager and its interfaces for interoperation with the Network Application Controller & MEC Orchestrator (northbound) and with the specific subnetwork and virtualization components (southbound). We describe this in Section 3.2. Instead, given the production environment of UC1, UC3, and UC4 and the utilization of the mobile network operators' 5G public commercial network infrastructure in its actual state, the set of 5G network management operations allowed for Smart5Grid was limited, mainly for the unavailability of fully fledged slicing features in the 5G NSA networks of UC1, UC3, and UC4. De facto, it has not been possible to deploy a Slice Manager for such use cases, and resource isolation and reservation are carried out as described in the following subsections via dedicated user-plan functionality deployment, APN differentiation techniques, etc.

3.1. Use Case 1

In Section 2.5.2.3.1 of D3.1 [1], we characterized the features of WI3's Radio Access Network (RAN) exploited in UC1. Here, we complement that description with further details on how the core network and the edge infrastructure are set up to handle the traffic generated by or directed to the UC's end devices.

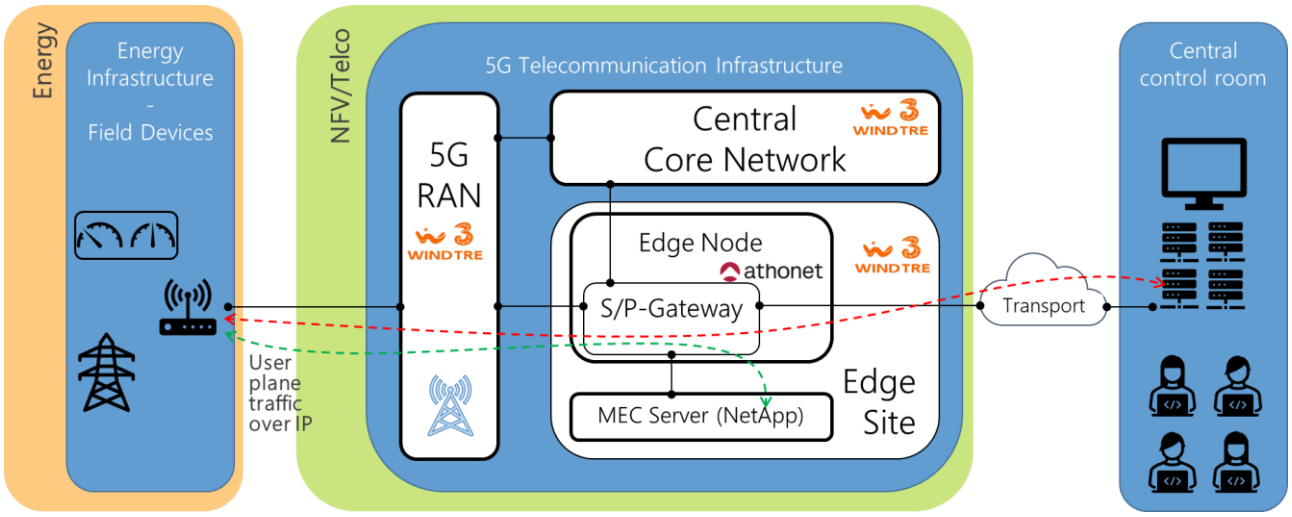

Figure 3-1: 5G network setup of UC1.

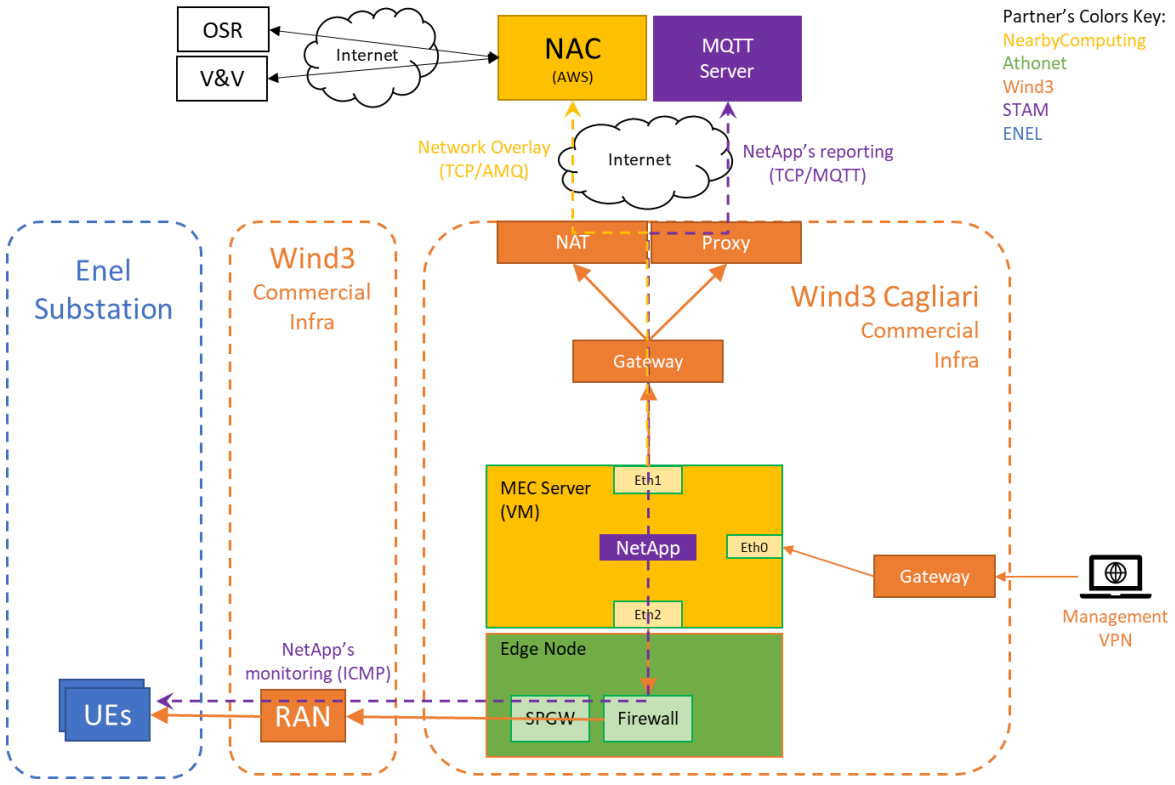

Figure 3-2: Low Level Design of UC1's infrastructure.

First of all, ATH endowed the 5G network of UC1 with an "edge node," whose functionalities allow implementing local traffic breakout, routing selected traffic from the field devices directly to the edge site (or vice versa), and simultaneously letting the rest of the traffic flow to other remote sites (e.g., the central control room of the energy distributor). The control of traffic routes obtained by the means of the edge node decreases the data exchange latencies from the field devices to the edge site where the Network Application is instantiated, thus enabling real-time processing of such data and a more efficient operation of the Network Application. From the mobile networking point of view, the edge node

implements user plane functionalities and contains a Serving Gateway and a Packet Data Network Gateway (S/P-Gateway), running as virtualized network functions over common-off-the-shelf hardware. As represented in Figure 3-1, the S/P-Gateway of the edge node is interfaced with the control plane of WI3's public core network and to WI3's public 5G RAN, and all the corresponding interfaces are 3GPP-defined. As further detailed in Figure 3-2, the edge node also implements a firewall and enhanced security mechanisms to protect the public network operated by WI3 and prevents the introduction of vulnerabilities to the whole 5G system.

The ATH edge node and the MEC Server that hosts the Network Application are installed in Quartucciu, Italy, the nearest WI3 Point of Presence (PoP) to Olbia's Primary Substation that must be monitored in the UC's scope. The edge node runs a Kubernetes cluster manually configured, onboarded into the NAC as cloudlet. Due to the stringent security requirements imposed by the utilization of the public network, the connectivity between the NAC and the edge node is provided by WI3 via a NAT where the edge node always initializes the communication.

Specific International Mobile Subscriber Identities (IMSIs) are used and routed to the S/P-Gateway through two dedicated Access Point Names (APNs), which allows the core network to properly distinguish the traffic sources and destinations (MEC server or central control room) and to route the traffic accordingly: one APN ("SMART5GRID") is dedicated to the traffic that has to be monitored by the Network Application and another ("SMART5GRID2") will be used to transport the remote traffic control to ENEL's central control room.

The dedicated and static routing from the field devices to the ATH S/P-Gateway is implemented through the standardized Mobility Management Entity's (MME) feature "SGW Selection Based on IMSI Number Series and Geographical Area for the MME." The following configurations have been performed in WI3's public core network:

- Home Location Register (HLR) and Home Subscriber Server (HSS) have been configured with a new profile associated to the two APNs and to the Subscriber Identity Modules (SIMs) used within the project.
- Domain Name System (DNS) has been configured to address the Gateway GPRS Support Node (GGSN) and the P-Gateway.
- MME has been configured with:
    - the Geographical Areas that include all the Tracking Areas needed to cover the sites to be monitored,
    - the SP Gateway to be used,
    - the IMSIs dedicated to the project.

A Multiprotocol Label Switching (MPLS) connection has been implemented to connect the edge node and MEC Server to ENEL's central control room.

Finally, the User Equipment (UE) utilized for the secondary substation involved in UC1 (cf. Figure 3-2) is a Cisco IR1101 series router, equipped with a 5G module. Designed to operate in very harsh environmental conditions, it supports a wide range of operating temperatures (-40 to 60°C in standard operation, -40 to 75°C in a forced-air enclosure with 200 LFM of air and tested at 85°C for 16 hours), and it ensures great versatility thanks to its modularity. In fact, it can exploit different communication standards both wired and non-wired. The model used in the Italian demo supports 5G stand-alone (SA) and non-stand-alone (NSA) standards, as well as backward compatibility with 4G LTE networks.

3.2. Use Case 2

3.2.1. 5G private Network of UC2

The private 5G Network used in UC2 features the main components necessary to enable the Network Application deployment at the edge for the ECOGARRAF substation, as described in D3.1 [1]. The network, deployed in the i2CAT labs for the pre-piloting phase and later to be moved to the substation, features three physical servers that host different functionalities: the first server hosts the 5G Core network (5GC), which is an Open5GS[2] instance with its default features and based on release 16. Since the 5G network is deployed locally, there is just one full instance of the 5GC and no edge UPFs are necessary. The second server is the vRAN server that interfaces the gNodeB of the setup to provide radio connectivity to UE. Finally, the third server hosts the Kubernetes cluster in which the Network Applications can be deployed.

Apart from the physical servers, the deployment includes a 4x4 Multiple-Input Multiple Output (MIMO) Remote Radio Unit (RRU). Omnidirectional antennas are available (for lab testing), as well as a directive antenna for the final pilot deployment. To provide connectivity to the sensors and cameras of UC2, two outdoor Customer Premise Equipment devices (CPEs) form part of the setup. These CPEs (2x2 MIMO) offer both Wi-Fi and Ethernet-based connectivity. The latter is used to connect the cameras and the industrial PC. These devices, which we can define as sensors, use the 5G connectivity to transmit the information they capture from the substation, in particular the location of workers and tools, to the Network Application. There, the sensor data is processed to detect whether workers or tools are staying within virtually delimited safety areas. If a safety breach is detected, the Network Application is capable of alerting the workers to prevent accidents. The overall setup of the 5G network including the key elements of the deployment is shown in Figure 3-3.

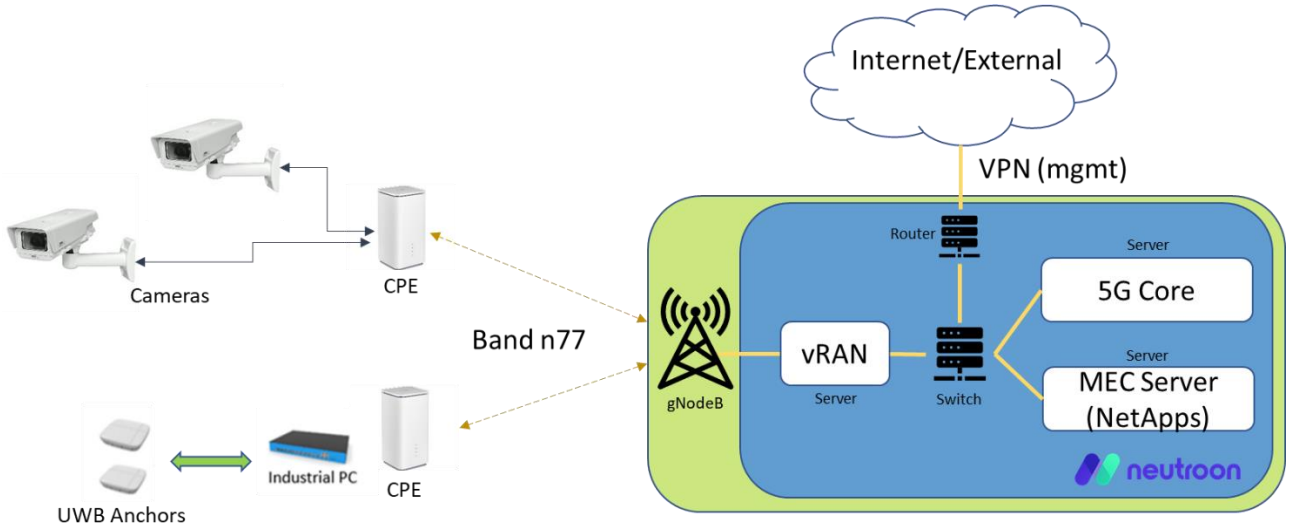

Figure 3-3: UC2 private 5G network setup and key functional elements.

The management platform of the 5G network provider that manages the MEC server, and the radio elements is deployed in the cloud. The platform exposes an API that the NAC developed by UC2

2 https://open5gs.org/

partners (see also Section 4.2.1) is adapting to allow for automatized Network Application deployment. Also, other key elements of the Smart5Grid platform, like the NAC or OSR, are to be deployed outside of the substation. The connectivity between the on-site elements (green box in Figure 3-3) and any element that lies outside the network is done over a dedicated VPN. At the time of writing this deliverable, due to restrictions of the networking configurations and traffic flows allowed in the substation, two options for the outgoing connectivity are considered: i) establishing the desired connectivity with special firewall rules, which is currently being evaluated ii) using a 4G router to connect to public network and to provide management access to the substation. Also, it is still under ongoing discussions whether some elements of the Smart5Grid platform can be deployed in the substation, simplifying the connectivity issues, but also introducing limitations when it comes to the flexibility of making changes to any element (once deployed in the substation, no changes are possible due to the limited connectivity).

