5G-PPP Software Network Working Group

Network Applications:
Opening up 5G and Beyond networks

Version 2.0, July 2023

Date:	2023	Status:	Public Release

DOI: XXX
URL: XXX

Abstract

It is expected that the communication fabric and the way network services are consumed will evolve towards 6G, building on and extending capabilities of 5G and Beyond networks. Service APIs, Operation APIs, Network APIs are different aspects of the network exposure, which provides the communication service providers a way to monetize the network capabilities. Allowing the developer community to use network capabilities via APIs is an emerging area for network monetization. Thus, it is important that network exposure caters for the needs of developers serving different markets, e.g., different vertical industry segments.

The concept of "Network Applications" is introduced following this idea. It is defined as a set of services that provides certain functionalities to verticals and their associated use cases.

The Network Applications is more than the introduction of new vertical applications that have interaction capabilities. It refers to the need for a separate middleware layer to simplify the implementation and deployment of vertical systems on a large scale. Specifically, third parties or network operators can contribute to Network Applications, depending on the level of interaction and trust.

In practice, a Network Application uses the exposed APIs from the network and can either be integrated with (part of) a vertical application or expose its APIs (e.g., service APIs) for further consumption by vertical applications.

This paper builds on the findings of the white paper released in 2022. It targets to go into details about the implementations of the two major Network Applications class: "aaS" and hybrid models. It introduces the Network Applications marketplace and put the light on technological solution like CAMARA project, as part of the standard landscape.

Table of Contents

List of Acronyms and Abbreviations

3GPP	3rd Generation Partnership Project
5G-PPP	5G Infrastructure Public Private Partnership
5GC	5G Core
API	Application Programming Interface
AR	Augmented Reality
AOEP	Automotive Open Experimental Platform
CAPIF	Common API Framework
CI/CD	Continuous Integration/Continuous Delivery
CNF	Container Network Function
CSP	Cloud Service Provider
EC	European Commission
EU	European Union
eNB	Evolved Node B
ETSI	European Telecommunications Standards Institute
ICT	Information and Communication Technology
IoT	Internet of Thing
IOPS	Isolated Operation for Public Safety
K8s	Kubernetes
KPI	Key Performance Indicator
LCM	Life Cycle Management
MANO	Management and Orchestration
MEC	Multi-Access Edge Computing
NEST	Network Slice Type
NFV	Network Function Virtualization
NFVO	Network Function Virtualization Orchestrator
NRF	Network Repository Function
NSD	Network Service Descriptor
NSMF	Network Slice Management Function
NSSMF	Network Slice Subnet Management Function
NWDAF	Network Data Analytics Function

PaaS	Platform as a Service
PPDR	Public Protection and Disaster Relief
OSS	Operations Support System
KPI	Key Performance Indicator
RAN	Radio Access Network
SBA	Service Based Architecture
SEAL	Service Enabler Architecture Layer
SFC	Service Function Chaining
SMO	Service Management and Orchestration
SLA	Service Level Agreement
SLO	Service Level Objective
SOA	Service Oriented Architecture
VAL	Vertical Application Layer
VNF	Virtual Network Function
VNFFG	Virtual Network Function Forwarding Graph
VNFD	Virtual Network Function Descriptor
VM	Virtual Machine
VR	Virtual Reality
WG	Work Group
WP	White Paper

1 Introduction

“Network Applications” is defined as a collection of services that provide specific functionalities to verticals and their associated use cases. In practice, this software consumes the network’s exposed Application Programming Interfaces (APIs) and can either be integrated with (a part of) vertical application or expose its APIs (e.g., service APIs) for further consumption by vertical applications. In short, Network Applications can be thought of as a separate middleware layer that helps to simplify vertical system deployments. In practice, three deployment models are identified in the first version: (1) as-a-Service, (2) Hybrid, and (3) Coupled/Delegated to be chosen based on the vertical use-case.

In essence, Network Applications interacts with a mobile network’s control plane by consuming exposed APIs, such as northbound APIs of the Core and Radio Access Network Intelligent Controller (RIC), and edge computing APIs, in a standardized and trusted manner to compose services for vertical industries. Furthermore, by gathering key network data for the managed slice/service, (re)training in Machine Learning (ML) pipelines can be applied to revisit decision enforcement limitations for Artificial Intelligence (AI) algorithms used by verticals. To evaluate this potential, an experimental platform for third-party experimenters and targeted verticals to test Network Applications in an integrated and fully featured environment is required.

Furthermore, the new architectural framework and Network Applications are gaining traction among vertical industries, industry forums (e.g., 5G-ACIA, 5GAA, 5G America, 5G-IA), and standardization bodies (e.g., 3GPP). Recently, the CAMARA project is shaped by the GSMA, Linux Foundation, and the wider ecosystem to align API requirements and publish API definitions and APIs to realize on-demand, secure, and controlled network exposures.

Network Applications is seen as a full-potential enabler for future vertical industries beyond current deployment. Therefore, it must be considered along with other 6G enabling technologies in the next-generation network architecture. This paper focuses on the different technical aspects of Network Application, new business models for all stakeholders, experimental facilities to support Network Application, and new Network Intelligence (NI) solutions that can be enabled by using Network Application.

This white paper is the second version of the published version in 2022. It targets to go into details about the implementations of the two major Network Applications class: “aaS” and hybrid models. It is organized as follows:

- Section 2 and Section 3 reminds the Network Applications requirements and classification based on the findings of the first release of the white paper made in 2022 [1].
- Section 4 introduces the tooling box to build a Network Application.
- Section 5 presents some specific “aaS” class implementations from ICT-41 projects.
- Section 6 presents some specific “hybrid” class implementations from ICT-41 projects.
- Section 7 introduces the design rules and best practices for the security.
- Section 8 highlights the business models toward monetization and the marketplace.
- Section 9 focuses on some key standards like CAPIF and CAMARA.
- Finally, conclusions are summarized in Section 10

2 Network Application requirements

The vertical space targeted by 5G/B5G system is very large. It considers diverse use-cases belonging to the following domains: Smart Cities & Utilities, Transportation, Automotive, Media & Entertainment, Agriculture & Agri-food, Smart (Air)ports, Energy and E-health & Wellness as it is pointed out in [2]. One can ask:

- Does each vertical want to create its own solution for each service?
- Does each vertical want to negotiate with each communication service provider (CSP) on how best to utilize resources?
- Does each vertical want to convince each CSP to use its defined API interface?
- Does each application developer want to adapt to each required CSP API interface?
- Does each private 5G deployment want to negotiate and adapt to each infrastructure provider
- Does each infrastructure provider want to negotiate and adapt to each private 5G/B5G system deployment?

We need an application layer *in the middle* offering common or vertical specific application function or services (for simplicity we call this ***Middleware Layer)*** to simplify the implementation and deployment of vertical system at large scale and this what we call Network Application.

Although there does not yet exist a standard definition of what a Network Application is, in the 5G-PPP Software Working Group, we identified some key characteristics that shall aid to the definition of a Network Application in the context of the 5G/B5G System. Specifically, a Network Application is defined as set of services that provide certain functionalities to the verticals and their associated use cases**.**

The following are some identified characteristics of a Network Application. Specifically, a Network Application:

- Should deliver services to 5G/B5G vertical sectors;
- May expose APIs to be consumed by other service consumers. The exposed APIs should be delivered in an Open API model and may follow the 3GPP recommended APIs for applications (i.e. 3GPP CAPIF, Service Enabler Architecture Layer for Verticals – SEAL [10]);
- A Network Application may be part of one or more vertical application services;
- One or more services of the Network Application may be attached to one or more 5G User Plane Functions (UPF);
- May be part of one or more 5G slices. The slices may be shared or not;
- Part of a Network Application may reside at the (UE) side. The part of the UE side may interact with a Network Application service that resides within the domain network. The UE part may follow the definition of the Vertical Application Layer (VAL) client of 3GPP;
- May interact with the 5G/B5G System by consuming 5G/B5G System's APIs (i.e. the NEF), if the 5G system allows. When interacting with the 5G Systems, it must support relevant 3GPP standards. Such interactions may include location services, Quality of Services (QoS) management, Assured Forwarding (AF) traffic;
- May support service continuity by minimizing service interruption when transferring application context;
- May have placement requirements (e.g. edge, region, core, etc.). Additionally, a network latency KPI must be specified by the Network Application when requesting a slice with specific characteristics by the 5G/B5G System;
- May consume monitoring and telemetry data from the 5G/B5G System. Such data from the 5G/B5G System should be consumed by functions like the Network Data Analytics Function (NWDAF);
- May interact with the service orchestrator or resources Orchestrator of the domain if this is not restricted;
- Should follow relevant 3GPP security definitions and recommendations.

Software networks provide high flexibility through implementation of virtual network functions (VNFs). This requires open platforms that provide access to networks resources which can then be used to develop Network Applications supporting requirements and developments from specific vertical sectors.

3 Network Applications Classification

The Software Network Working Group published a first version of a white paper introducing the concept of the Network Applications [1]. The Network Application ecosystem is more than the introduction of new vertical applications that have interaction capabilities. It refers to the need for a separate middleware layer to simplify the implementation and deployment of vertical systems on a large scale. Specifically, third parties or network operators can contribute to Network Applications, depending on the level of interaction and trust.

Different implementations have been conducted by the different projects considering different API types and different level of trust between the verticals and the owner of 5G platforms.

Considering the level of interaction and trust, the Network Applications could be classified following their architectural position:

- Network Applications as part of 5G/B5G System
- Network Applications adjacent to the 5G/B5G System,
  - still in the CSP domain, typically as part of a Network Operator network slice
  - in interconnected (CSP / Service Provider) Domain, typically as part of a tenant / application slice

Following the level of Network Applications integration, it results three categories, as depicted in Figure 1:

- **aaS Model**: it is the model where the vertical application consumes the Network Applications as a service. The API is offered by a (Mobile / Communication) Service Provider (CSP) or a Vertical (Sector) specific Digital SP (DSP). The vertical application deployed in the vertical service provider domain. It connects with the 3GPP network systems in one or more PLMN operator domain.
- **Hybrid**: it is the model where the vertical instantiates a part of its Vertical Application in the operator domain like the EDGE. The other part remains in the vertical domain. A similar approach has been followed in TS 23.286 related to the deployment of V2X server.
- **Coupled/delegated**: it is the model where the vertical delegates its application (in short app) to the operator. The Network Applications will be composed and managed by the CSP. This approach is the one followed in the platforms like 5G-EVE.

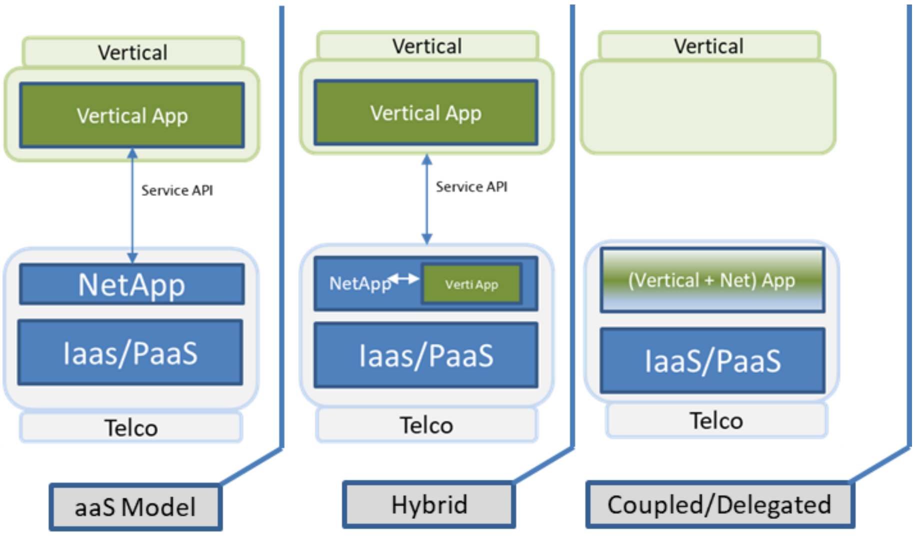


Figure 1: Network Applications classification

4 Network Applications technologies

As per [1], a Network Application is a piece of software working as a middleware that can interact with a mobile network by consuming the available APIs and that provides features powered by network information to vertical applications exposing service APIs. Network Applications have been categorized based on their integration mode within the mobile network domain as documented in chapter 6 [1].

Any implementation of a software produces one or more by-products, referred to as artifacts. An artifact provides the required information to execute part or the totality of an application. Artifacts also enable an ecosystem where applications are built as the combination of many artifacts and thus, it is possible to reuse, extend and distribute them. Chapter 8 extends this concept by introducing the role of the Artifact Registry as a unified and consistent storage and distribution system exposed to interested stakeholder with a Marketplace on top of it to truly leverage the potential of the new business opportunities brought by the Network Applications. In fact, several of the projects that have been studied within the Software Network WG, have identified the need for a place where Vertical Service Providers, Communication Service Providers and End Users can engage into business transactions to offer, host, and consume Network Applications.

The Network Applications, as part of the 5G and Beyond ecosystem, heavily rely on of disruptive paradigms such as cloudification and edge computation. In practice, high-end technologies, such as Kubernetes, build the baseline of Network Applications which is then orchestrated using novel techniques.

The exploratory research from [1] indicates that the implementation of a Network Application involves the release of several artifacts structured in a specific format and packaged depending on the underlaying technology which are then linked together using unique identifiers.

4.1 Technology layers

Figure 2 depicts a high-level view of components and technology selection of a Network Application. It is worth highlighting that the application logic side is too extensive in terms of the complete list of technologies that could be used, and more importantly, it lacks interest to consider it as part of the components of the Network Application as it is embedded into the virtualization layer (denoted with dotted lines in Figure 2) and distributed altogether. The remaining layers are distributed as individual artifacts and references are required by the outer layer and offered by the inner layer. Additionally, each layer in Figure 2 is described as a chain of services, meaning that each layer is built as the combination of several implementation of the inner layer in a *1:N* type of relationship.

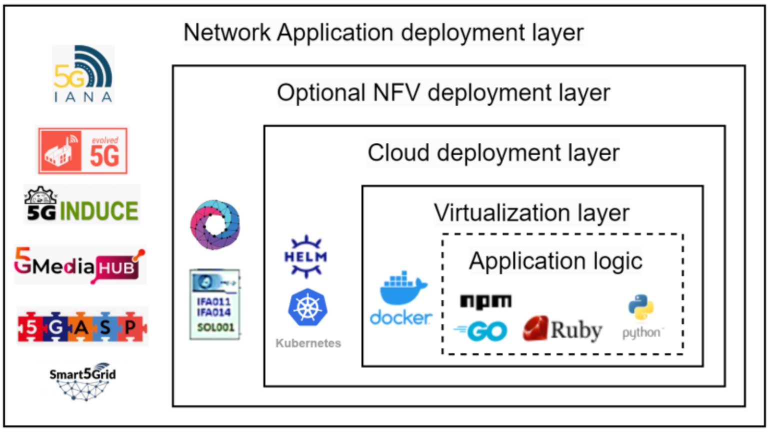


Figure 2: Technology layers for the implementation of Network Applications

All inner layers from Figure 2 represent mature technologies in the 5G industry and have a large community in charge of their evolution and standardization. However, as expected from a research program, all projects evaluated by this WG have their own individual communities lead by their objectives which in turn, although partly based on common standards, have created a different implementation of what a Network Application artifact looks like and that is further elaborated in section 4.2.

4.1.1 Network Function Virtualization

NFV is an approach to network architectures that focuses on reducing the degree of dependency that traditional network elements have with the physical equipment. Decoupling the functionality from the underlying hardware permits a more flexible and standardized evolution of network elements and consequently reducing the overall costs and opening potential paths for a more autonomous network management and players [6].

NFV is a specification produced by the ETSI ISG NFV group in 2012 [6] which is key to fully leverage the potential of 5G. The NFV specification distinguishes between three different kinds of implementations:

- Virtual Network Functions (VNF) are applications deployed using virtual machines over a hypervisor belonging to the cloud infrastructure.
- Physical Network Functions (PNF) are applications delivered embedded in a custom hardware element.
- Container Network Functions (CNF) also referred as Cloud-native Network Functions, are applications distributed as a container running inside a virtualized platform.

In the context of Network Applications, only CNFs have been identified in the projects evaluated. Furthermore, the NFV layer is mostly required by those implementations following the hybrid or delegated model.

4.1.2 Cloud

The cloud is a software-defined, highly resilient virtualized infrastructure that continues to grow in the 5G ecosystem to deploy workloads and leverage computing, storage and network resources on-demand in a flexible way.

Kubernetes (K8s) [52] is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem in which services, support, and tools are widely available. K8s is used for automating deployment, scaling and management of the containers. It also solves one of the biggest problems with container-based deployment which is how all the containers should be coordinated and scheduled.

Helm is a tool for managing packages of pre-configured K8s resources [53] that greatly simplifies packaging and deployment of containerized applications.

4.1.3 Virtualization

Containers, usually referred to as Docker [52], are similar to VMs, but they have a lower degree of isolation properties since they share the Operating System (OS) among every container running in the same host. But they do have their own file system, share of CPU, memory, process space, etc.

Containers maintain the principle of decoupling the software functionality from the hardware platform by packaging the applications and their dependencies into isolated lightweight boxes. Built using a microservices architecture, they are dynamic, flexible, and easily scaled.

4.2 Modelling languages

Network control and management has evolved, and nowadays Multiple Standard Organizations (SDO) have contributed with data modelling languages and the corresponding data models to describe a service and/or device capabilities, attributes, operations, and notifications to be performed or received from a device or service [4].

- YAML (YAML Ain't Markup Language) is a data serialization language mostly used in configuration files and playbooks definitions. YAML’s minimal syntax makes it easy to read (except when using complex data structures) and so it’s characterised ‘human readable’.
- XML (Extensible Markup Language) is a markup language that defines a set of rules for encoding documents and representing data structures, primarily designed to store and transport data. It is both human and machine readable, but is seen far more in API transactions between systems.
- JSON (JavaScript Object Notation) is a data interchange format that uses human readable text to transmit data objects consisting of attribute and value pairs. It was derived from JavaScript but is language independent. JSON is typically preferred over XML due to JSON`s lightweight nature (i.e there is no requirement for closing tags like XML).
- The Internet Engineering Task Force (IETF) has proposed Yet Another Next Generation (YANG) data model language

Later, the introduction of Protocol Buffers has signified a new step in data model definition. The proposed data modelling languages have an associated transport protocol, which provides primitives to view and manipulate the data, providing a suitable encoding as defined by the data-model.

Ideally, data models should be protocol independent. Each proposed transport protocol shall provide an architecture for remote configuration and control based on client / server. It should support multiple clients, provide access lists, include transactional semantics, and deploy roll-back functionalities in case of error. In practice, current data modelling languages and transport protocols are tightly interrelated.

4.3 API protocols

Figure 3 depicts the different APIs in the architecture from the application developer/consumer to the 5G network composed of RAN, Transport and CORE.

Most standards organizations today publish Open (REST) APIs (3GPP, TMForum, ETSI, ZSM, …). For example, 3GPP has defined its Service-Based Architecture (SBA), standardized REST APIs, a Network Exposure Function (NEF) and a Common API Framework (CAPIF) [13] as a structural part of the Core Network architecture.

CAMARA APIs is a new industry effort on defining common, global and simplified APIs focused on usage by the widest possible developer community to access CSPs network capabilities.

In the sequel, we list the most used API type in addition to the REST APIs.

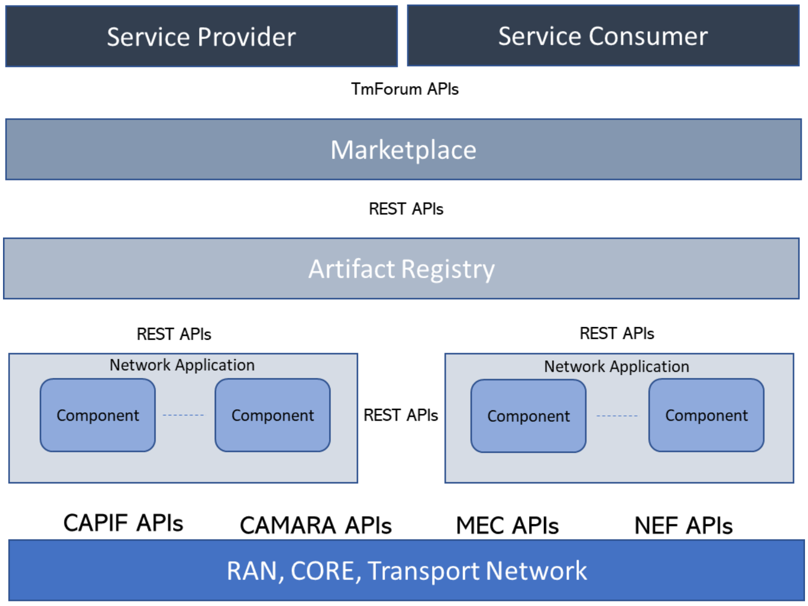


Figure 3: Architecture and APIs

4.3.1 OpenAPI and HTTP

Finally, OpenAPI is a data modelling language that is provided through the HTTP protocol. It allows the definition of JSON/XML schemas for data exchange, the multiple defined URLs, and finally, the possible defined operations and the involved data in JSON/XML.