3.2.2. Slice Manager – Operational Description

Slice Manager in Use Case 2 (UC2) performs the management and orchestration of the virtualized network services and the infrastructure resources, such as radio equipment and computing nodes. Also, UC2's mobile networking is supported by a 5G private network provider, which is in charge of providing most of the components comprising the Smart5Grid Telco Layer, e.g., Slice Manager, RAN Controller, and NFVO. The reader can refer to [3] for further details. Therefore, the 5G private network platform is seen as a single whole component which exposes a single northbound API to orchestrate the network slices and the applications. In other words, the northbound interface allows to execute some actions regarding the creation and deployment of applications along with the 5G core services. Therefore, Slice Manager is the core component since its functionalities are implemented in the 5G Private network platform.

Since the NAC is "on top" of the Slice Manager, it only uses some specific methods in order to create, read, deploy, and delete Network Applications. Slice Manager exposes three groups of methods for the provision of the 5G end-to-end communication, such as, device registration, slice creation, and application instantiation, which were introduced in [3] and whose interfaces were defined in [1]. It means that the Slice Manager creates and configures network slices composed of radio part, 5G core network, and applications. The NAC only focuses on the management and orchestration of the applications; therefore, it only makes use of the methods related to the creation, instantiation, and deletion of the applications. We will detail in Section 4 4.1.2the methods exposed by the Slice Manager to explain the integration between NAC and Slice Manager using the interface NAC_SM defined in the Smart5Grid platform.

3.3. Use Case 3

A description of the user and RAN equipment utilized in UC3 has been provided in Section 2.5.2.3.3 of [1]. To complete the presentation of the 5G network setup of the use case (see Figure 3-4), we give here further details on the core network configuration. As represented in Figure 3-5, VIVACOM's public core network and public 5G RAN is providing secure connectivity via dedicated private APN. The use of

private APN provides enhanced security to protect user plane data via public network operated by VIVACOM and prevents the introduction of vulnerabilities to UC3's ecosystem.

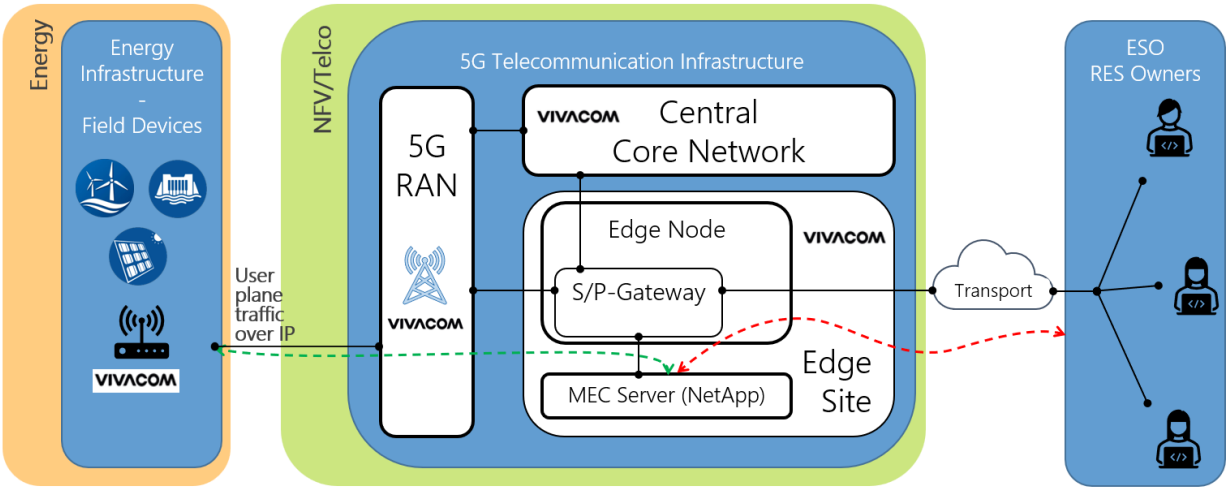

Figure 3-4: 5G Network Setup for UC3.

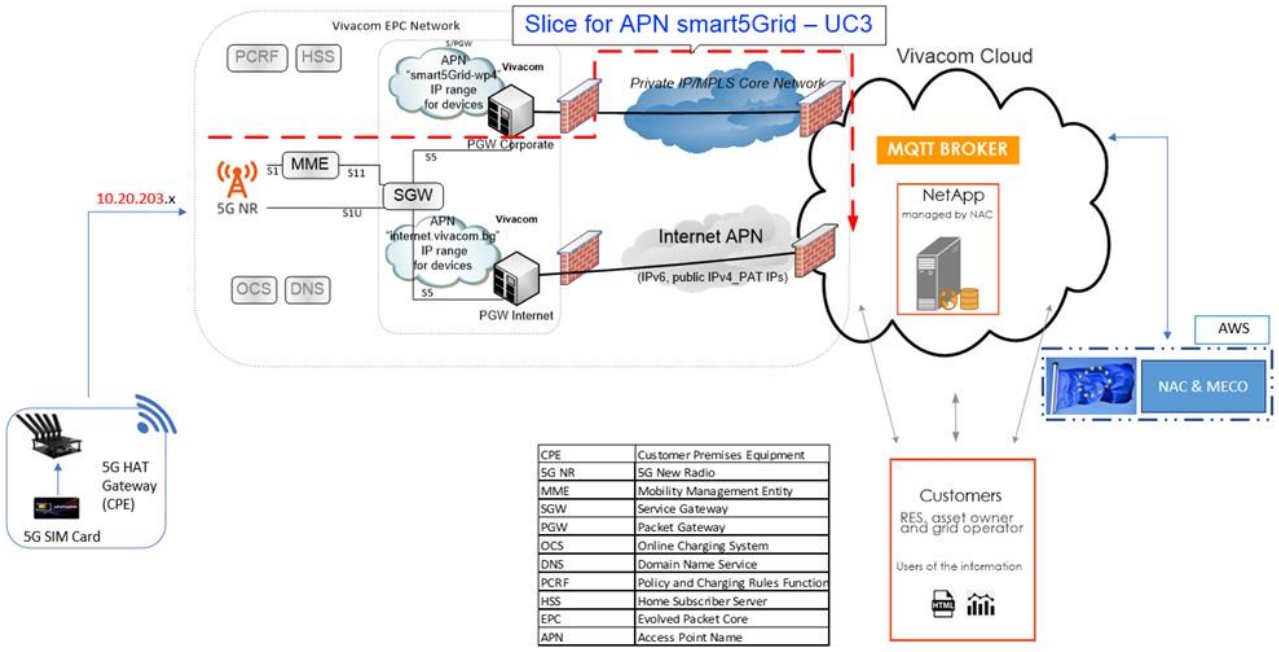

Figure 3-5: Infrastructure Low Level Design of UC3.

Traffic flow of user data is as follows:

1. Measurements from monitored devices (UEs) are sent by 5G CPE to 5G-gNB
2. 5G-gNB forwards the data via S1-U (user plane) interface to S-Gateway.
3. S-Gateway keeps context information such as parameters of the IP bearer, routing information, and stores the UE contexts when paging forwards the data to P-Gateway via S5 interface.

4. P-Gateway is applying policy enforcement, packet filtering, charging support, packet screening and forwards the data outside 3GPP network – in the current case this is VIVACOM Cloud. This is done via IP data links established via IP/MPLS fixed network of VIVACOM.
5. vRouters in VIVACOM Cloud forwards data to MQTT Broker service to corresponsive virtual machine based on destination IP of the packet.

In addition, the following configurations have been performed in the VIVACOM public core network:

- HLR and HSS have been configured with a new profile associated with the project's dedicated APN.
- SIM cards have been provisioned with 5G profile and permission to create a Packet Data Protocol (PDP) session via dedicated private APN.
- The APN configurations' DNS has been added to address the GGSN and the P-Gateway.

An IP/MPLS connection was established to connect the P-Gateway to the cloud infrastructure hosting the MQ Telemetry Transport (MQTT) broker and Network Application services.

This use case shares the same edge nodes used in UC4 and hosted in VIVACOM's data center in Sofia, Bulgaria. This data center is part of VIVACOM Cloud and is a virtual private cloud managed with Huawei Cloud Stack. Here VIVACOM reserved processing, memory, storage, and elastic IPs to create two edge nodes, each with a public IPv4 exposed to the public Internet. These nodes are manually configured and onboarded in the NAC as cloudlets.

3.4. Use Case 4

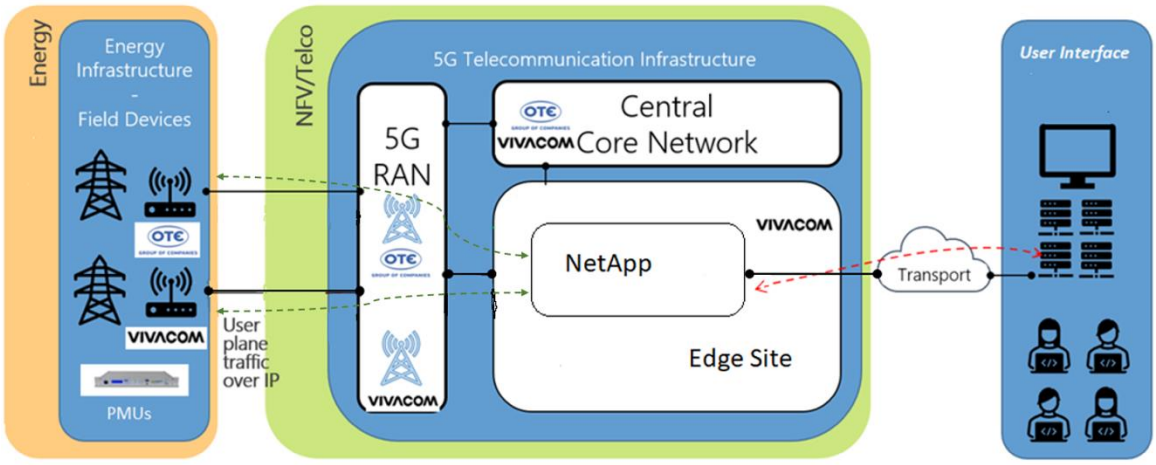

Figure 3-6: 5G Network Setup for UC#4 (Greek and Bulgarian Sides of Demo).

Figure 3-6 illustrates a high-level view of the network setup for both the Greek and Bulgarian sides of the demonstrational testbed of UC4. One of the key points of this setup is the involvement of two different telco operators in Greece (OTE) and Bulgaria (VIVACOM), connecting and forwarding the traffic measurements to the edge servers geographically hosted in Sofia (Bulgaria). On the left part of Figure 3-6 one can see the Energy Infrastructure and Energy Field Devices (PMUs) for the Demo; the PMU for

the Greek site is located at Thessaloniki IPTO's substation, and the PMU for the Bulgarian site is in Blagoevgrad ESO's substation. Both PMUs are covered adequately through the 5G RAN accordingly by OTE and VIVACOM's networks. The green dotted lines illustrate the traffic flow path for each site, initiating from the Substations through the 5G NSA networks and finally successfully reaching the edge/cloud server where the Network Application is hosted.

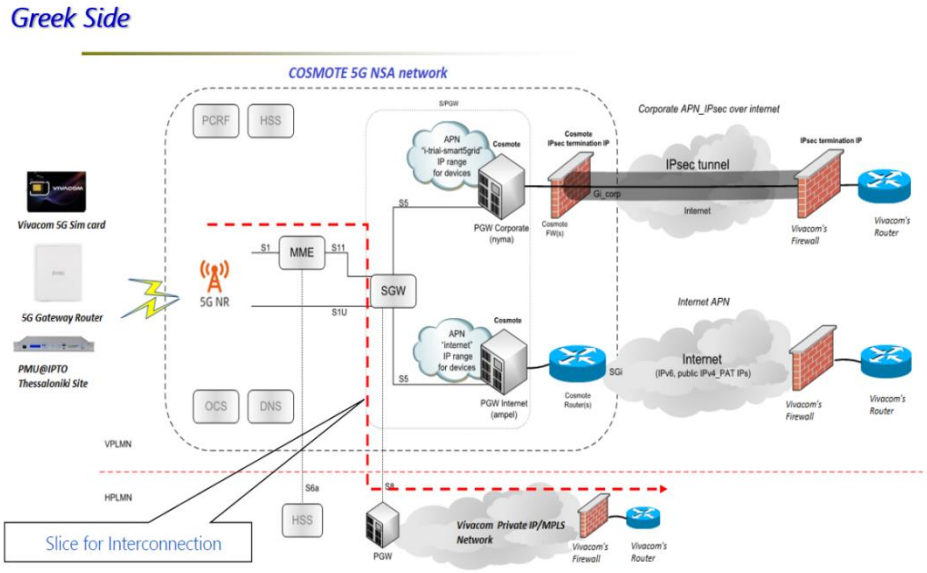

Figure 3-7: Infrastructure Low Level Design for Greek Side Demo (Roaming Scenario).

For the Greek site, as it is illustrated in Figure 3-7, it is decided to use a specific IMSI belonging to VIVACOM's telco network, thus exploiting the 5G commercial roaming agreement activated between the two telco operators. This decision was made between the two telco operators, since for the roaming case the S-Gateway of OTE's mobile network forwards directly the traffic towards the P-Gateway of VIVACOM, through the roaming interconnection path between the two operators. By means of this interconnection path, a minimum number of intermediary hops is used between the two networks, thus it is expected to minimize latencies. For the realisation of this scenario, VIVACOM has created the APN ("smart5grid-uc4") in the subscriber's profile, so that the core network forwards the traffic to the edge/cloud server according to the traffic rules implemented.

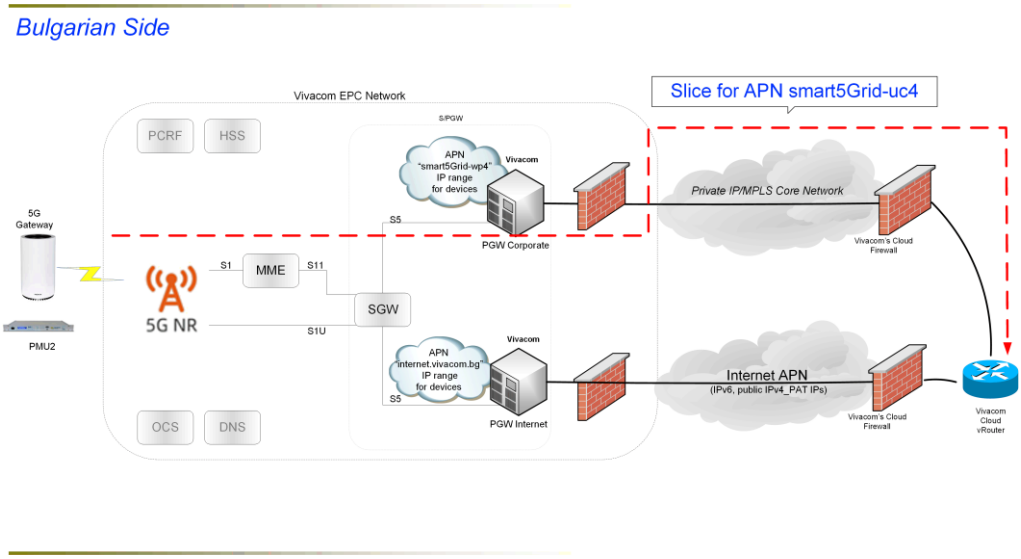

Figure 3-8: Infrastructure Low-Level Design for Bulgarian Side Demo.

For the Bulgarian site, whose low-level architecture is represented in Figure 3-8, an IMSI belonging to VIVACOMs network is used and configured with a specific APN ("smart5grid-uc4"). The specific APN is created in the subscriber's profile for the needs of the Bulgarian Side of the Demo and routes the traffic through the S/P-Gateway) towards the edge/cloud server where the Network Application is hosted.

Configuration illustrates a unified cross boundary telemetry data transmission solution. Both parts of the solution use the same APN, as well as a single link between the 5G NSA network and the cloud infrastructure that hosts Network Applications. The IP/MPLS part of the network guarantees a safe transfer of information and unites Layer 3 routing and Layer 2 switching advantages. MPLS allows using the Quality of Service (QoS) in order to prioritize the different types of traffic according to the level of its importance which assures a simultaneous work of the applications in real time and non-real time applications.

The advantages and characteristics of IP/MPLS can be summarized as follows:

- No limitations about the number of end points
- Static or dynamic routing
- A possibility to choose between private (RFC 1918), or public IP addressing
- A possibility of establishing an IPSec tunnel in the client's VPN
- Back-up and high availability options
- Online monitor tools for IP VPN ports and their specifications: Load, Traffic information

Moreover, the utilized PMU is a STERPMU-R1 rack-mounted synchrophasor measurement unit that includes a rechargeable battery backup, GPS for time, a 10/100Mbps Ethernet link for up to two simultaneous C37.118.2-2011 links, a GPRS modem for remote data reporting to PDC systems, controllable digital inputs and outputs, continuous waveform recording, harmonic analysis, and an SD card slot for local storage of 5 months of synchrophasor measurements [4].

In the Greek side of UC4, we are also using a Zyxel Gateway NR7101, built with industrial-grade components, allowing it to perform in severe environments with challenging signal strength or weather

issues such as high winds and rainstorms. NR7101 ensures high-quality connections and a longer device lifespan, allowing for commercial adoption across large geographic areas. NR7101 is dustproof and waterproof (IP-68 certified), and it provides better protection against harsh weather conditions such as lightning, hurricanes, and extreme temperatures. The NR7101 is compatible with both SA and NSA deployments. Under SA mode, the NR7101 may provide network slicing with specialized network capabilities that are critical to meeting various business objectives, such as dealing with complicated environments with massed wired and wireless networks [5].

On the Bulgarian side, instead, the gateway is a ZTE MC7010, a 5G outdoor FWA that can provide both high speed data service and voice service to mobile users. MC7010 supports data downlink speeds up to 3.8 Gbps and 542 Mbps for uplink. With built-in high-gain antenna, MC7010 provides wide service coverage and high data throughput. The MC7010 is also compatible with both stand-alone (SA) and non-standalone (NSA) deployments and its backwards compatible with 4G LTE networks [6].

Finally, this use case shares the same edge nodes used in UC3, and the reader can refer to Section 3.3 for more details. Finally, a more detailed view of the setup will be thoroughly explained at a later Deliverable D6.1.

4. Integration and Interfacing Between 5G and Energy Vertical's services

This section explains in detail the main joint between the Smart5Grid platform and the power grid infrastructure. It describes how the software stack sits on top of the vertical domain.

This integration in the final place is enabled by the deployment of the Network Applications and their related operations once in action.

Main enablers of this integration are i) the Network Application controller, which is in charge of orchestrating the Network Applications life cycle and ii) the OSR and V&V framework thanks to which the Network Applications are stored and retrieved once verified and validated.

The following subsection will put a strong emphasis on the interfaces developed among NAC, OSR and V&V leaving further specification for the complete definition of OSR in D3.3 and for the V&V in D4.2.

In this sense, each subsection, tailored on the UCs, introduces the architecture and the main functionalities of the Network Application Controller and then focuses on the interfaces needed to allow the deployment in the project infrastructures of the different Network Applications. The definition of these interfaces is followed by the technological choices and then by the report regarding the actual realization of the tests for the given interfaces.

4.1. Use Case 1, 3, and 4

For the UC1, UC3, and UC4, we have used Nearby Computing's NearbyOne as Network Application Controller and MEC Orchestrator. NearbyOne is a commercial product that provides end-to-end cross-platform edge-cloud orchestration and management of multi-site infrastructures. The usage of a multi-site orchestrator allowed the Smart5Grid partners to deploy one single instance of NearbyOne in the public cloud - https://smart5grid.nearbycomputing.com - to manage the infrastructure of all three use cases form: the MEC server used in UC1 and hosted in WI3's datacentre in Cagliari, Italy and the infrastructure used for both UC3 and UC4 in VIVACOM's premises in Sofia, Bulgaria.

The choice and adoption of a commercial solution that is being used by Telco operators and 5G Core providers in different countries provided a threefold advantage. First, it offered the opportunity to jump-start and focus the work on the interfaces with other component of the Smart5Grid architecture. Second, it brought insights and expertise in daily operation of 5G and edge environment and infrastructures. Third, it offered a cloud-native approach, which differs to the NAC of UC2, and led to the convenience of testing different approaches.

Finally, in this section we will discuss i) the changes that the commercial product had to undergo to operate Smart5Grid's Network Applications and its architecture; ii) the main functionalities offered by

NearbyOne; iii) the role of NearbyOne in the three use cases; and iv) the interfaces with other Smart5Grid components.

4.1.1. Architecture and Functionalities

NearbyOne's internal architecture for Smart5Grid can be resumed as in Figure 4-1. This is the usual NearbyOne architecture with the addition of a Network Application adapter (purple rectangle) that is a new component created specifically for the Smart5Grid project. While we describe the low-level details of the Network Application adapter below in Section 4.2.1, due to intellectual property rights we cannot do the same for the rest of the components. However, these components reflect those defined in Section 4.4.2.1 Network Application Controller & MEC Orchestrator - System Level Components of D2.2 [3].

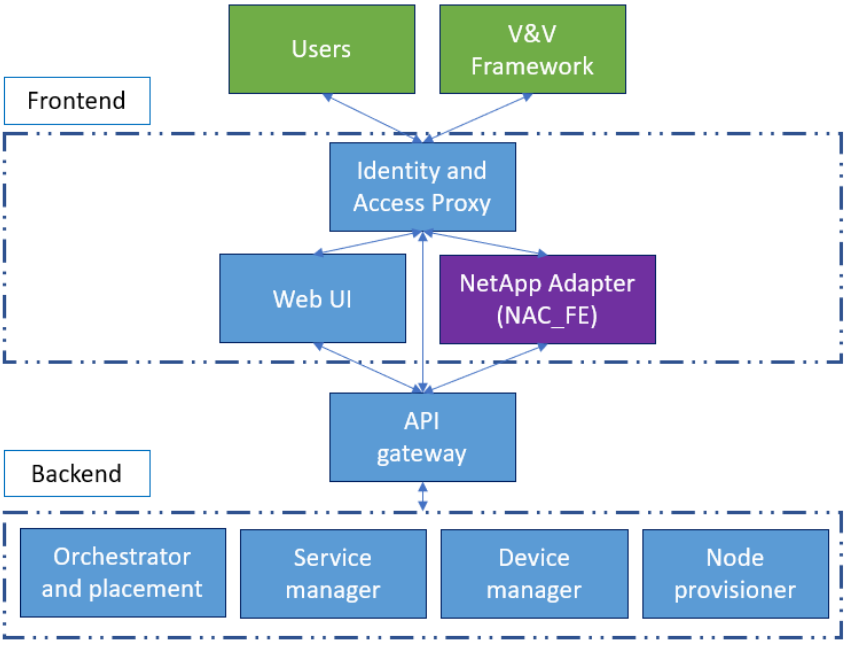

Figure 4-1: NearbyOne Internal Architecture.

As a NAC and MECO, NearbyOne offers functionalities to manage the lifecycle of Network Applications and to add edge or cloud infrastructure in the orchestration – either provision new nodes or onboarding existing infrastructure. Furthermore, it also offers basic features to manage users, organizations, and other settings. Below we offer an overview of these functionalities providing a description and screenshots divided by topics.

Login and User Management

The first view offered by NearbyOne and illustrated in Figure 4-2, is a normal login page where users can either log into the platform or recover their password. Once logged into the platform, users are presented a page with a top bar representing the user and the instance of the platform, and a simple lateral main menu with a right menu. Using this main menu, users can access the Settings page where they can edit the various settings including the user management (Figure 4-3)

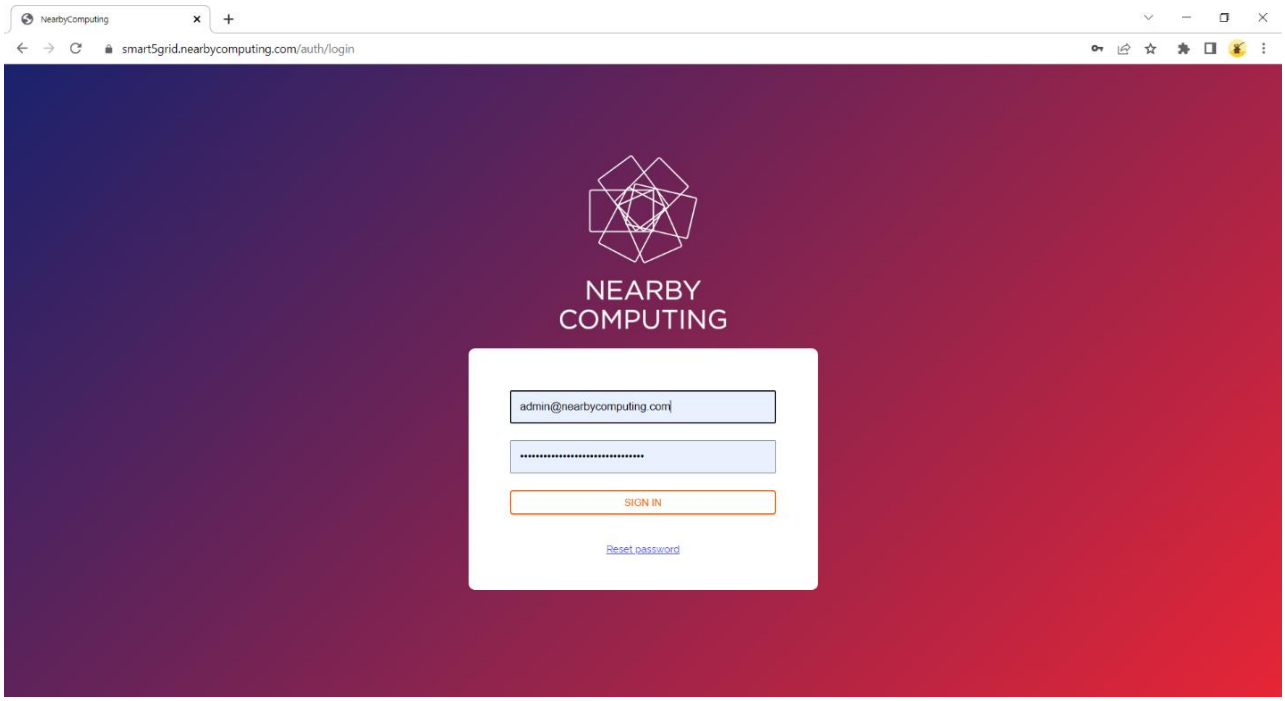

Figure 4-2: NearbyOne Login Page.

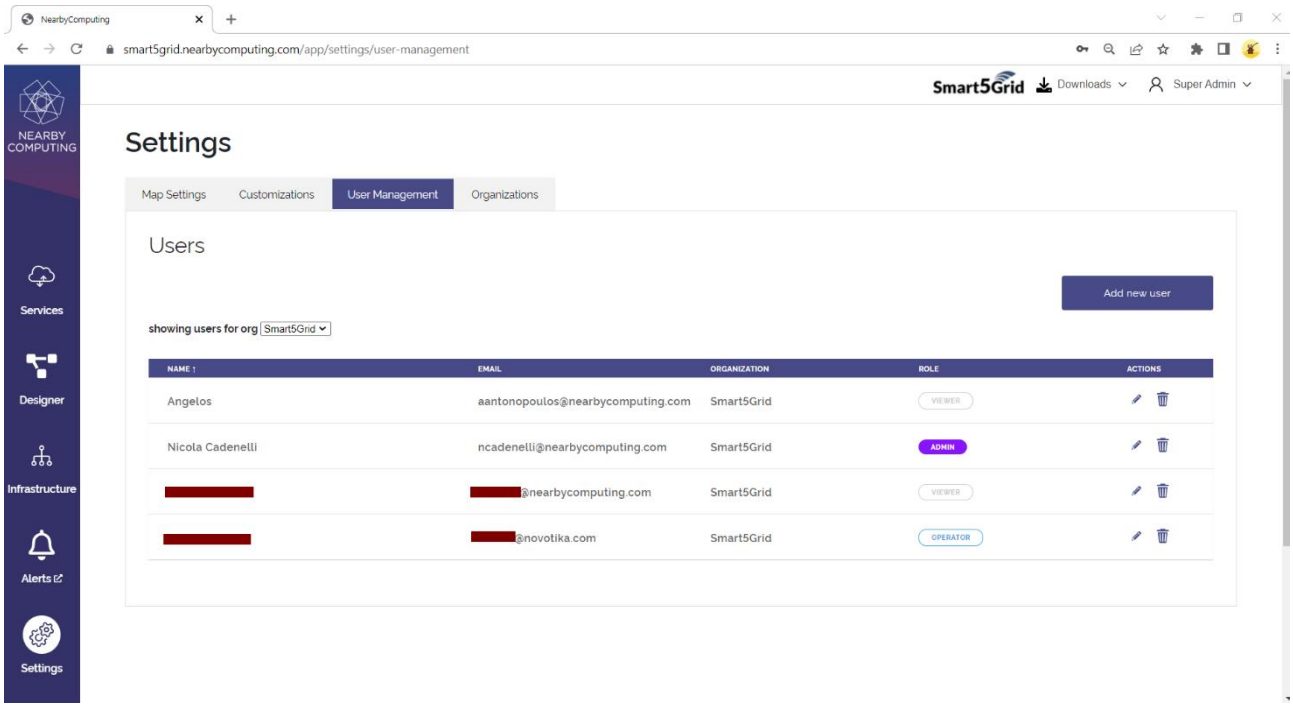

Figure 4-3: NearbyOne Setting View - User Management.