4.3.1.1 What is HTTP REST?

- Representational State Transfer (REST) is a software architectural style for Application Programming Interfaces (APIs) that consists of guidelines and best practices for creating scalable web services. REST uses simple HTTP to make calls between machines.
- This happens via a request/response mechanism between the server and the client. For example, a client, let’s say an Android application, makes a request for the most recent posts from the website. The server knows how to interpret this request, through REST, and satisfies the response by providing the most recent posts in a format understood by the client.
- REST requests interact with the resources in your application (e.g. a Post or Page). These interactions are typically Reading, Creating, Updating, or Deleting. Combined with HTTP, REST requests are formed using four verbs:
  - POST: Create a resource
  - GET: Retrieve a resource
  - PUT: Update a resource
  - DELETE: Delete a resource
  - The data retrieved is supplied in a machine-readable format, often JSON in modern web applications.

- REST was proposed by Roy Fielding in his 2000 dissertation Architectural Styles and the Design of Network- based Software Architectures

4.3.1.2 What makes an API RESTful?

- An API must have the following architectural features to be considered RESTful:
- Client-server: the client is separated from the server. This means that clients are not concerned with data storage and servers are not concerned with display. This ensures that data is portable and can be reused in multiple clients, and servers are simpler and more scalable.
    - Cacheable: clients can, and should, cache responses to improve performance, and avoid the server with every request.
    - Stateless: the necessary state to handle the request is contained in the request itself, whether as part of the query parameters, URL, body, or headers.
    - Uniform interface: information transferred via REST comes in a standardised form, creating a simplified, decoupled architecture.
    - Layered System: the architecture is composed of hierarchical layers. Each component cannot “see” beyond its layer: a client cannot tell if it’s connected to the server or to an intermediary.
- A separate, but closely related concept is hypermedia. Hypermedia allows a client to more fully discover a REST API without needing to know anything about the structure of the API. It’s similar to hyperlinks on the human-readable web (which enable discovering new sites and content). The server provides the information the client needs to interact with it. This means that the client can interact with the server in complex ways without knowing anything beforehand about it.

4.3.1.3 What is an open API?

- Open APIs are publicly available APIs that give developers access to proprietary software information that they can make use of in their own software and applications. REST is the ideal architecture for creating an Open API for the web because, by using HTTP, it is built on the principles of the open web. To leverage an open REST API a developer just needs to make a HTTP request.
- By making data available for developers to use in their own applications, open APIs are transforming the internet. Developers can access data across services, creating applications that aggregate information from different providers. The impact of APIs cannot be overestimated; they are transforming the way businesses and services are run.
- In general, then Open API allows to describe, develop, test, and document APIs conforming to the REST architecture, so it allows to create RESTful APIs.

4.3.2 Protocol buffers and gRPC

Protocol Buffers (protobuf) are a language-neutral, platform-neutral extensible mechanism for serializing structured data. Its encoding in byte-oriented messages increases the efficiency compared to XML/JSON encodings.

Following the proposal of protobuf, novel protocols such as gRPC and gNMI have been proposed. Google Remote Procedure Calls (gRPC) is based on HTTP/2 and considers protocol buffer byte-oriented messages, thus introducing low latency. gRPC Network Management Interface (gNMI) is a protocol for configuration manipulation and state retrieval. It is built on top of gRPC and it is described using protobuf and it can use binary or JSON encoding for payload. This allows the usage of YANG data models, allowing the integration of all efforts for defining them in Standard Defining Organizations (SDO).

4.3.3 Message-oriented middleware

A Network Application can utilise message-oriented middleware as APIs for flexible, efficient, reliable, and secure inter-component communications, across diverse platforms and networks if needed. There are a number of standards in this category such as the Advanced Message Queuing Protocol (AMQP), the MQ Telemetry Transport (MQTT), the eXtensible Messaging and Presence Protocol (XMPP) and so on. For instance, AMQP is an open-standard application-layer protocol, able to support a wide variety of messaging patterns including publish/subscribe, request-response, point-to-point etc. In AMQP, the messaging between a server and a client with implementations from different vendors are interoperable. RabbitMQ is an open-source message broker that implements AMQP. It acts as a common channel among multiple microservices for them to publish messages under different numbers of queues available inside the RabbitMQ service bus. Other microservices can then subscribe to interested messages available in RabbitMQ service bus queues. An example of Network Application that employs RabbitMQ as inter-component APIs is shown in Figure 4.

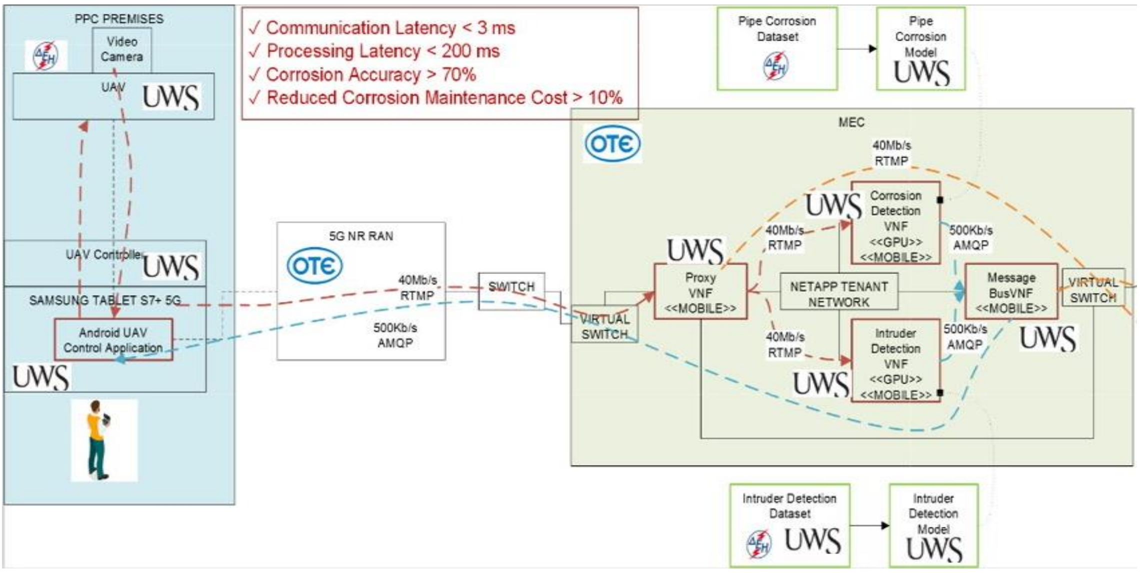

PPC PREMISES
Video Camera
UAV
UWS
UAV Controller
SAMSUNG TABLET S7+ 5G
Android UAV Control Application
✓ Communication Latency < 3 ms
✓ Processing Latency < 200 ms
✓ Corrosion Accuracy > 70%
✓ Reduced Corrosion Maintenance Cost > 10%
5G NR RAN
OTE
40Mb/s RTMP
500Kb/s AMQP
SWITCH
VIRTUAL SWITCH
Pipe Corrosion Dataset
Pipe Corrosion Model
MEC
Proxy VNF <<MOBILE>>
Corrosion Detection VNF <<GPU>> <<MOBILE>>
NETAPP TENANT NETWORK
Intruder Detection VNF <<GPU>> <<MOBILE>>
Message BusVNF <<MOBILE>>
Intruder Detection Dataset
Intruder Detection Model

Figure 4: Network Application example that employs AMQP/RabbitMQ for APIs

In this Network Application example, the key components including a Proxy VNF (for delivering video feeds to be processed for detection), a Corrosion Detection VNF (for detecting corrosion on the surface of industrial pipes based on the video feeds), and an Intruder Detection VNF (for detecting intruders to industrial premises based on the video feeds) communicate with each other using a Message Bus VNF, which is a dockerised RabbitMQ. In particular, the detection results in the terms of a bounding box, the detected class, and the detection confidence are published to the message bus for interested components.

5 “as a Service” Network Applications class

Anything as a Service (XaaS) is a term used to describe a wide range of cloud computing services that are offered to users over the Internet. The term "X" in XaaS represents the specific service or application that is being delivered, such as Software as a Service (SaaS), Platform as a Service (PaaS), or Infrastructure as a Service (IaaS).

The concept behind XaaS is to provide users with a flexible and scalable way to access technology services and resources without having to invest in and maintain their own hardware or software. This model allows users to pay for the services they need on a subscription or pay-per-use basis, which can be more cost-effective than traditional models of technology acquisition.

XaaS is also known for its ability to support rapid innovation and experimentation, as users can easily test new services and applications without having to make a significant upfront investment. Additionally, XaaS providers are responsible for managing and maintaining the infrastructure and resources that are required to support the service, which allows users to focus on their core business activities rather than IT infrastructure management.

5.1 Different models of XaaS

There are several different models of Anything as a Service (XaaS), each of which provides users with access to different types of technology services and resources. Here are the most common models of XaaS:

- Software as a Service (SaaS): SaaS provides users with access to software applications and services over the internet, eliminating the need for users to install and maintain the software on their own devices.
- Platform as a Service (PaaS): PaaS provides users with a platform for developing, testing, and deploying applications without the need to manage the underlying infrastructure.
- Infrastructure as a Service (IaaS): IaaS provides users with access to virtualized computing resources, such as servers, storage, and networking, allowing them to build and run their own applications and services in the cloud.
- Data as a Service (DaaS): DaaS provides users with access to data on demand, allowing them to access and use data resources without having to manage the underlying infrastructure.
- Desktop as a Service (DaaS): DaaS provides users with virtual desktops and applications that are hosted in the cloud, allowing them to access their desktops and applications from anywhere with an internet connection.
- Security as a Service (SECaaS): SECaaS provides users with security services, such as antivirus, firewalls, and intrusion detection, delivered over the internet.

These different models of XaaS provide users with a range of technology services and resources that can be tailored to meet their specific needs and requirements.

5.2 Network as a Service

NaaS paves the way for transforming telco networks into programmable service platforms, enabling the integration of the network with 3rd party applications, with direct and open interactions between them. This is a win-win situation for the two stakeholders involved. For operators, this represents a business opportunity to generate new revenue streams, and one of the ways to monetize investment made in infrastructure (fiber, edge computing and 5G). For 3rd parties, it allows them to flee from traditional over-the-top, best-effort service delivery approaches, tapping now into offered capabilities to provide enhanced user experiences and contribute to digital ecosystem with new services.

In the NaaS model, the provider hosts and manages the network infrastructure, including routers, switches, firewalls, load balancers, and other networking components, and users can access and configure these resources through a web-based portal or API. Users can scale their network resources up or down as needed, pay only for the resources they use, and avoid the upfront capital expenses of purchasing and maintaining physical networking equipment.

NaaS can provide a range of networking services, such us:

- Virtual Private Network (VPN): NaaS providers can offer VPN services to users, allowing them to securely connect to the provider's network and access resources from anywhere with an internet connection.

- Software-Defined Networking (SDN): NaaS providers can offer SDN services, which allow users to configure and manage their network infrastructure through software rather than hardware.
- Bandwidth Management: NaaS providers can offer bandwidth management services, allowing users to allocate network resources based on their needs and optimize network performance.
- Network Security: NaaS providers can offer network security services, such as firewalls and intrusion detection and prevention, to protect users' networks from cyber threats.