Infrastructure

To manage the infrastructure, NearbyOne offers one dedicated page, see Figure 4-4, to show the overview of the infrastructure using a list of all sites in the organization (left side) and a map with all the

nodes onboarded (right side). When multiple nodes are closed geographically, the map shows clickable bubbles to zoom in to a level comfortable to see each single node.

On the other hand, the Inventory view of the infrastructure, depicted in Figure 4-5, offers a table view of the infrastructure with the possibility to edit, remove and even add new nodes. Here users, can add new resources, see Figure 4-6. NearbyOne has the notion of different resources but for Smart5Grid users would either add existing Kubernetes Clusters provisioned by third parties as *Cloudlet* or new nodes using the near Zero-Touch Provisioning (nZTP) where NearbyOne would install all the software stack, from OS to Kubernetes, of the nodes.

Finally, the user interface also offers the possibility to edit, review, and remove resources in the infrastructure.

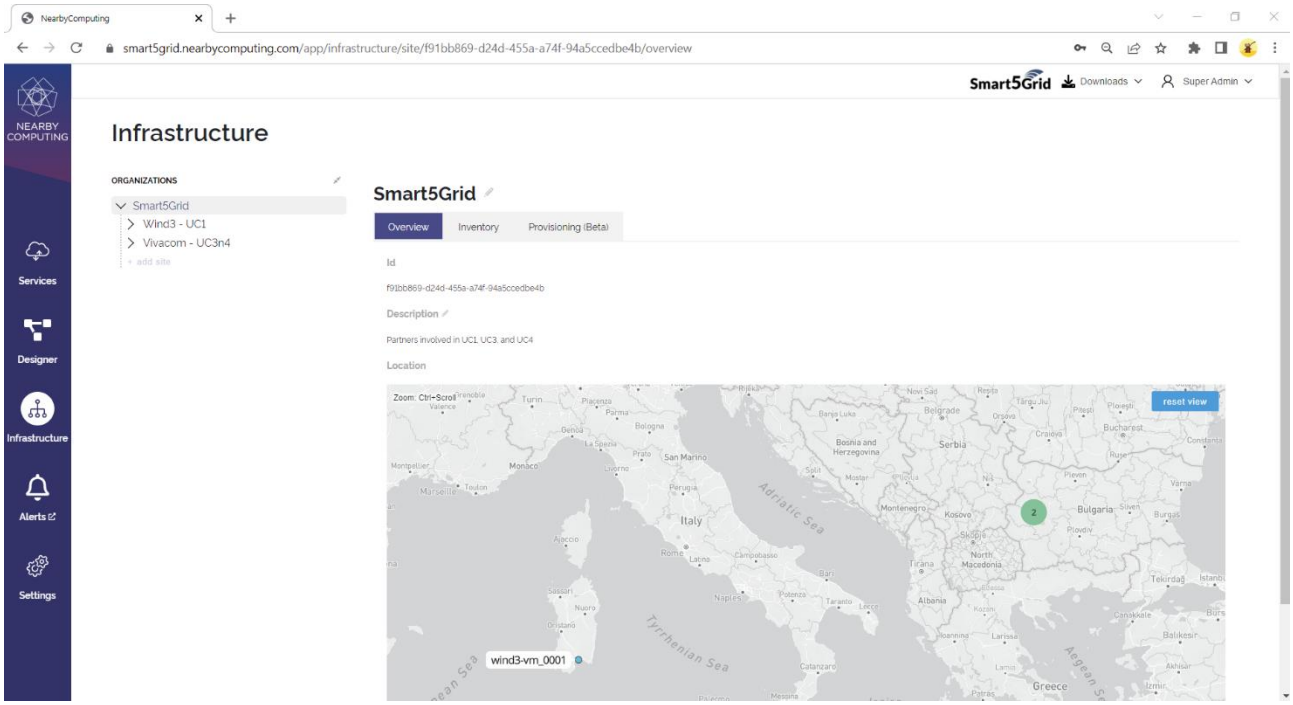

Figure 4-4: NearbyOne Infrastructure View – Overview.

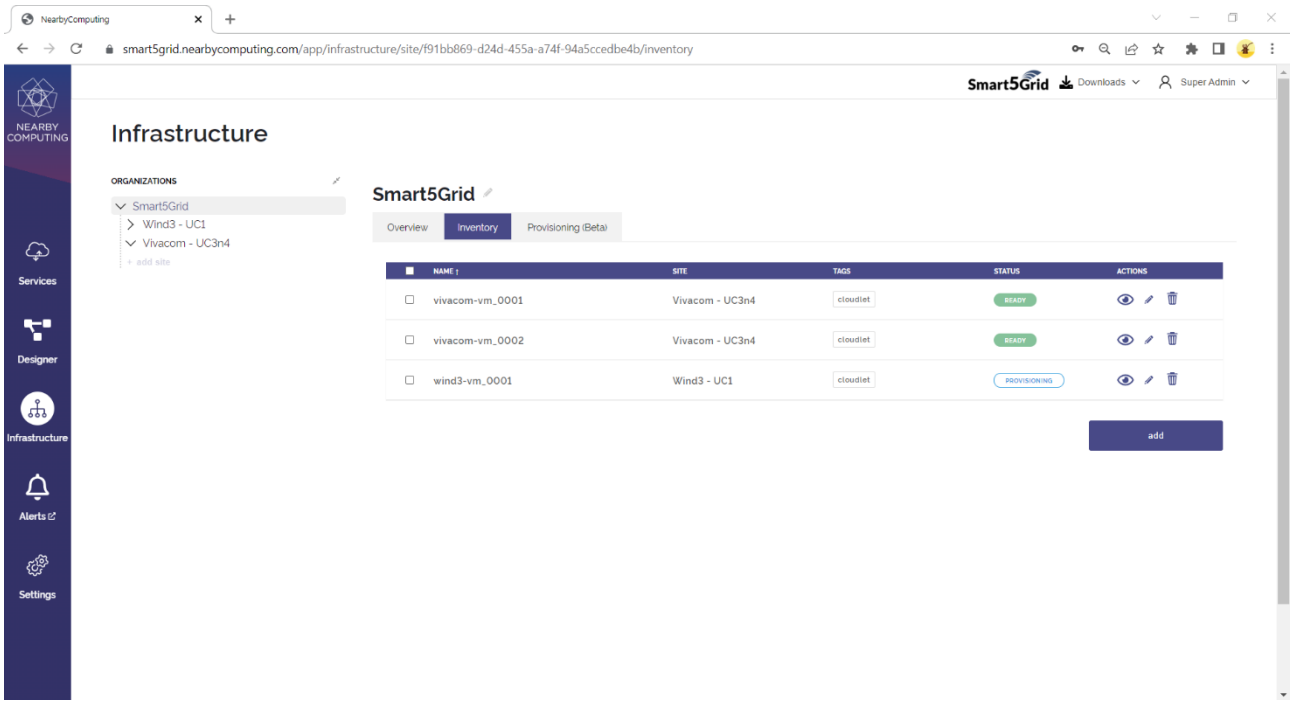

Figure 4-5: NearbyOne Infrastructure View – Inventory.

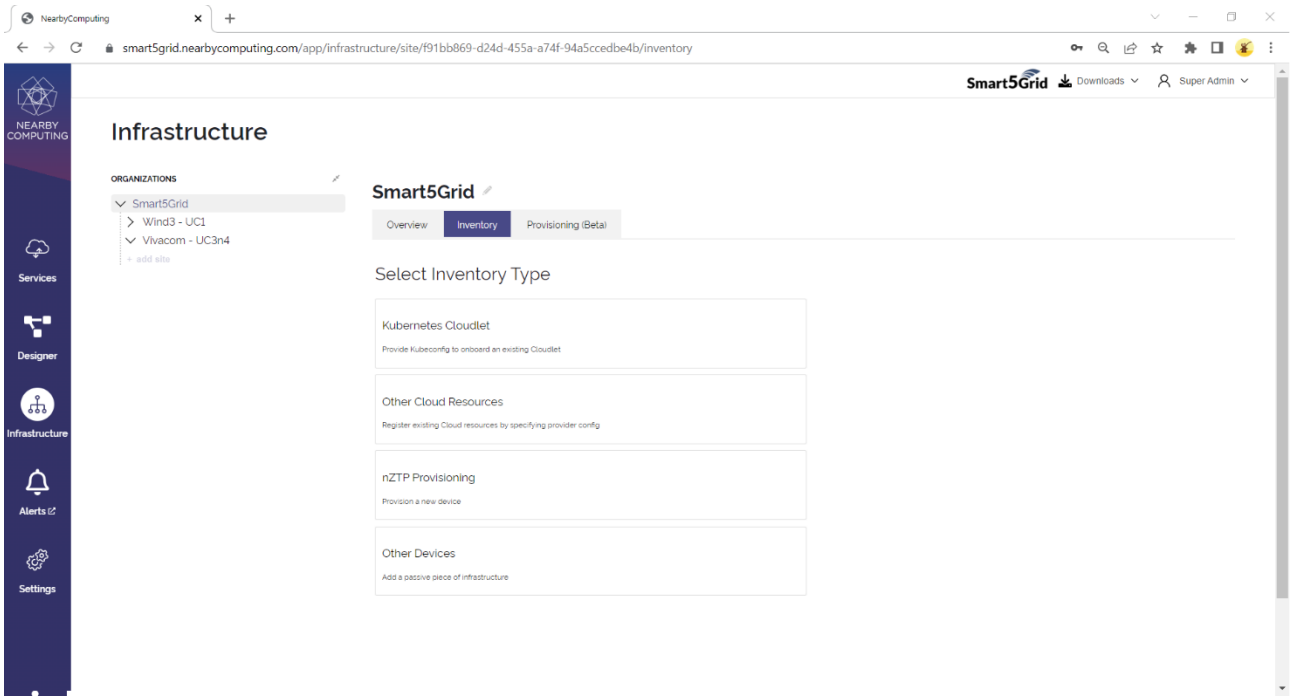

Figure 4-6: NearbyOne Infrastructure View – Add a new Cluster, Node, or other devices.

Designer

The second main view is the Designer view, where users can see, edit, and design a Network Application deployment. Figure 4-7 reports the Market place tab of the Designer view, where users can see all the onboarded Network Applications in this instance of NearbyOne. In this tab, users can filter, preview, and add Network Applications to be deployed. Network Applications added to be deployed can be configured in the designer tab (Figure 4-8). Here is where users can edit the default settings for each

service in the Network Application; allowing them to customize every instance of a Network Applications. Once the user is satisfied, the deployment tab illustrated in Figure 4-9 can be used to add a user-friendly name and finally deploy the customized instance of the Network Application.

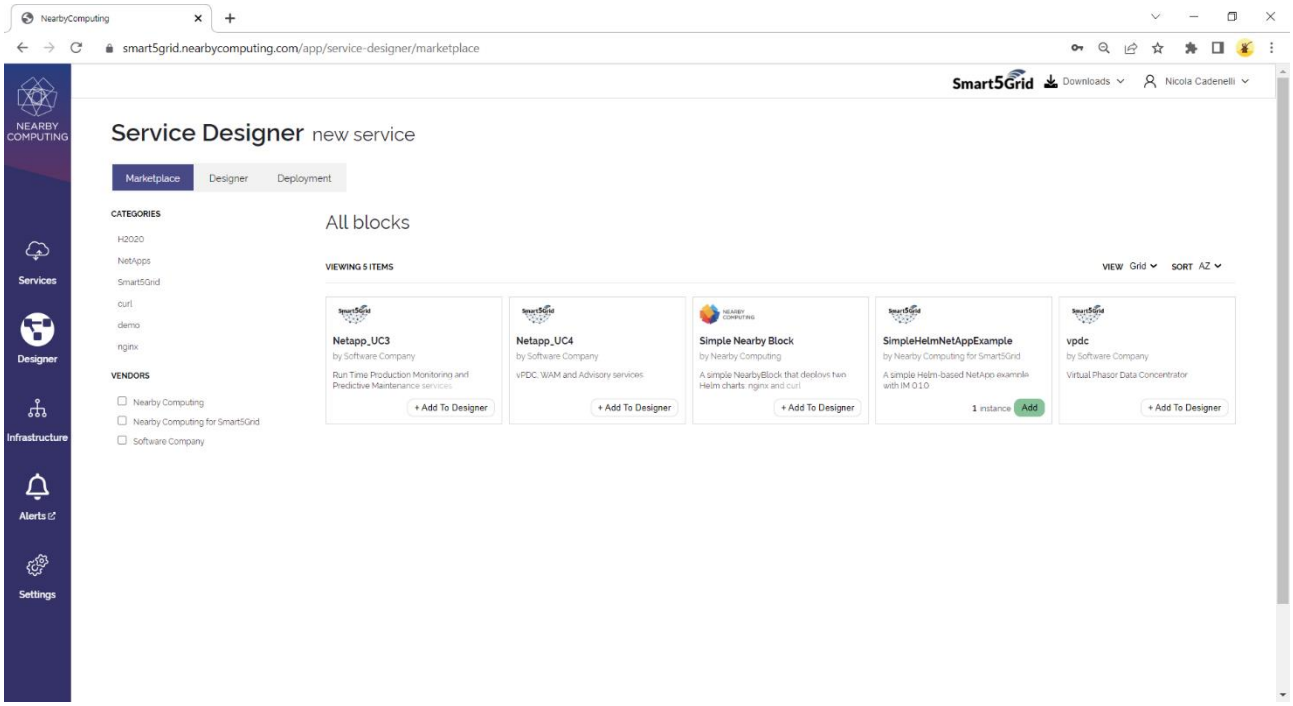

Figure 4-7: NearbyOne Service Designer View – Marketplace.

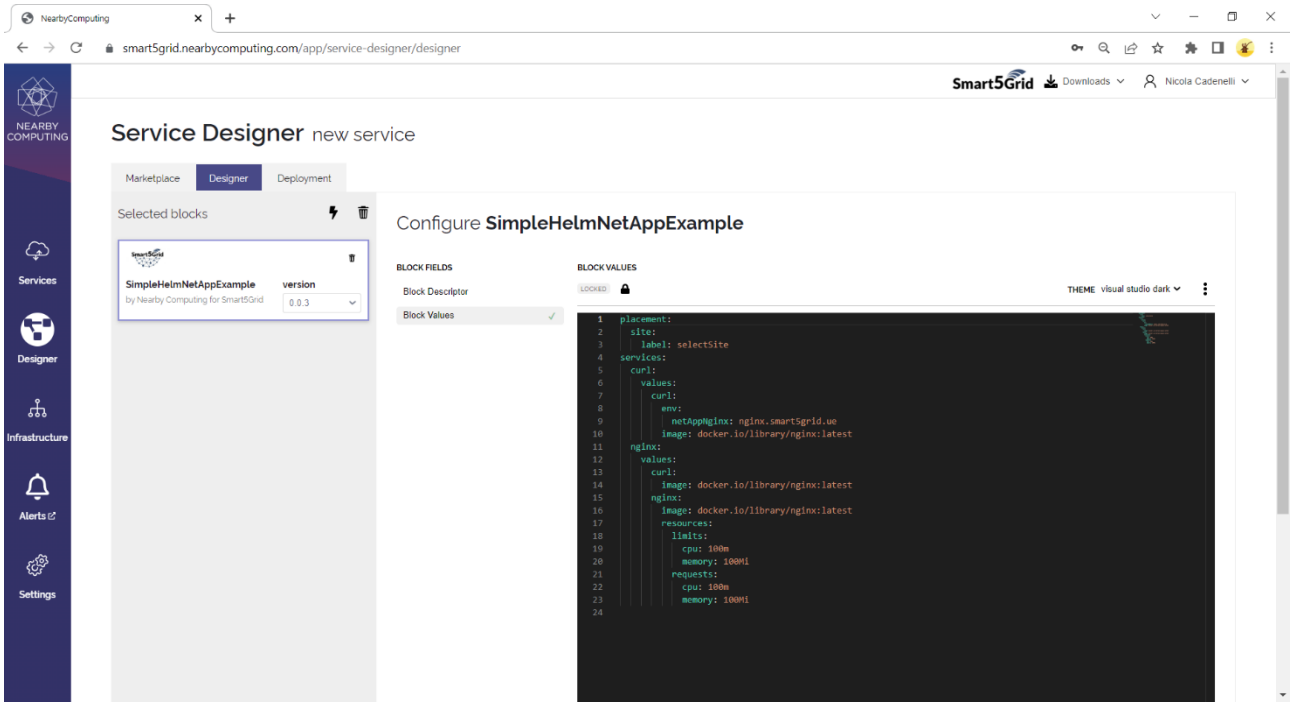

Figure 4-8: NearbyOne Service Designer View – Designing a service to be deployed.

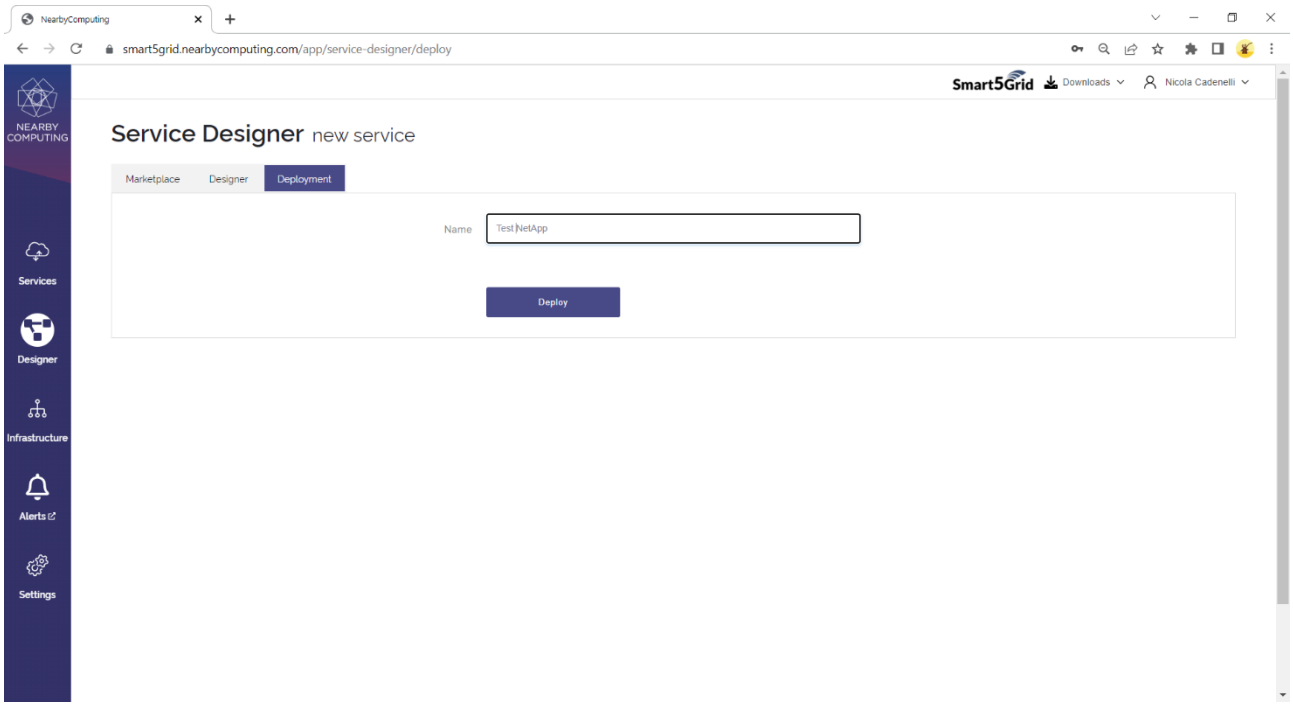

Figure 4-9: NearbyOne Service Designer View – Deploying a service.

Services

The Services page reported by Figure 4-10, is the main page to check the status of every service instance. Here, users can check the details of every instance, edit an existing instance, and delete running instance, see Figure 4-10 and Figure 4-11.

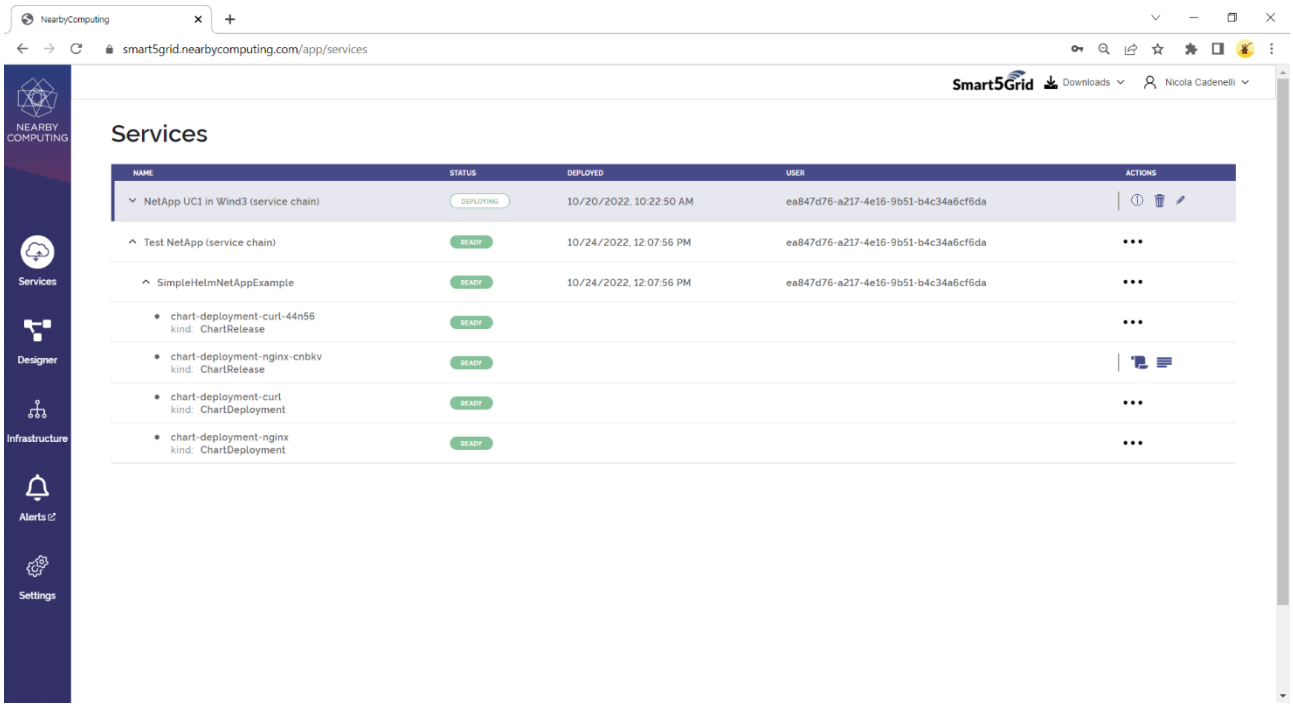

Figure 4-10: NearbyOne Service View – Show deployed services.

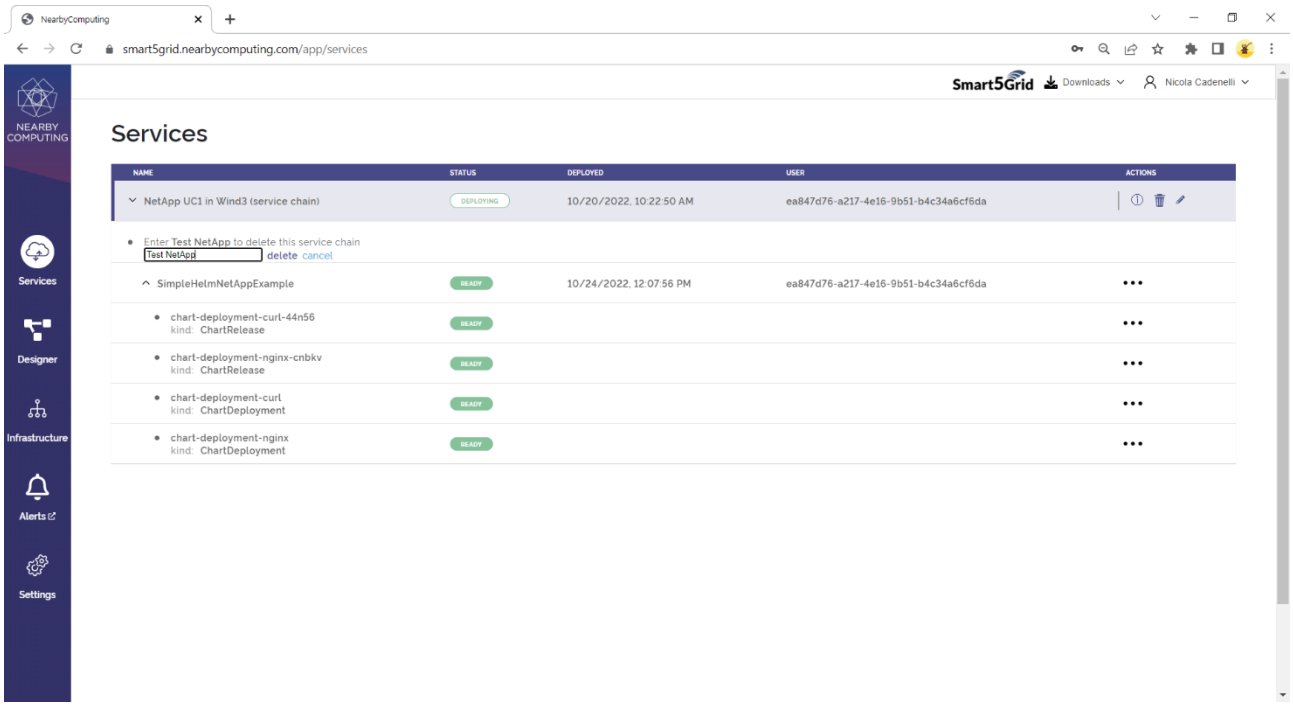

Figure 4-11: NearbyOne Services View - Deleting a service.

4.1.2. Interfaces with the Smart5Grid Architecture and Network Application concept

In all three use cases, NearbyOne's role is to act as the NAC and MECO. This means that most of the interfaces regarding the NAC and other components of the Smart5Grid infrastructure need to be provided. Table 2 offers a reminder of these interfaces and reports their support by NearbyOne. Next, we describe how NearbyOne implements these NAC interfaces of Smart5Grid and see the integration tests done validate them.

Table 2: NearbyOne's support for Smart5Grid Interfaces

Interface	Method	NearbyOne Support (Standard or reference)
NAC_FE	Onboard and (un)deploy Network App	Developed (Smart5Grid)
NetApps NAC	Scrape Network Apps' metrics	Developed (Smart5Grid)
OSR_NAC	Download helm charts from the OSR	Natively Supported (OCI)
OSR_NAC	Download docker images from the OSR	Delegated to the NFVO (OCI)
NFVO_NAC	Manage NFV deployments	Natively Supported (Kubernetes)
VIM_NAC	Node provisioning and registration	Natively Supported (NearbyOne)
Nodes_NAC	Scrape nodes' metrics	Natively Supported (PromQL)
AF_NAC	Set traffic rules in the 5G Core	Unauthorized by 5GCore Providers
SM_NAC	Set network slices in the 5G Core and Network	Unauthorized by 5GCore Providers

Since the concept of a Smart5Grid Network Applications, and its descriptor, is one of the key contributions of this project, there was no solution that could offer native support to Smart5Grid's

Network Application concept. Thus, we needed to extend NearbyOne to create the NAC_FE interface. This way, NearbyOne is now able to onboard Network Application using the Network Application descriptor for Helm charts of Smart5Grid initially described in D2.2 [3] and then refined in its the reference repository [7]. For this interface we defined this [8] REST API interface using Swagger 2.0. This interface exposes two paths. One to on-board, retrieve, and list Network Applications and a second to deploy Network Application instances, retrieve their status, and delete them. The NearbyOne adapter that implements this interface is written in Golang and it is deployed together with the public instance of NearbyOne, like depicted in Figure 4-12, and protected using API Keys and HTTPS.

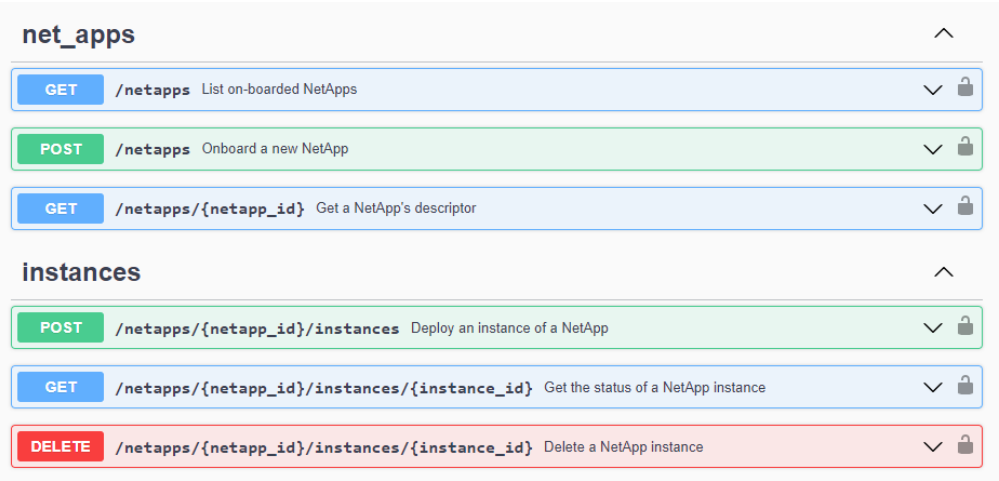

Figure 4-12: NearbyOne NAC_FE REST Interface.