NaaS enables CSPs to deliver on transformative, flexible consumption of Network Services, providing a flexible, scalable, and cost-effective way to manage their network infrastructure, allowing them to focus on their core business activities rather than IT infrastructure management. The NaaS offers these features:

- On-demand self-service: Customers can automatically provision network functions and services without any human intervention on the CSP side.
- Network agnosticism: Service offerings and service requests are abstracted so network implementation can be configured any number of ways, even across multiple providers.
- High resource availability: NaaS services are intent-based, with location-specific performance characteristics such as low latency or the ability to attach to resources such as mobile devices.
- Measurement: Network systems automatically control and optimize resource use through metering and measurement, with transparent reporting for CSPs and customers.
- Assurance: Network functions and services are defined with service-level agreements (SLAs) or service performance metrics.

5.3 Implementation examples

5.3.1 VITAL5G

This section is describing the VITAL-5G NaaS model approach for the 5G E2E resources provisioning and services implementation, through which the users get access to the cloud and 5G networking resources and services through secured connections. The VITAL-5G is proposing a NaaS model and architecture which allows users to easily manage and customize their services and network resources through APIs (on-demand services), by using the already existing testbed’s infrastructure high available resources(network agnostic and high availability), without the immediate need of physical resources extensions, end-to-end network and services monitoring (measurements) and services performance assurance(S:As), as highlighted in Figure 5.

The Vital-5G concept spans 3 different pillars, the 1st pillar is the T&L facilities & 5G-enabled use cases, 2nd pillar is the 3 different 5G testbeds and the 3rd pillar is the VITAL-5G Experimentation Platform & Open Repository, described as:

- VITAL-5G Experimentation Platform & Open Online Repository, composed by Network Applications development, Onboarding, Deployment and Experimentation tools
- 5G-Testbeds, the 3GPP Release 16 5G Stand Alone testbeds in Antwerp, Athens and Galati
- T&L facilities & 5G-enabled use cases, the automated vessel transport, Warehouse/freight logistics, Data-enabled assisted navigation

There are two key elements of the VITAL-5G project in relation to the NaaS model implementation, the VITAL-5G Platform as an overall open platform implementation and the VITAL-5G Portal. VITAL-5G Platform includes the VITAL-5G Portal, the VITAL-5G Catalogue, and the Management Backend systems. Vertical users and/or vertical applications access the VITAL-5G Platform to conduct experiments using the VITAL-5G toolset and Network Applications. The VITAL-5G Platform interfaces with the VITAL-5G testbeds in order to enable provisioning and testing of 5G-enabled services. VITAL-5G Portal is a core element of the VITAL-5G Platform and is a collection of tools and functions for Network Applications and service life cycle management (LCM). The portal allows the onboarding, instantiation, monitoring and benchmarking of Network Applications and (T&L) vertical services. Users use the VITAL-5G Portal to run experiments via a dashboard or programmable API. The intent-based Network Application descriptions from vertical actors are automatically translated into 5G slice and Network Function Virtualization (NFV) service descriptions and lifecycle management actions. The VITAL-5G Portal contains tools for Key Performance Indicator (KPI) monitoring, analysis and diagnostic. The VITAL-5G platform is using the 5G Catalogue, a core element of the VITAL-5G Platform that facilitates the design, onboarding, and validation of Network Applications in target environments. The VITAL-5G Catalogue is used to share and compose Network Applications for complex services and implements a Network Application and service catalogue where internal or 3rd-party software developers can onboard their own applications.

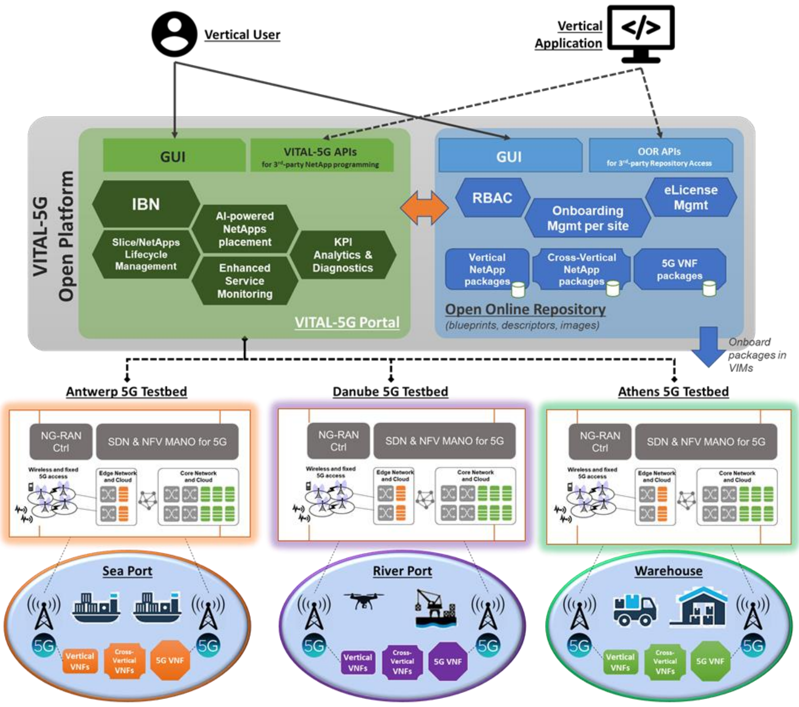


Figure 5: VITAL-5G's Concept & Approach

Basically, the 5G network is composed by several components and layers, as 5G RAN, Core, Transport network, MEC and Virtualization layer, including the specific hardware and software that ensure the 5G testbeds the end-to-end capabilities.

The VITAL-5G is providing an open, virtualized 5G-enabled testing and validation experimentation facilities (Athens, Antwerp and Galati), benefit and showcase the added value of 5G connectivity and customized and virtualized access to network and T&L infrastructure. The VITAL-5G testbeds and facilities, 3GPP Rel. 16 architectures, are described in Figure 6.

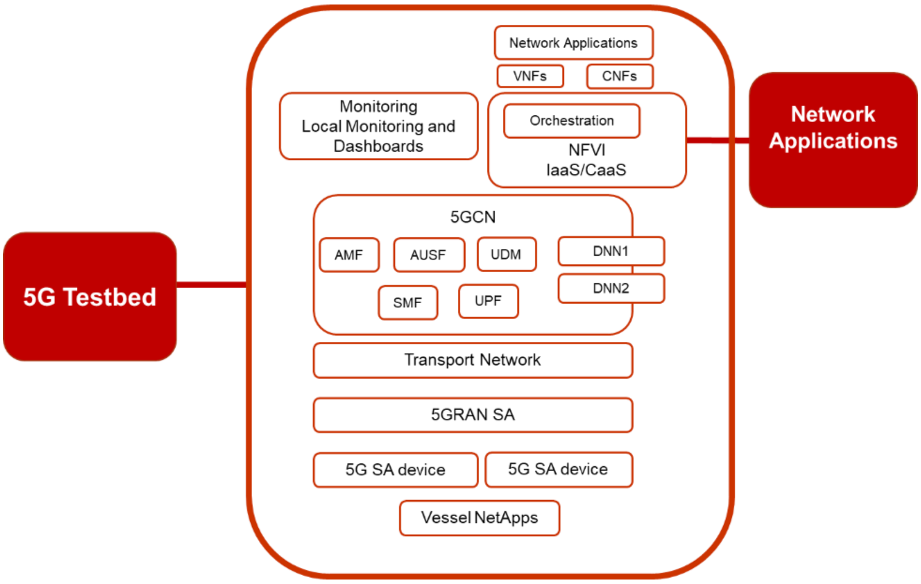


Figure 6: VITAL-5G testbeds SA components

VITAL-5G is providing a comprehensive overview of the technology and 5G solution, based on the facility overall concept, providing the envisioned architecture that should answer to the VITAL-5G and the Network Application concepts. The VITAL-5G E2E facility architecture, represents the innovative technical solution and implementation framework for the 5G communication services required by the T&L industry's verticals. The architecture definition is trying also to cover today's existing mismatches between Telco infrastructure's and verticals: (1) where network services can be deployed, (2) capabilities of dynamic services setup over 5G infrastructures with respect to testbeds available resources and interfaces interconnections and (3) new Vertical's Network Application interactions and trials in real T&L flexible infrastructures.

The VITAL-5G expose to the end-users and Network Application developers, including possible 3rd parties' developers, the 5G network resources for the end-to-end services deployment. It offers RAN, Core and MEC capabilities, network slicing services (eMBB, URLLC) and virtualized resources in the containerized or virtual machines environment (CaaS and IaaS approach). VITAL-5G implements a VITAL-5G Platform level concept, defining Platform-to-testbeds interfaces, for all the three testbeds. The VITAL-5G platform interacts with the various testbeds through a unified South-Bound Interface (SBI) implemented through common APIs, while the translation between the specific interfaces implemented by the testbed management, monitoring and orchestration platforms and these common APIs is handled through testbed-specific plugins. Such plugins, when required, implement the translation between the messages, protocols and information models adopted in the particular testbed and the common messages expected at the unified SBI of the VITAL-5G platform, seamless interoperability of the various testbeds with the centralized platform. The centralized platform offers controls of the testbed's resources, as described in Figure 7, the programmable APIs dynamically creates new custom Iaas/CaaS resources and network services slices, based on the available resources in the testbeds. For example, when provisioning a new vertical service, the VITAL-5G platform will decide whether to ask for a new slice or select an existing one depending on the capabilities offered by the target testbed.

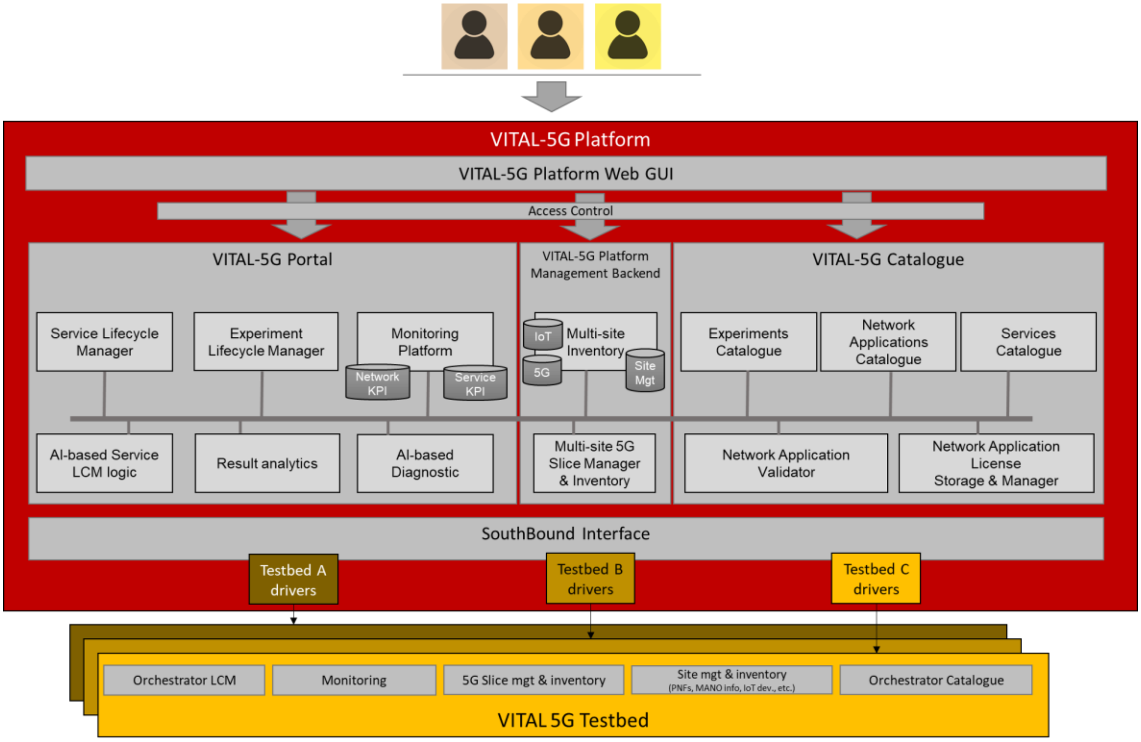


Figure 7: VITAL-5G Platform and VITAL-5G testbeds interfaces

The following functions are supported by VITAL-5G, that supports the Applications Package Onboarding, Vertical Services Instantiation, Experiment Creation, Execution and Monitoring, as follows:

- Network slice management, for creation, selection and configuration of network slices in the 5G infrastructure deployed in a VITAL-5G testbed (Slice_conf interface);
- Onboarding of Network Application and VNF/CNF packages and network service descriptors for vertical services in the orchestration platform at the edge/cloud computing infrastructure of a VITAL-5G testbed (Nfvo_cat interface);
- Provisioning and lifecycle management of network services for the instantiation of vertical services composed by Network Application (Nfvo_lcm interface);
- Configuration (Mon_conf) and retrieval through a publish/subscribe mechanism (Mon_pub) of monitoring data, for both service and infrastructure KPIs, including 5G network KPIs and metrics related to the consumption of computing resources.

For NaaS approach and 5G implementation, the VITAL-5G Platform ensures to the end users a rich set of capabilities in terms of automation, as Application onboarding, Test Cases blueprints, Virtualized or Containerized network functions, Verticals services and experiments blueprints and network services, highlighted in Figure 8.

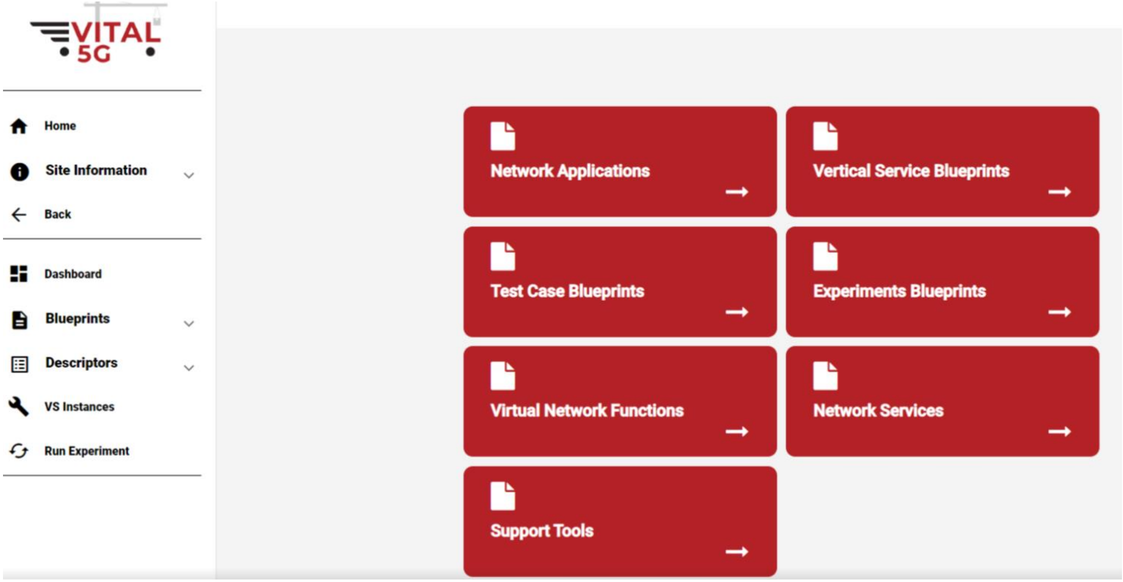


Figure 8: VITAL-5G Platform onboarding capabilities

In the NaaS provided model, there are offered resources and capabilities to the users, as the available 5G resources of the testbeds in terms of virtualization capabilities, VNFs available resources, used as templates by the Network Application developers or 3rd party experiments, service automation, monitoring and dashboards visualization, as an example of available capabilities being described in Table 1, with a focus on the network slicing and monitoring services, considered as key novelties provided to the end customers

Table 1: VITAL-5G available resources pool example

Capabilities VITAL-5G Testbeds	VNF capabilities (up to)			VMs/ CNFs	Virtualized Infrastructure resources			MANO ETSI	Services
	Cores/ VNF	RAM/ VNF	Storage/ VNF	Concurrent VMs	CPU (vCPU)	RAM (GB)	Storage (GB)	Details Network Application	Details
Testbed's	16	32	40	200	3200	6400	8000	OSM IaaS[1]/CaaS[2] BareMetal (GPU[3])	Slicing & monitoring

The provided network slicing concept is characterized by a logical network that provides specific network capabilities and characteristics, being composed of different 5G network domain components, as RAN, Core and transport. In VITAL-5G NaaS model, it is implemented the network slicing, the QoS differentiation, resource management between slices, support for UE associating with multiple network slices simultaneously and resource isolation between slices (QoS Flow Identifiers) and service flow preservation through 5QI mapping to transport DSCP. The Network Slicing Management is an important feature of the VITAL-5G, delivered in the form

[1] Openstack based

[2] Kubernetes based

[3] GPU if required by use case owner

of the Multi-site 5G Slice Management and Inventory component, a software component that performs real-time management of network slices in all three 5G testbeds:

- Real-time overview of 5G slices in VITAL-5G testbeds
- Dynamic provisioning and life-cycle management of 5G slices
- Optimal 5G slice selection during Vertical Service Instantiation procedure

The Network and Service monitoring is another NaaS's VITAL-5G concept, to perform real-time network and service monitoring by collecting the data that reflects on the network and service performance within the VITAL-5G testbeds, and making that data available to the users or other platform components that are responsible for analytics (used for services experiments), by performing the next activities:

- collects performance metrics such as network, service, platform, and infrastructure, in real-time with the predefined frequency (depending on the metric type), by interfacing with the distributed 5G testbeds
- exposes the performance metrics to the other VITAL-5G platform components such as AI-based Diagnostics and Result Analytics, to perform advanced analysis of Key Performance Indicators (KPIs) and to trigger various life-cycle management operations on the vertical services in order to improve service performance

In the described VITAL-5G NaaS model, the testing and validation procedures of the services are provided by a Testing as a Service (TaaS), allowing for the automated execution and evaluation of network-oriented and service-oriented testcases, based on high-level test plan descriptions of the intended experiments, based on the experimentation service principals and available platform tools. In the NaaS model, the monitoring capabilities are provided to the experimenters in a central manner via the VITAL-5G portal, through which experiments are defined, configure the specific testcases of interest, define the validation parameters and receive the validation report.

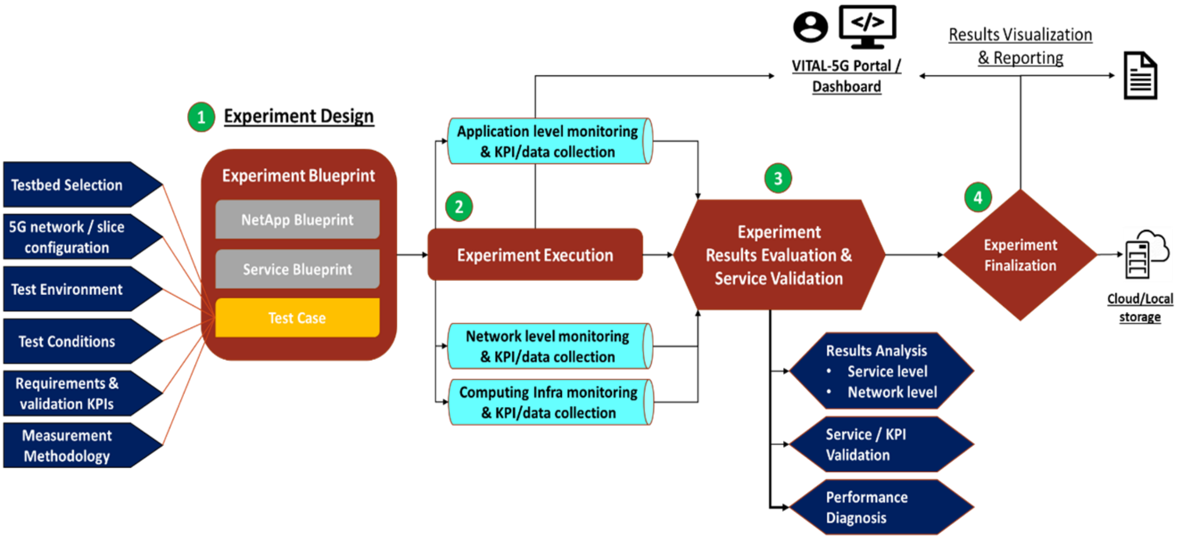

VITAL-5G Portal / Dashboard
Results Visualization & Reporting
1 Experiment Design
Testbed Selection
5G network / slice configuration
Test Environment
Test Conditions
Requirements & validation KPIs
Measurement Methodology
Experiment Blueprint
NetApp Blueprint
Service Blueprint
Test Case
Application level monitoring & KPI/data collection
2
Experiment Execution
Network level monitoring & KPI/data collection
Computing Infra monitoring & KPI/data collection
3
Experiment Results Evaluation & Service Validation
4
Experiment Finalization
Cloud/Local storage
Results Analysis
• Service level
• Network level
Service / KPI Validation
Performance Diagnosis

Figure 9: VITAL-5G testing procedure in the NaaS model

In VITAL-5G, constant experiment monitoring is available at the VITAL-5G platform, while after the experiment(s) have conducted, a detailed performance evaluation and service assessment phase is enabled, proving the successful execution and monitoring of the experiments, the metrics collection and storage and the post-processing and evaluation of the metrics, in order to validate the service under test, according to the validation KPIs provided by the experimenter.

The Testing & Validation framework developed by VITAL-5G governs the interconnection and interfacing of the various testbed and platform modules, enables necessary Network Application and Service onboarding and configuration, defining of specific tests in details, the data/metric collections and their processing, as seen in Figure 10.