Like NAC_FE, the NetApps_NAC interface and the ability to scrape the metrics of a Network Application's instance from its monitoring endpoint is also something specific to this project. Thus, NearbyOne had to be adapted and extended to support it. Since Smart5Grid decided to rely on Prometheus and its PromQL query language [9], whenever a Network Application with a monitoring endpoint and at least one SLOs (see Smart5Grid Network Application Descriptor) is deployed, NearbyOne will also deploy a monitoring agent alongside the Network Application. This agent will constantly query monitoring endpoint and inject into the orchestrator the result of such query. Finally, such results will be used by the orchestrator to enforce lifecycle decision and edit the current service's instance, as shown in Figure 4-13.

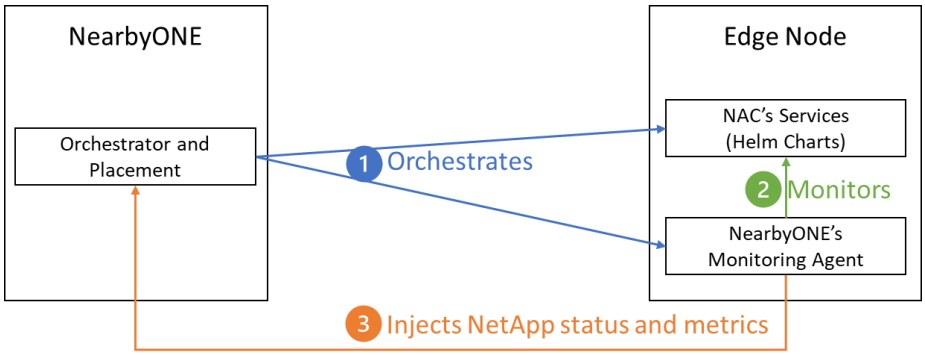

Figure 4-13: NearbyOne's Monitoring Agent.

On another hand, the fact that the consortium used a commercial product that already existed at the beginning of the project, meant that some interfaces where already supported. Particularly for those interfaces that use de-facto standards or broadly accepted solutions. For instance, this is the case of the OSR_NAC's method to pull Helm charts, here the OSR adopted the interface defined by the Open Containers Initiative (OCI), and NearbyOne already supported it.

For the NFVO_NAC and the Nodes_NAC interfaces, NearbyOne already provided native support for Kubernetes clusters, which meant that these interfaces could be reused as is. Furthermore, since Kubernetes provides native support to download docker images, we delegated this method of the OSR_NAC interface, to this component. Besides, NearbyOne already embedded the responsibility of a Virtual Infrastructure Manager (VIM). Thus, the VIM_NAC interface is embedded and is part of the product.

Finally, since the UC infrastructure is deployed in the production environment of national Telcos, some limitations and concerns apply. For security reasons, the Telco partners don't want to expose the critical interfaces of their production 5G Core to influence slices and traffic routes, even for testing purposes. This consideration is valid for all three use cases, and it implied that neither the AF_NAC nor the SM_NAC interfaces are authorized and thus, not required.

4.1.3. Integration Tests

To test the integration of NearbyOne as Smart5Grid NAC, we planned and carried out the following set of tests:

- Network App Onboarding
- Network App Deployment
- Network App Undeployment
- Network App Migration
- Network App Scaling

These tests are meant to evaluate all the Smart5Grid interfaces regarding the NAC supported by NearbyOne and reported in Table 3.

Table 3: Integration tests UC 1, 3, and 4

Name	Network App Onboarding	Locations	Internet and Vivacom DC
Component under test	• V&V Framework • NAC_FE • NAC	Responsible	Ubiwhere and Nearby Computing
Test environment		NAC (NearbyOne), a test Network App, and UC3 and UC4 edge nodes.	
Feature under test		• The NAC_FE interface to onboard a Network App • NearbyOne NAC adapter to correctly translate a Network App Descriptor into the internal format of NearbyOne	
Preparation		1. NearbyOne instance deployed with the NAC_FE	

	interface ready and an API KEY for V&V framework defined and shared 2. UC3 and 4 edge nodes onboarded and operative 3. V&V Framework is deployed
Dependencies	N/A
Steps	1. The V&V framework perform a POST REST request to NearbyOne endpoint passing the Network App Descriptor 2. NearbyOne Network App Adapter parses the descriptor and translates the Network App in the internal representation 3. The Network App is onboarded and NearbyOne return to the V&V Framework an HTTP 200 status code
Pass Criteria	All steps are carried out without errors
Result	Successful

Name	Network App Deployment	Locations	Internet and Vivacom DC
Component under test	• NAC • NFVO_NAC • OSR_NAC • Nodes_NAC	Responsible	Nearby Computing
Test environment		NAC (NearbyOne), OSR *, a test Network App, and UC3 and UC4 edge nodes.	
Feature under test		• The NAC, its orchestrator, and the NFVO_NAC interface to deploy Network App services • The Nodes_NAC interface used by the NAC's orchestrator to select running nodes • The OSR_NAC to download the Helm charts and docker images	
Preparation		1. The OSR instance is operative (-registry.s5g.gos.y-cloud.eu) and we have permission to pull artifacts. 2. Login into NearbyOne interface with an existing user	
Dependencies		Network App Onboarding	
Steps		1. The user adds the Network App to the designer and deploys it in UC3 and UC4 site. 2. The NAC picks one of the nodes in the site that is active (Nodes_NAC). 3. The NAC fetches the Helm charts of the Network App services from the OSR (OSR_NAC) 4. The NAC deploys the helm charts in the node's	

	NFVO (NFVO_NAC) 5. The NFVO deploys the Network App pulling the docker images from the OSR (OSR_NAC) 6. The NFVO_NAC API reports a ready status (NFVO_NAC) 7. The NAC UI reports a successful status.
Pass Criteria	All steps are carried out without errors
Result	Successful

Name	Network App Undeployment	Locations	Internet and Vivacom DC
Component under test	• NAC • NFVO_NAC	Responsible	Nearby Computing
Test environment		NAC (NearbyOne), a test Network App, and UC3 and UC4 edge nodes.	
Feature under test		The NAC and the NFVO_NAC interface to undeploy Network Application services	
Preparation		1. Login into NearbyOne interface with an existing user	
Dependencies		Network App Deployment	
Steps		1. From the NAC UI, we delete the Network App service. 2. The NAC forwards the command to the NFVO to undeploy all Network App services (NFVO_NAC) 3. The services are stopped. 4. The NFVO API doesn't show the services anymore (NFVO_NAC) 5. The NAC UI doesn't report the services anymore.	
Pass Criteria		All steps are carried out without errors	
Result		Successful	

Name	Network App Migration	Locations	Internet and Vivacom DC
Component under test	• NAC • Nodes_NAC	Responsible	Nearby Computing
Test environment		NAC (NearbyOne), a test Network App, and UC3 and UC4 edge nodes.	
Feature under test		• The Nodes _NAC interface for the NAC to be able to scrape the lifeless of a node. • The NAC orchestrator to migrate the Network App's services to a new active node	
Preparation		1. Login into NearbyOne interface with an existing user 2. Have access to the edge node where Network App services are deployed	

Dependencies	Network App Deployment
Steps	1. We manually cause the shutdown of the node where the Network App services are running. 2. The NAC detects that the node is no longer active. (Nodes_NAC) 3. The NAC finds a new placement in the second node, which is active, and redeploys the Network App services exactly as in the step 2 to 7 of the Network App Deployment
Pass Criteria	All steps are carried out without errors
Result	Successful

Name	Network App Scaling	Locations	Internet and Vivacom DC
Component under test	• NAC • NAC_FE • Nodes_NAC	Responsible	Nearby Computing
Test environment		NAC (NearbyOne), a test Network App, and UC3 and UC4 edge nodes. In this case the test Network App has the following requirements: 1. Its descriptor has a monitoring endpoint and at least one SLO with a scaling action – See 4.6 Smart5Grid Network Application Descriptor of D2.2 [3]. 2. The Network App service exposing the metrics will systematically publish a metric that will trigger a scaling action after 60 second that it's running. Together with the metrics, the Network App will also expose the overwrite configuration to be used when scaling.	
Feature under test		The Network App_NAC interface for the NAC to be able to scrape the metrics of a deployed Network App instance.	
Preparation		1. NearbyOne instance ready and with the UC3 and 4 edge nodes onboarded. 2. An instance of the test Network Application that Network Application exposes a Prometheus endpoint is deployed successfully.	
Dependencies		Network App Deployment	
Steps		1. Deploy the test Network Application with a monitoring endpoint and the SLO. NearbyOne will automatically also deploy its monitoring agent that will inject the Network Application metrics into the orchestrator. 2. Wait for the Network App to publish the	

	metric value that will trigger the scaling action. 3. When the Network App SLO metric is invalid, the orchestrator will react by redeploying the Service using the overwritten parameters requested by the Network App.
Pass Criteria	All steps are carried out without errors
Result	Successful

4.2. Use Case 2

4.2.1. Architecture and Functionalities

A major update with respect to D3.1 has been provided with the creation of an ad hoc Network Application controller dedicated to UC2. The aim of providing this component is to broaden the area of intervention of Smart5grid adding a NAC designed from scratch capable of addressing both the approaches of cloud native as well as telco world to manage Network App. To cope with these approaches, NAC will be deployed in a private cloud which, through internet access, communicates with Neutroon platform (deployed in a public cloud) for the deployment of Network Applications. Neutroon is a 5G private network provider which provides the Slice Manager, RAN Controller, NFVO, NFVI and 5G Core Controller components that are part of the Smart5Grid architectural framework. For this purpose, the NAC exposes through the Internet the northbound interfaces to be used by end users or external components (e.g., OSR and V&V) and implements southbound interfaces which, through Internet as well, are used to orchestrate applications deployed in the network infrastructure. In this section, the functionalities of the subcomponents that comprise Network Application Controller for UC2 are discussed and a detailed description of the interfaces implemented is presented.

The architecture of NAC is described by Figure 4-14 and can be summarized by the following main parts/blocks.

1. NBI: Exposes methods/information to verticals
2. LCM: in charge of coordinate the Network Application lifecycle
3. Adapters: Plugins for interaction with orchestrator components
4. Local Registry: Storing of Network Application Packages and onboarding of NSDs/VNFDs
5. Telemetry component: Support the LCM in decision for smart placement of Network Applications

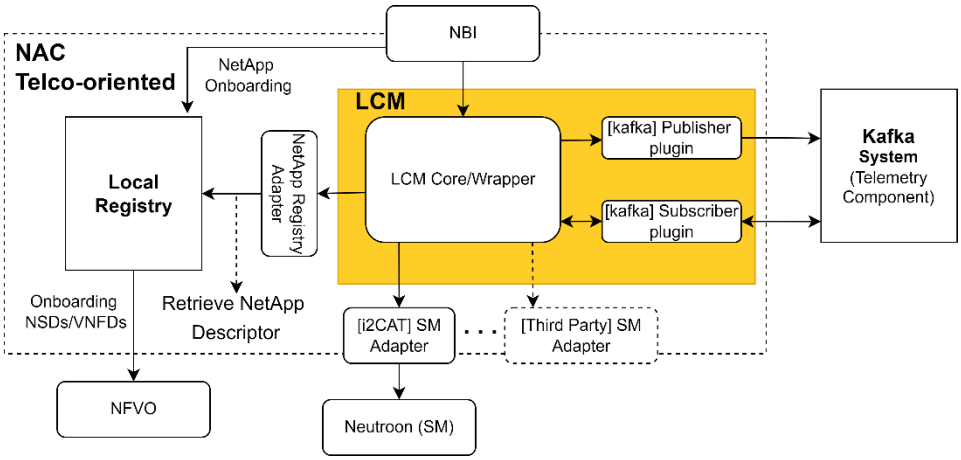

Figure 4-14: Network Application controller components.

In the following subsections the detailed descriptions of the different subcomponents of NAC are provided:

4.2.1.1. Life Cycle Manager

This component is the core part of the Network Application Controller. It handles the whole lifecycle stages of the Network Applications. LCM is in charge of create, deploy, upgrade and delete Network Applications over the 5G Platform. In UC2 the 5G Platform is provided by a 5G Private Network Provider called Neutroon which implements most of the components of the Telco and Infra Layer of the Smart5Grid platform, such as: Slice Manager, RAN Controller, VIM, NFVO, RAN, Edge Server and 5G Core Network. The main functionality of LCM is to extract the Network Application information filled into the Network Application Descriptor and translate in terms of calls based on API REST methods implemented in different adapters. Basically, LCM comprises three main submodules: northbound API, southbound API (adapters), and wrapper. Below, the description of each submodule is presented:

- Northbound API: This module exposes the methods to external components and users to create, instantiate, read, and delete Network Applications (Figure 4-15). NAC's NBI component sends the name of the Network Application Descriptor through the LCM's Northbound API during the instantiation process, then the LCM should search the descriptor inside of the Local Registry component.

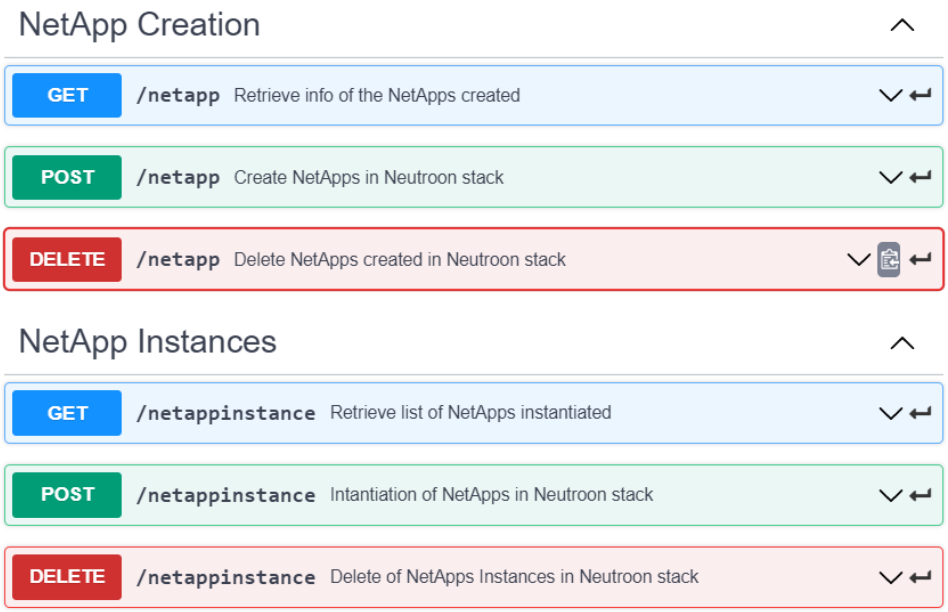

Figure 4-15: LCM's NBI definition.