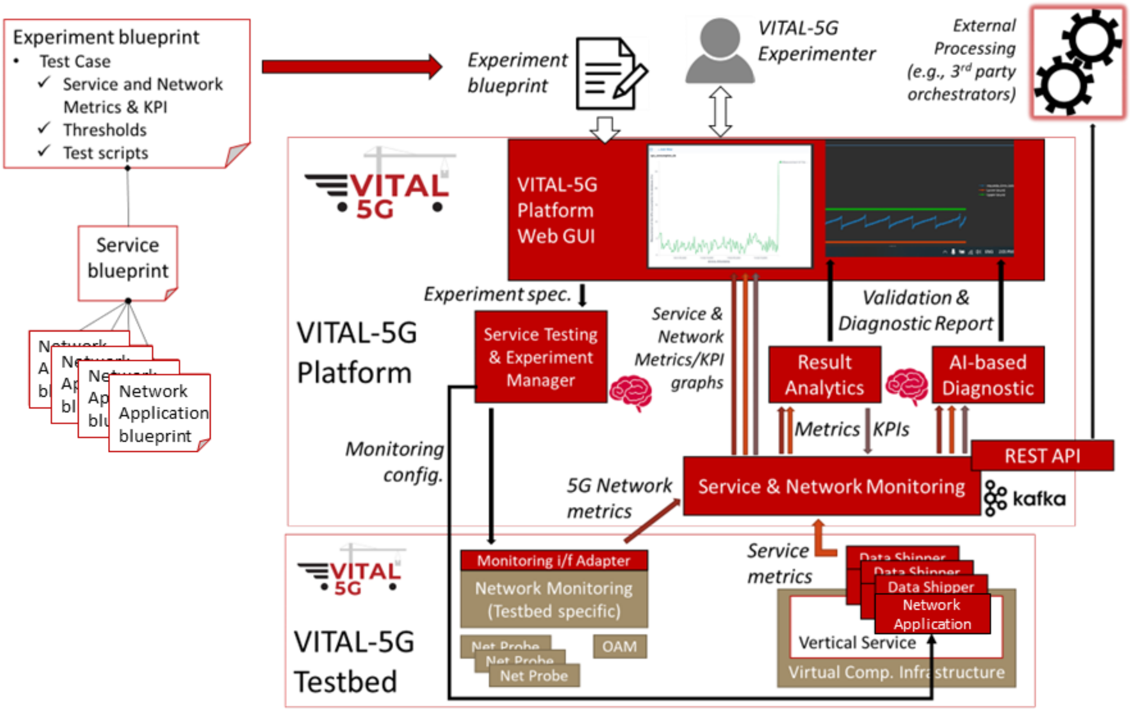


Figure 10: VITAL-5G Testing & Validation Framework

For sake of clarity related to the complete VITAL-5G NaaS implemented model, focusing on the element’s role in the testing and use cases validation phase, the following elements are described in details:

- Service Blueprint (Network Application blueprint) includes the list of the service components, which in VITAL-5G are Network Applications, with details endpoints and interconnectivity
- Testcases, that provides all the necessary experiment settings, to perform network or service-oriented experiments and to make them reproducible, including testbed selection, network/slice configuration, test environment, conditions and route, validation KPIs, measurement methodology and more
- Experiment Blueprint, aggregated information that holistically define the services, execution environments and experiment conditions over the VITAL-5G platform and testbeds
- Service, Testing & Experiment Manager, responsible for the creation and execution of the experiments, including the collection and storage of the experiment results, activated by the vertical user
- Service & Network Monitoring (and KPI collection), a centralized monitoring platform that allows for the collection of data from the various VITAL-5G facilities during experiment execution, monitors the progress of the experiment and of the specified KPIs
- Testbed monitoring tools, the testbed-specific monitoring tools engaged by each of the VITAL-5G facilities, to monitor the defined metrics during experiment execution and to record the specified network and infrastructure level KPIs

- Results Analytics, the module that receives the collected metrics from the Service & Network Monitoring module, and is responsible for the evaluation of the results after the experimentation cycle and the validation of the selected service
- Portal/GUI & visualization tools, VITAL-5G portal and Graphic User Interface (GUI), it allows for the user input and experiment definition and provides feedback and insights to the experimenter during and after the experiment finalization
- Performance Diagnosis, the module used to identify under-performing elements in the experimentation chain or detecting anomalous behaviour of one of the elements, based on the collected metrics, performs root cause analysis to identify the exact issue, in order to assist the experimenter in improving the performance of their end-to-end service

for the NaaS model, VITAL-5G foresees an offline experiment preparation phase during which testbed owners and experimenters agree on the experiment to run, the dates and the resources required. This allows to plan in advance the availability of the required computing and network resources to execute the targeted experiment on a specific date. In any case, if at execution time an experimenter requests the instantiation of a service which requires more resources than the ones currently available, the instantiation of the service will fail either during the provisioning of the network services or during the allocation of the 5G slices associated to the service.

5.3.2 5GINDUCE

The 5G INDUCE platform is specifically conceived for exploiting, self-optimizing, automating, and simplifying the management of Network Applications onto 5G and beyond (B5G) infrastructures. At a glance, the proposed platform aims to hide the complexity of B5G environments to services' developers, by making their experience fully equivalent to the one on well-known cloud computing systems, while unlocking the full potential offered by network slicing, edge computing and network function virtualization.

5.3.2.1 Network Application deployment and management platform

The 5G-INDUCE platform supports the so-called "separation of concerns" between the end-user driven Network Applications and the network platform administrative domains. Under this concept, the lifecycles of Network Applications and network services are managed by **separated orchestration tools**, devoted to their specific (applicative or network) administrative domain owned/used by their own stakeholders, e.g., a vertical industry and a B5G network operator respectively. These orchestration tools are the **Network Application Orchestrator (NAO)** and the **Operations Support System (OSS)** and need to strictly cooperate to allow coherent deployment and use of resources and of their configuration. Two control loops are identified deployment loop and the runtime management loop.

During the **deployment phase,** and upon the reception of Network Application deployment requests by vertical end users (through a user-friendly graphical user interface), the NAO requests, negotiates and obtains from the OSS both the needed computing resources at the edge facilities (where Network Application components require to run), as well as the connectivity among such resources and User Equipment (UE). The OSS analyses the operational and performance (soft and hard) constraints expressed by the NAO slice request, and, consequently, selects the most suitable computing facilities and network services complying with the requirements. The initial request produced by the NAO is called "slice intent" and contains the application graph annotated with QoS and operational requirements. Special links in the graphs represent the connectivity between front-end Network Application components and UEs (i.e., the PDU sessions to be realized by the radio mobile network and dedicated to the Network Application). When the NAO accepts the solution provided by OSS, the OSS provides back the "materialized slice," which is the set of needed network services and resources instantiated and configured, ready to host the Network Application. Then the deployment and the management of the Network Application functions is performed by the NAO.

During the **runtime management phase**, the platform supports the modification and reconfiguration of the slice within its lifecycle and enables advanced operations to deal with UE mobility and dynamic QoS/operation Network Application requirements. In more detail, 5G-INDUCE Network Applications (or some of their components) can be lively scaled and relocated onto computing facilities in new geographical areas or at different network infrastructure aggregation levels in a transparent and smooth fashion. Traffic from and to UEs is steered accordingly (by coherently updating the configuration of network slices and related network services), while the platform can adapt the Network Application geographical scope, and properly scale the components and network services on each area depending on the number of hosted Network Applications, the local workload, and the dynamic QoS/operational requirements. This geographical scope modification, on one side, entails the capability of the OSS to select/deselect proper resources at edge facilities updating/creating/deleting network services, and to dynamically reconfigure network slices to transparently and smoothly redirect UE incoming and outgoing traffic, during the reconfiguration phases. On the other side, through a proper synchronization with the OSS, the NAO can update the Network Application instance graph by deploying/removing Network Application components in the selected edge facilities.

Furthermore, a key consideration in the development of the 5G-INDUCE platform relates with the south bound OSS interfacing and its ability to **flexibly handle any NFV virtualization levels**. Network services can be composed by a mixture of VNFs realized not only with Infrastructure-as-a-Service resources, but also with cloud-native containers over platforms offered as platform-as-a-service (e.g., 3GPP 5G core network functions over Kubernetes clusters), as well as physical programmable/configurable devices (e.g., gNodeBs, eNodeBs, etc.). The number and the types of network services are extended by including cloud-native 3GPP 5G network functions, and zero-touch procedures to enable the dynamic management of slices, traffic steering, etc. In this respect, the OSS at its southbound interface includes a special module, namely the NFV Convergence Layer (NFVCL), which fully drives NFV service orchestration along all the lifecycle phases. NFVCL communicates with an external NFV Orchestrator (e.g., ETSI Open-Source MANO – OSM) through standard ETSI NFV interfaces. During Day-0 operation (infrastructure resource discovery phase), NFVCL produces and onboards the ETSI SOL006 descriptors of services and of related Virtual/Physical/Container Network Functions onto the NFVO by defining the needed number of virtual links and of virtual resources to be applied. In Day-1 operations (deployment phase), the NFVCL requests the NFVO to instantiate network services selecting the computing facilities, and the networks where to attach network functions. At Day-2 (runtime reconfiguration phase), the NFVCL produces the configuration files and commands for each of the deployed VNFs and applies them through the VNF Managers at the NFVO. Finally, through a Metal-as-a-Service (MaaS) approach, the OSS is also provided with the capability to manage and to terraform bare-metal resources (i.e., servers, switches, routers, etc.) to install, to build, and to configure complex and distributed IaaS and PaaS environments, where to host VNFs and Network Applications components.

It is noted that the 5G-INDUCE platform has been specifically designed to adaptively exploit the programmability level offered by the underlying network infrastructure(s) by enabling the aforementioned capabilities when and where possible or necessary.

5.3.2.2 Platform architectural details and Network Application handling processes

The 5G-INDUCE platform is designed and developed according to the overall system architecture depicted in Figure 11 and Figure 12 and representing the NAO and OSS parts of the platform respectively. Details for each one of the platform tools and the supported Network Application handling functionalities are provided in the following paragraphs.