- Wrapper: The main objective of this module is to read and extract the Network Application information fulfilled into the Descriptor. Once the main Network Application information is obtained, the LCM creates the calls by using the adapters to communicate with the 5G Platform.

4.2.1.2. Adapters

The Network Application Controller includes two main adapters to deploy Network Applications over the 5G Platform:

- Local Registry Adapter: LCM via Local Registry Adapter interacts with Local Registry component during the Network Application Instantiation process. This module is used to retrieve the Network Application descriptors from the Local Registry component that should be deployed over the 5G platform. Basically, it implements the basic REST calls to request the descriptor to the Local Registry component.
- Neutroon Adapter: This module allows the Network Application Controller to interact with 5G Platform to create, instantiate and delete Network Applications. It implements some REST methods only for the creation and deployment of Network Applications. However, 5G Platform's NBI exposes more methods regarding the creation of devices and slices which are out of the scope of the Network Application Descriptor.

4.2.1.3. Local Registry

The local registry is responsible for supporting the onboarding of the Network Application package to the Network Application controller, so that the Network Application can be deployed. The main functionalities are:

- API: exposes the different endpoints, which allow the NBI to incorporate a Network Application package with all subcomponents; it can list the Network Application packages according to the

chosen type of package it can return the Network Application descriptor if requested by the LCM to perform the Network Application deployment, and also the update and deletion of the packages. The API is provided as an OpenAPI specification with the endpoints from Figure 4-16.Figure 4-16

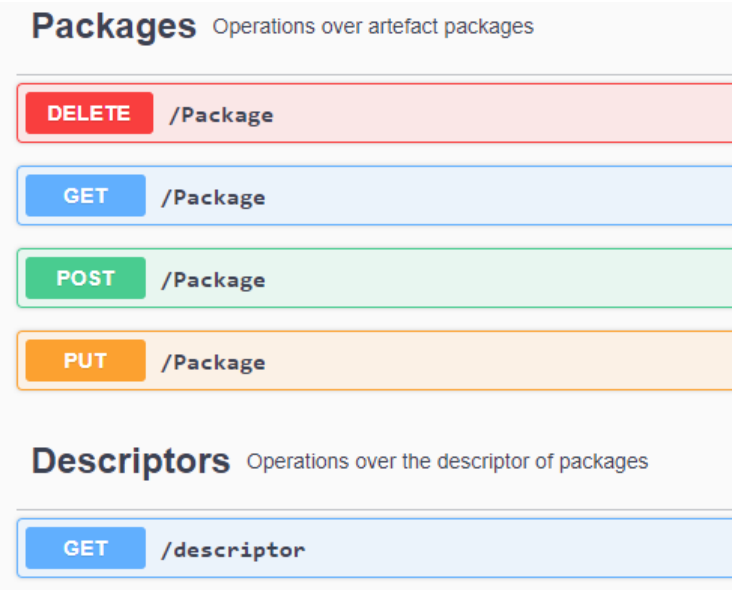

Figure 4-16 Local Registry API definitions

- Storage: supports storing and maintaining all types of artefacts involved in a 5G deployment, including Network Function Descriptors and Packages (Network Application, NSD, VNFD), Helm Charts and Container Images. Each of these packages will be stored and exposed using the appropriate service according to the package type as shown in Figure 4-17.
- Provisioning: the system is capable of provisioning a 5G infrastructure by configuring the full NFV MANO stack to be able to access those devices and onboarded directly in the NFVO the NSDs and VNFDs.

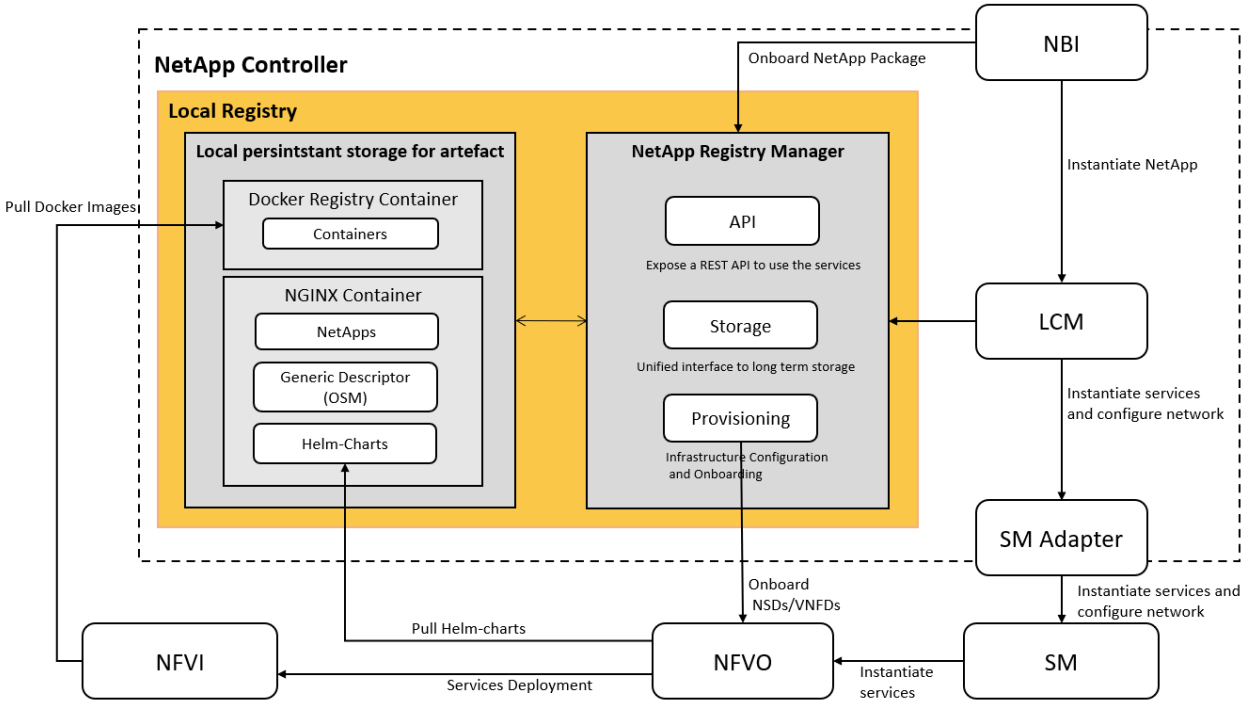

Figure 4-17: Local Registry.

Figure 4-17 shows the overall functional blocks, services, and interfaces of the local registry. From the Network Application Controller side and UC2, the local registry interacts with the two main components while it also needs to have access to the underlaying NFVI and NFVO to assist the Controller correctly

First, the NBI interacts with the local registry for the onboarding of the Network Application package. This package can be received in several forms, The simplest way in which all of the sub-packages required to instantiate the Network Application (Docker images –optional-, Helm chart, VNFs, NSs) are received at once bundled inside the Network Application artefact (.tar.gz file). Each of these packages will be extracted and uploaded to its the corresponding repository (Repositories block from Figure 4-17Figure 4-17). As part of the provisioning stage of the local registry, the NFVO is configured to use the required exposed repositories and the VNFs and NSs packages are onboarded.

Second, the LCM interacts with the local registry at instantiation time when a new Network Application is to be installed. The LCM will request the Network Application descriptor from the local registry and start the installation by pointing the SM Adapter to the NFVO to deploy the services it has referenced in the Network Application descriptor. Thanks to the provisioning stage carried out earlier, the NFVO contains the required NS and VNF descriptors and is able to pull the helm-charts that are referenced in the VNFs. This process is what enables the deployment of the VNFs in the VIM.

4.2.1.4. Telemetry Component

The telemetry component, composed of a particular instantiation of the software defined in D3.1, supports the LCM in the smart placement of Network Applications, depending on the environmental and contextual information. It comprises different components as shown in the Figure 4-18.

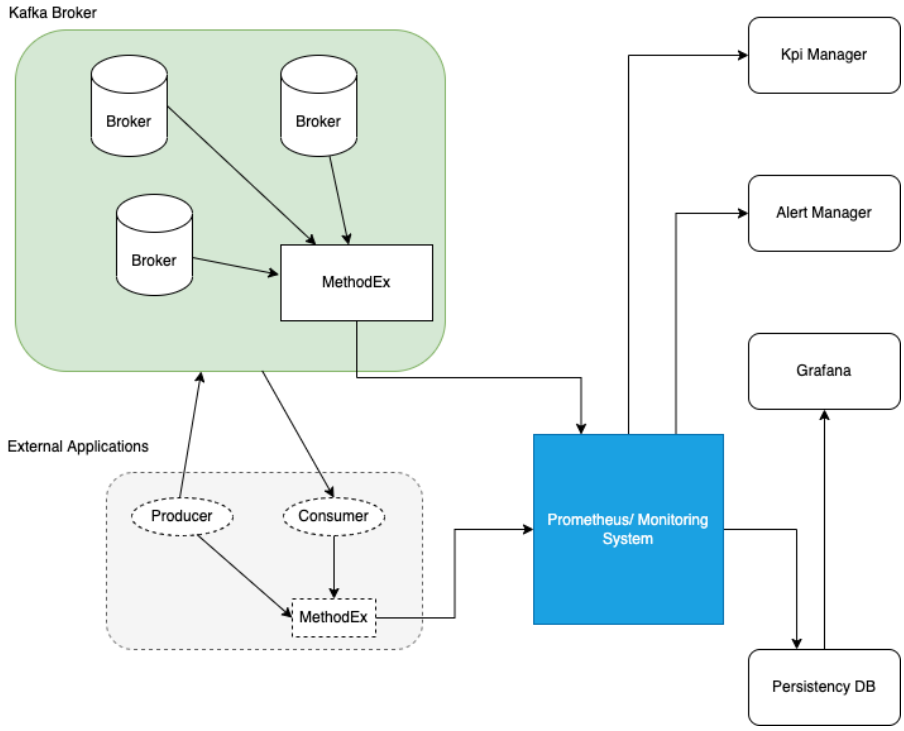

Figure 4-18: Telemetry Component.

Kafka Broker and External applications:

The software architecture in Figure 4-18 is based mainly on Kafka broker, used as a messaging system or more precisely as event bus to process and distribute streams of data in real time, allowing systems and applications to communicate with each other and process data as it is generated.

The broker is a server that runs a Kafka instance and is responsible for managing the storage and distribution of messages within the Kafka cluster. A Kafka cluster typically consists of multiple brokers, and each broker can handle thousands of incoming messages and multiple concurrent client connections. Here, Kafka is used as a telemetry system by setting up a Kafka cluster and configuring applications and devices to send data to the cluster as messages using the MethodEx interfaces. The brokers in the cluster can then distribute the messages to any interested consumers in real time, allowing them to process and analyze the data as it is generated.

With this aim and referring to Figure 4-18 the external applications could be any component of the Smart5grid platform that acts as an external entity, being able to represent a producer or a consumer of information and using the MethodEx interface to address the data pipeline towards Prometheus with monitoring purposes.

In more details, when an application that act as a publisher connects to a Kafka cluster, it queries which partitions exist for the topic and which nodes are responsible for each partition. Publishers assign messages to each partition using a hashing algorithm and deliver them to the broker responsible for that partition.

The way in which Kafka and Prometheus can work together is by setting up a Kafka cluster and configuring applications and devices to send data to the cluster as messages. The brokers in the cluster can then distribute the messages to any interested consumers, including Prometheus. Prometheus can consume the data from Kafka as a data source, allowing it to process and store the data in its time-series database. It will depend on the kind of persistence needed that will drive the choice of the database.

The question on how to monitor Kafka Cluster, producers, and consumers is addressed via Confluent and precisely with the Confluent Control Center as established in [10].

MethodEX:

Although we can view Kafka and Zookeeper metrics in jConsole, in a real-world scenario we want to automatically collect these metrics and show them in an informative dashboard. To collect Kafka metrics for Prometheus, use the MethodEX export. This exporter can be tailored to produce metric JMX that is a common technology in Java world for exporting statistics of running application as well as control it.

At the same time the MethodEx interface can wrap libraries like MongoDB exporter or PostgreSQL exporter depending on what kind of data could be of interest for the Prometheus server to get. Generally intended then MethodEx is an API in that allows us to export different data.

Prometheus:

The Prometheus server gets the different metrics from the different node exporters. The MethodEX interface is a general node exporter used by the application to push data to Prometheus server.

Grafana:

To show complex data in a dashboard we need to install Grafana and import Prometheus data into that view. After importing data source, one can create the needed dashboard, which can be done using customizable configuration. Finally, this dashboard can be added to Grafana as a plugin, so that other developers can use it.

To provide a comprehensive view of the overall health of the Smart5Grid Kafka cluster, we included the following features:

- how many brokers are alive in the cluster,
- metrics for partitions,
- throughput,
- requests,
- response queues size,
- Zookeeper connections,
- Producer and consumer metrics.

Alert Manager and KPI Manager Optional:

These functionalities can be expressed sending alert like message or email if something goes wrong and will be implemented in the pilot use case if needed.

Persistency DB Optional:

The possibility to use a time-series database (TSDB) will be investigated in the pilot site fine tune so this part will be implemented if needed. In the NAC, the main part of it is an Apache Kafka broker that acts as the main BUS for the Network Application controller. The different components that compose the Network Application controller circulate different messages through it to assure the Network Application management via LCM.

For this software the docker landoop/fast-data-dev[3] image was used. The package includes the following services:

- Kafka Broker: 9092
- Schema Registry: 8081
- Kafka REST Proxy: 8082
- Kafka Connect Distributed: 8083
- ZooKeeper: 2181
- Web Server: 3030

The overall component is in charge to check if the metrics indicated by the Network Application descriptor are inside a given threshold and must trigger an alarm via proper topic if these thresholds are violated.

This is mainly operated by a producer of the Kafka topic_metric_alerts, which emits JSON format message, as shown in Figure 4-19.