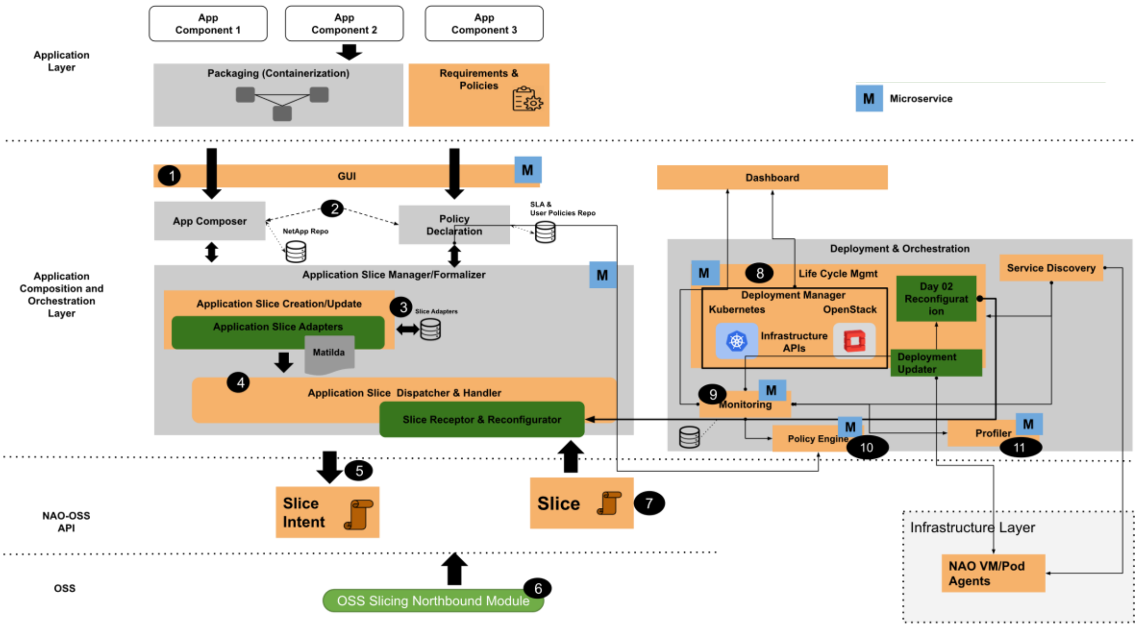


Figure 11: 5G-INDUCE NAO design and Network Application onboarding, registration and management functionalities.

The front-end blocks of the NAO are the interface with end users and the service providers. It involves all the application layer tools needed for the registration, creation and management of Network Applications. These front-end blocks are combined into a multifunctional panel (NAO GUI) which aggregates several interfaces, like from the wizard which allows to drag and drop application components, to policy definition forms and various overseeing and reporting mechanisms.

The back-end blocks of the NAO are responsible for the deployment, the lifecycle operations, and the monitoring of applications, also including the management of the interactions with the OSS. At first, Network Application providers register their application components and their operational characteristics (step 1). A Directed Acyclic Graph (DAG) is then composed through the GUI panel (App composer) including the specific networking and localisation features of the Network Application and the policy declaration criteria per Network Application (step 2). After that the entire application deployment procedure follows.

The composed Network Application graph is converted into an application slice intent adhering to the resource and networking parameters that can be controlled by the OSS (step 3). In principle the NAO may interface with different types of OSS handling parameters in different format according to the targeted infrastructure or underlay technologies. Therefore, an adapter module is inserted to provide such interconnectivity independently to the underlay OSS handling type. Next, a slice dispatcher micro-service propagates the slice intent to the appropriate OSS tool instance (step 4). This functionality is adopted to extend the NAO towards multi-domain and multi-vendor edge deployments supporting potentially the creation of service chains across different vertical industries. The created slice intent is sent to the advertised interface (step 5) and consumed by the OSS northbound interface module (step 6). Upon successful creation or identification of allocated network slice and also reservation of the requested computational resource at the targeted nodes by the OSS, a slice is returned back to the NAO (step 7) with the corresponding targeted network addresses and port numbers. This information propagates through the slice handler to the deployment manager (step 8) which initiates the deployment of the corresponding Network Application images to the targeted resources according to the order defined by the end user and the policies.

The runtime management process follows the deployment cycle described above and is activated either directly by the end user, in the form of a slice update (repeating steps 4,5,6,7,8) or in an automated fashion through the monitoring and policy engine. For the latter case, the procedure relies first on the monitoring micro-service that collects specific networking metrics (from OSS) and application parameters as defined by the end user during the onboarding phase (step 9). A policy engine micro-service analyses the monitoring parameters and combines them with a set of rules extracted by the user defined policy criteria (step 10). At this stage any other analytics and policy handling engine can be inserted (e.g. an AI based decision engine) as long as it is aligned to the deployment manager API. The Profiling mechanism is added next (step 11) for supporting various profiling aspects in application component and application graph level. The control loop closes with the propagation of the generated slice update decision for the targeted Network Application slice through the deployment manager (step 8).

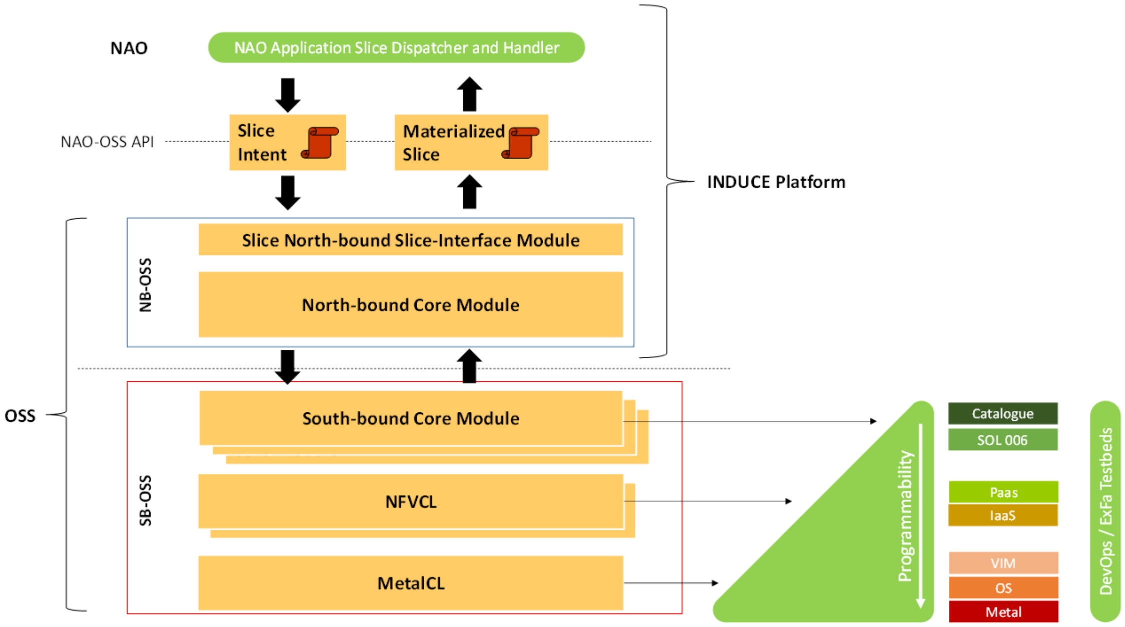


Figure 12: 5G-INDUCE OSS design and Network Application driven infrastructure configuration functionalities

The north bound of the OSS is composed of two main services: the Slicing-Interface and the North-Bound Core services. The Slicing-Interface service is meant to implement the OSS APIs from/towards the NAO that handle the slice intent requests according to the defined data structure for the targeted vertical industries. The North-Bound Core service is in charge of the south bound OSS (SB-OSS) instances onboarding, and the accurate processing and propagation of the slicing requests/replies between the Slicing-Interface service (prior to be sent to NAO) and the relevant type of SB-OSS(es) that can be attached.

In turn, the SB-OSS includes three “chained” services, namely a) the South-Bound Core service, b) the NFV Convergence Layer (NFVCL), and c) the Metal Convergence Layer (MetalCL). The South-Bound Core service is the only mandatory element in the SB-OSS, and it is devoted to the processing of the slice instantiation/modification/de-instantiation requests and the related resources. This service is the key component for providing adaptive programmability; if the NFVCL and the MetalCL services are available, the South-Bound Core can request to them the setup or the change of new or existing network slices/services, and of infrastructure resources (e.g., of OpenStack VIM instances and of the physical servers composing it). In case that bare-metal or virtualization programmability levels in an administrative domain are not exposed to the 5G-INDUCE platform, the South-Bound Core can dynamically request to an external NFV framework the needed slices/configurations, or simply cataloguing the pre-configured resources (e.g., a 5G network slice) statically dedicated to the 5G-INDUCE platform.

The role of the NFVCL within the SB-OSS is to manage the lifecycle of NFV services to provide suitable connectivity to Network Application components and UEs in fully automated and zero-touch fashion. If not provided by the bare metal layer, the NFVCL is also in charge of providing and maintaining cloud-native computing frameworks at edge facilities (i.e., realizing Kubernetes clusters as NFV services).

The MetalLB is the service dedicated to manage and terraform bare-metal resources (i.e., physical servers and hardware network equipment) to create IaaS/PaaS environments compliant with the 5G-platform needs. Also in such a case, this service allows the dynamic lifecycle management of operating systems in the servers, of configurations in network equipment, an of complex distributed applications like OpenStack and Kubernetes.

5.3.3 5GASP

The overall 5GASP facility is composed of several interworking sites, each deployed at a different geographic location and defining a single administrative domain. To provide a unified abstraction for all sites, the necessary experiment modelling and transformations need to be defined so that onboarding, activation and testing can be properly performed not only on any 5GASP facility but also on any NFV/3GPP compliant 5G System, regardless of the internal details. To achieve this, in the context of this project, we propose a unified standards-based model that has the form of a "triplet" of entities triggering a service deployment order, as depicted in Figure 13.

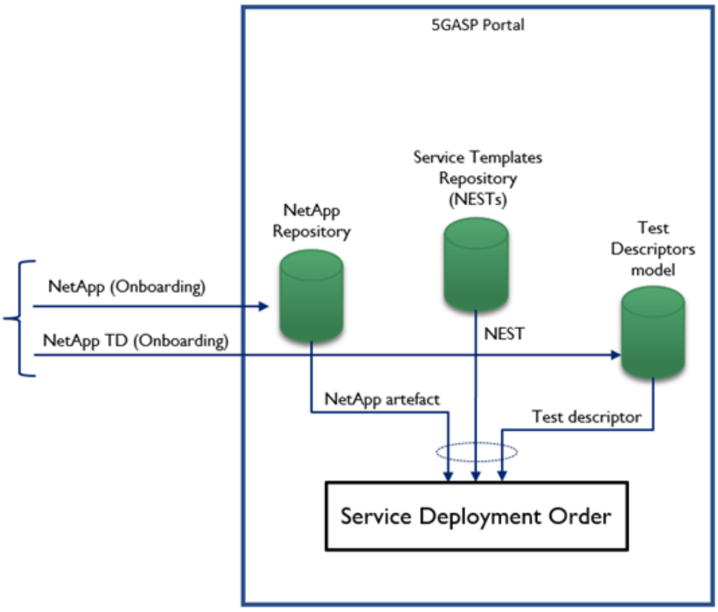


Figure 13: 5GASP experimental triplet model

As seen, the "triplet" consists of the following entities:

- The Network Application Artefact, bearing the link to the actual NSD(s) comprising the Network Application.
- The Network Slice, that is activated by a target 5G facility and provides the host to facilitate the Network Application ‘s requirements.
- The Test Suite, represented in terms of a test descriptor model, that is executed after the activation of the Network Application.