[3] https://hub.docker.com/r/landoop/fast-data-dev/dockerfile

```
{
   "name": "netapp_example",
   "event": "deployed",
   "SLOs":
   [
      {
         "name": "percentage of CPU consumption",
         "expresion": "CPU[5m]",
         "metric": "CPU_usage",
         "threshold": "10",
         "threshold-type": "GT",
         "action":
         {
            "target-ref":
            {
               "target-ns-ref": "service1-ns",
               "target-vnf-ref": "kube-prometheus_vnfd"
            },
            "action-step": "trigger-scale-up"
         },
         "granularity": "3",
         "cycles": "4"
      }
   ]
}
```


Figure 4-19: JSON for metric alerts

A producer of the Kafka topic_lifecycle_events topic emits JSON format message as established in Figure 4-20.


```
{
   "name": "netapp_example",
   "event": "deployed",
   "monitoringURL":
   {
      "ns-ref": "service1-ns",
      "sap-ref": "ns1_sap_monitoring",
      "url": "192.168.10.10:9090/metrics"
   }
}
```


Figure 4-20: JSON for lifecycle events

A producer of the Kafka topic_policy_breaks_event topic emits JSON format message, see Figure 4-21.


```
{
   "name": "netapp_example",
   "SLOs":
   [
      {
         "name": "percentage of CPU consumption",
         "target_ns": "service1-ns",
         "target_vnf": "kube-prometheus_vnfd",
         "action-step": "scale-up",
         "cycles": "4"
      }
   ],
   "time_violation": "2022-08-05T06:11:06.514Z"
}
```


Figure 4-21: JSON for identification of break event.

4.2.2. Interfaces with Smart5Grid Architecture

In UC2, the NAC component has the role of the communication service provider in the Smart5Grid platform. It means it provides interfaces to enable the commissioning and decommissioning of Network Applications over the 5G infrastructure layer. Table 4 summarizes the interfaces supported by NAC which in the next subsection is described their implementation and integration with the rest of Smart5Grid components. The following interfaces are active to support the integration of the Network Application controller and Smart5Grid Architecture.

Table 4: Interfaces with Smart5Grid platform.

Interfaces	Method	Support
OSR_NAC	Download VNF Image data	To be completed in WP4
OSR_NAC	Download resource from registry	To be completed in WP4
V&V_M&O-Adapter	Provide a Network Application validation test	To be completed in WP4
VIM_V&V	Launch VIM operations	Neutroon
NFVI_V&V	Launch NFVI operations	Neutroon
NAC_FE	Network Application Onboarding	OK
NAC_FE	Network Application Deployment	NBI
NFVO_SM	(Opt.) Network Application Deployment (NFVO)	Neutroon
VIM_SM	Register Computing Resources (VIM)	Neutroon
RAN-Controller_SM	(Opt.) Register Radio Resources (RAN-Controller)	Neutroon
5GCN-Controller_SM	(Opt.) Provide 5GC Resources (5GCN-Controller)	Neutroon
NFVO_ OSS/BSS	NSD Management	Neutroon
NFVO_ OSS/BSS	NS Lifecycle Management	Neutroon
NFVO_ OSS/BSS	VNF Package Management	Neutroon
NAC_SM	Network Application Creation and deployment	SM Adapter

NAC_SM Interface:

This interface is used by NAC to communicate with Slice Manager. Since the NAC used in UC2 is telco-oriented, the NAC only implements this interface due to it acts as a Communication Service Management Function (CSMF) according to the Smart5Grid platform. Specifically, the Slice Manager implements northbound and southbound interfaces to deploy applications, which are detailed below:

1. Northbound Interface (SM): As mentioned in the previous section, only the methods related to the creation, instantiation, and deletion of Network Applications will be described. The main methods used are detailed as follows:
   - GET Applications: Lists all the applications created in Slice Manager. The information of the applications is listed together with the information of the network slices created previously by Slice Manager due to the applications belonging to a specific network slice.
   - CREATE Applications: Creates applications composed of several VNFs inside of the Slice Manager. The applications contain information regarding the VNF type if it is VM-based or Container-based application. This process is stored in the database of the Slice Manager.

- DELETE Applications: This method deletes the applications stored in the Slice Manager.
- GET Instances: This method lists all the applications instantiated in a specific network slice.
- INSTANTIATE Applications: Instantiates applications linked to a specific network slice. The method includes information related to the application name (created previously) and slice ID. The Slice ID field is obtained when the network slice is created.
- DELETE Instances: The method deletes the application instances along with the network slice. The method includes information of the network slice name.

2. Southbound Interface (SM): This interface is used by Slice Manager to communicate with the rest of the components composing the Smart5Grid platform. Basically, Slice Manager interacts with RAN Controller, Core Network Controller and VIM controller to create slices. On the other hand, Slice Manager deploys applications by making use of the NFVO which communicates with compute nodes through the VIMs (Virtual Infrastructure Managers) to instantiate applications as VMs or as Containers. This means that Slice Manager implements several clients to perform actions over the management and orchestration components. Figure 4-22 illustrates the basic workflow to instantiate Network Applications by using Network Application Controller and Slice Manager.

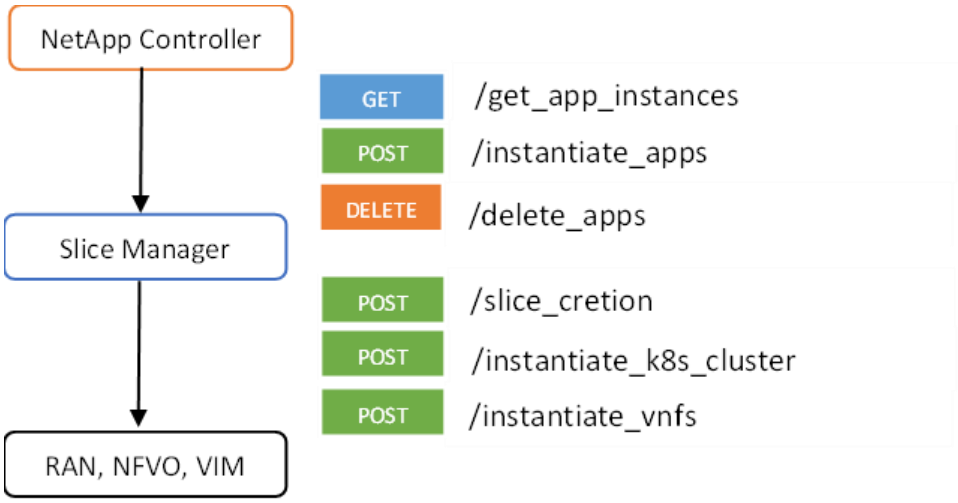

Figure 4-22: Slice Manager northbound and southbound interfaces description.

4.2.3. Integration Tests

Table 5: Integration tests UC2

Name		Locations	UC2
Component under test		Responsible	ATOS
Test environment		V&V Framework and NAC	
Feature under test		Onboarding Network Applications packages	
Preparation		1. UC2 NAC instance deployed with the NAC_FE interface ready for Network Applications onboarding waiting for	

	onboarding by the V&V defined. 2. V&V framework deployed
Dependencies	Network Application Controller NBI
Steps	1. The Network Application package has to have been previously verified by the V&V Framework. 2. The V&V Framework makes a request to upload the Network Application package. 3. The NAC returns an OK message when the package has been onboarded.
Pass Criteria	All steps executed without errors.
Result	Successful

Name	NAC_FE	Locations	
Component under test	NAC_FE	Responsible	I2CAT
Test environment		NAC, Neutroon and edge nodes for UC2	
Feature under test		Deploy and undeploy Network Application's services	
Preparation		1. Connectivity enabled to end users and FE components to execute the deployment process 2. A valid Network Application descriptor already onboarded in the Local Registry subcomponent.	
Dependencies		Local Registry Component	
Steps		1. A Network Application already onboarded on NAC 2. The user fills the POST method to deploy the Network Application 3. The NAC's NBI receives the requested method and sends to LCM component. 4. LCM gathers the name of the Network Application descriptor and requests the descriptor to the Local Registry component. 5. Local Registry Component sends the Descriptor to be read and translated by LCM. 6. LCM deploys the Network Application by using the NAC_SM interface. 7. The NAC receives the status provided by Neutroon 8. The Network Applications are deleted through POST methods in the NAC's NBI. 9. The NAC send the request to Neutroon	

	platform by using the NAC_SM interface 10. The Neutroon platform deletes the VNFs via the NFVO 11. The NAC receives the confirmation from Neutroon platform
Pass Criteria	All steps executed without errors
Result	Successful

Name	NAC_SM	Locations	UC2
Component under test	NAC_SM	Responsible	I2CAT
Test environment		NAC, Neutroon and edge nodes for UC2	
Feature under test		Deploy and undeploy Network Application's services	
Preparation		1. Connectivity among NAC (i2CAT lab), Neutroon platform (public Cloud) and infrastructure nodes where the services are onboarded. 2. A valid Network Application descriptor, NSDs and VNFDs are already onboarded.	
Dependencies		Neutroon platform	
Steps		1. A Network Application already onboarded on NAC and NSDs/VNFDs already onboarded in OSM 2. The NAC sends the deployment request to the Neutroon platform 3. The Neutroon platform receives the name of the VNFs to be deployed by the NFVO 4. The NFVO deploys the Network Applications over the edge nodes previously registered by Neutroon 5. The Network Applications are linked to the 5G Core Network previously deployed by Neutroon to enable the E2E communication 6. The NFVO reports the status of the VNFs to Neutroon platform 7. The NAC receives the status provided by Neutroon 8. The Network Applications are deleted from NAC's NBI 9. The NAC send the request to Neutroon platform 10. The Neutroon platform deletes the VNFs via the NFVO 11. The NAC receives the confirmation from the Neutroon platform	

Pass Criteria	All steps executed without errors
Result	Successful

Name	Telemetry _ Protocol	Locations	UC2
Component under test	Telemetry and external application	Responsible	Engineering
Test environment		Engineering	
Feature under test		Production and consumption of a metric	
Preparation		1. Preparation of a service API that provide random data mimic the metric	
Dependencies		N/A	
Steps		1. Creation of topic on Kafka 2. Push data on the topic 3. Create a consumer 4. Retrieve data	
Pass Criteria		All steps are carried out without errors	
Result		Successful	

Name	Telemetry _ AL	Locations	Local
Component under test	Telemetry and external Monitoring instance to raise alert	Responsible	Engineering
Test environment		Engineering	
Feature under test		Metric visualization	
Preparation		Topic set up	
Dependencies		N/A	
Steps		1. Expose Metrics Using methoDEX 2. Store Kafka Cluster Metrics in Prometheus 3. Visualize 4. If a certain pre-defined value is exceeded, send an email alert	
Pass Criteria		All steps are carried out without errors	
Result		Successful	

Name	Grafana_server_dashboard	Locations	Local
Component under test	Grafana and Prometheus	Responsible	Engineering
Test environment		Engineering	
Feature under test			
Preparation		Preparation of an API	
Dependencies		N/A	
Steps		1. Stand-up a local Grafana server as a Docker container.	

	2. Configure Prometheus as a data source in Grafana 3. Import sample dashboards provided by Grafana and/or community 4. Visualization data on dashboard
Pass Criteria	All steps are carried out without errors
Result	Successful

5. Conclusion and Next Steps

In this deliverable, we provided an update of D3.1 regarding the work carried out in the context of WP3.

In Section 2, we provided a reminder of the components of the Smart5Grid infrastructure from the previous deliverables.

In Section 3, we provide an update regarding all four use cases. Here, we provided a high- and low-level infrastructure overview of all uses case as well technical details that were not known at the time of writing D3.1. Where relevant, each use case highlighted the challenges faced because of the security concerns that need to be taken into consideration when working in critical infrastructure of national importance. Due to the nature of Smart5Grid, the project and its use cases faced this challenge with both stakeholders involved: National Telco Operators and energy Distribution/Transmission System Operators. Although these challenges have slowed down the use case, we believe that such challenges, alongside with the solutions proposed, are of great importance. Thus, similar projects should take note.

In Section 4, we describe the two Network Application Controllers and MEC Orchestrators deployed in the four use cases. We have reported how a commercial orchestrator (NearbyOne) was adapted and how, instead, a custom solution has been built from the partners. The choice of using two solutions, one cloud oriented and the other telco oriented, was beneficial and allowed the project to explore solutions to two worlds that over the last years have been converging and that with 5G started to overlap. Furthermore, we have provided the integration tests done between the different components of the Smart5Grid architecture. These tests have also involved components that were not in the scope of this deliverable, i.e., the OSR and the V&V Framework. In particular, the OSR will be described in D3.3 (parallel to D3.2 and also due in M24) and the V&V Framework in D4.2 (due in M30).

For what concerns WP3, as the project reached its deadline the work has concluded. However, since some components are being developed in other WP that are still running, some minor adjustment work might still be required so if further work is required, it will naturally fit within the activities of the initial pilots site setup. The work done during WP3 and resumed in this D3.2 will constitute an input for tasks of WP4, WP5 and WP6. WP4 will conclude the Network Application deployment and their testing using the V&V Framework. WP5 and WP6 will instead conclude the installation of the Smart5Grid field platform, Network Application deployment, and carry out the actual field pilots for UC1&2 (WP5) and UC3&4 (WP6).

6. References

[1] Smart5Grid deliverable D3.1 "Interim Report for the development of the 5G network facilities". https://smart5grid.eu/wp-content/uploads/2022/11/Smart5Grid_WP3__D3.1_PU_Interim-Report-for-the-development-of-the-5G-network-facilities_V1.0.pdf

[2] Smart5Grid deliverable D2.1 "Elaboration of Use Cases and System Requirements". https://smart5grid.eu/wp-content/uploads/2021/07/Smart5Grid_D2.1_Elaboration-of-UCs-and-System-Requirements-Analysis_V1.0.pdf

[3] Smart5Grid deliverable D2.2 "Overall architecture, design, technical specifications, and technology enablers". https://smart5grid.eu/wp-content/uploads/2021/11/Smart5Grid_WP2_D2.2_V1.0.pdf

[4] STERPMU-R1 RACK MOUNTED PMU http://www.wamster.net/downloads/STER-PMU-R1-Quick-Start-Manual.1.1.2700.pdf

[5] NR7101 5G NR Outdoor Router https://download.zyxel.com/NR7101/datasheet/NR7101_5.pdf

[6] ZTE 5G Outdoor CPE MC7010 https://ztedevices.com/en-eu/mc70101/

[7] Information Model of the Network Application Descriptor https://bitbucket.org/AntonelloCorsi/netapp-im/src/master/

[8] NearbyOne's Smart5Gird NAC_FE REST API definition in Swagger 2.0. https://bitbucket.org/AntonelloCorsi/NearbyOne/src/master/

[9] Prometheus's functional query language – PromQL https://prometheus.io/

[10] Confluent Control Center https://docs.confluent.io/platform/current/platform.html