To end up, extensive reference to the aforementioned experimental model can be found in D3.1 [40]

5.3.3.1 Preflight tests

Following the successful onboarding process, modelled by the experimental triplet mentioned in the section above, preflight check service is utilised before the service order fulfilment can be initiated. This service, acting as the actuator of the remaining process, ensures that computationally expensive procedures, like network services' deployments, are not triggered before some elementary standards are met, as described in Figure 14.

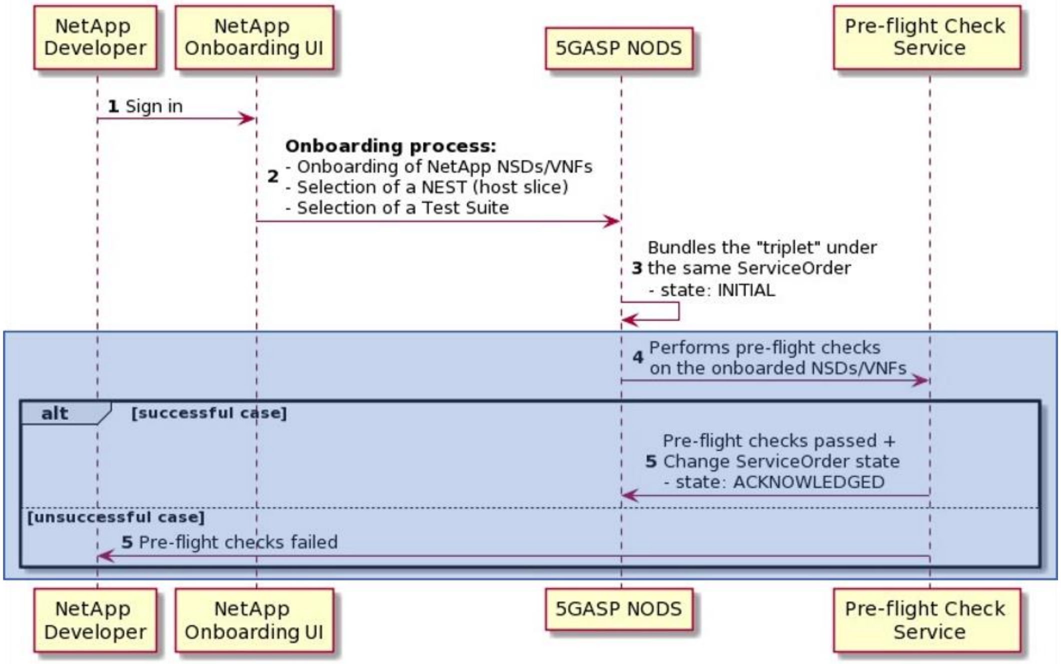


Figure 14: Preflight check procedure

The central portal, i.e. 5GASP NODS, integrates a preflight check service but can also incorporate external ones exposed by APIs. Currently, the internal service is implemented in terms of OSM version recognition and syntax checking. As for today, OSM supports VNF/NS Descriptors designed towards YANG model [4]. Since Release 9, OSM is fully aligned with ETSI NFV SOL006 [39] featuring some augmentations. On that notion, the preflight check service can distinguish SOL006 modelling and thus, identify the corresponding OSM version.

Specifically, the uploaded archived is unzipped (expected packaging format is .tar.gz), and the descriptor is parsed and checked upon SOL006 model formatting. Subsequently, depending on the model formatting and assuming correct syntax, information is acquired about VNFs, such as name, image utilised etc. Finally, the archive is onboarded on the respective OSM. This overall process is depicted in Figure 15.

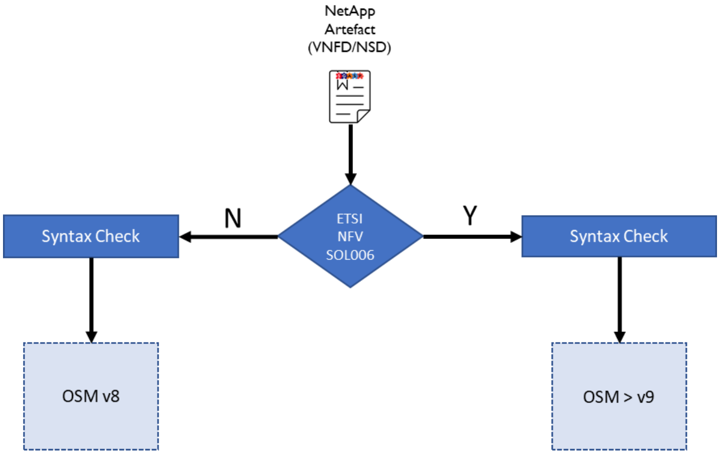


Figure 15: Preflight check service activity diagram

5.3.3.2 Service deployment process

After successful validation of the Network Application, through Preflight tests, the Network Application can enter the Service deployment process on one or more 5GASP facilities according to a matching process handled by the 5GASP portal. The matching process checks the Network Application requirements, including vertical-specific aspects such as infrastructure capabilities to support Automotive and PPDR specific functionality, against the information provided by the 5GASP facility upon registration. In addition, a directory service will be developed to list facilities and capabilities, providing an API for testers to list and discover the desired facility.

5.3.3.2.1 5GASP facility registration

Before the matching process can occur, it is necessary that each facility can describe and register its capabilities in the 5GASP directory service. The directory service will be constituted by a semantically rich document written in YAML format containing available computational, network and vertical-specific resources available for Network Application deployment.

5.3.3.2.2 Service deployment order

Whenever possible, Network Application will be automatically deployed without any time/resource constrain. Whenever Network Application tests require local facility support, a Scheduling Service (using iCal protocol) will provide the means to programmatically schedule local collaboration (e.g. operating the UE). The Scheduling Service will notify all parties involved by replying with an agenda appointment and URL to the collaboration platform (e.g. Google Meet). Network Application requiring such support will only be deployed in the scheduled timeslot. All requests in the platform are processed First-In-First-Out without any privileged access.

5.3.3.3 Service teardown

Network Applications will be automatically shut down after a grace period of 24 hours. All resources allocated will be freed and all data deleted. Network Application developers/testers are the sole ones responsible for retrieving test information data after they finish testing.

5.3.3.4 Network Application orchestration

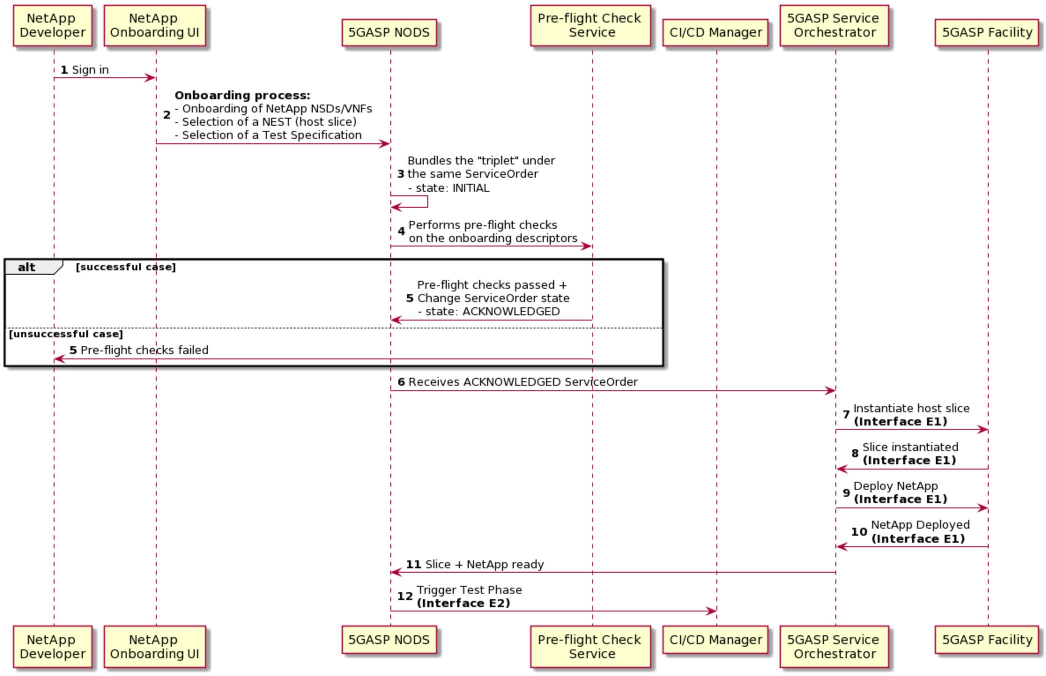

NetApp Developer
NetApp Onboarding UI
5GASP NODS
Pre-flight Check Service
CI/CD Manager
5GASP Service Orchestrator
5GASP Facility
1 Sign in
2 Onboarding process:
- Onboarding of NetApp NSDs/VNFs
- Selection of a NEST (host slice)
- Selection of a Test Specification
3 Bundles the "triplet" under the same ServiceOrder
- state: INITIAL
4 Performs pre-flight checks on the onboarding descriptors
alt
[successful case]
5 Pre-flight checks passed + Change ServiceOrder state
- state: ACKNOWLEDGED
[unsuccessful case]
5 Pre-flight checks failed
6 Receives ACKNOWLEDGED ServiceOrder
7 Instantiate host slice (Interface E1)
8 Slice instantiated (Interface E1)
9 Deploy NetApp (Interface E1)
10 NetApp Deployed (Interface E1)
11 Slice + NetApp ready
12 Trigger Test Phase (Interface E2)
NetApp Developer
NetApp Onboarding UI
5GASP NODS
Pre-flight Check Service
CI/CD Manager
5GASP Service Orchestrator
5GASP Facility

Figure 16: Network Application orchestration steps

The 5GASP Service Orchestrator is responsible for creating a host network slice and for deploying a Network Application based on corresponding descriptions. Both the network slice and the Network Application are described using the *onboarding model* (based on standardised data model) passed to the NODS during the onboarding process (Figure 16 – steps 1 and 2). Unified standards-based onboarding model provides NEST information, Network Application Artifact and Testing Descriptor. Before the description of a certain Network Application and its corresponding network slice is passed to the orchestrator (Figure 16 – step 6), NODS performs necessary preflight checks (Figure 16 – steps 3 to 5). To create a host network slice and deploy the Network Application, the orchestrator communicates with the facilities involved using the interface E1 (Figure 16 – steps 7 to 10). Finally, when the orchestration process is finished (Figure 16 – step 11), the Network Application enters the test phase (Figure 16 – step 12).

5.3.4 5GMediaHUB

This section describes a model for implementation, validation, and verification of Network Application in the 5GMediaHUB project [38]. The project aims to provide a comprehensive platform for testing and validation of innovative 5G-empowered media applications and Network Applications from 3rd party experimenters and Network Applications developers, through an open, integrated and fully featured Experimentation Facility. This facility offers an elastic, trusted, secure, and integrated service execution environment based on open-standards, cloud-based architecture and APIs, supporting multi-tenancy. The architecture, presented in Figure 17, is split into the five main layers such as Application layer, Experimentation Tools layer, Network Application & Slice layer, Infrastructure layer and UE layer. Here, we just review the first two layers.

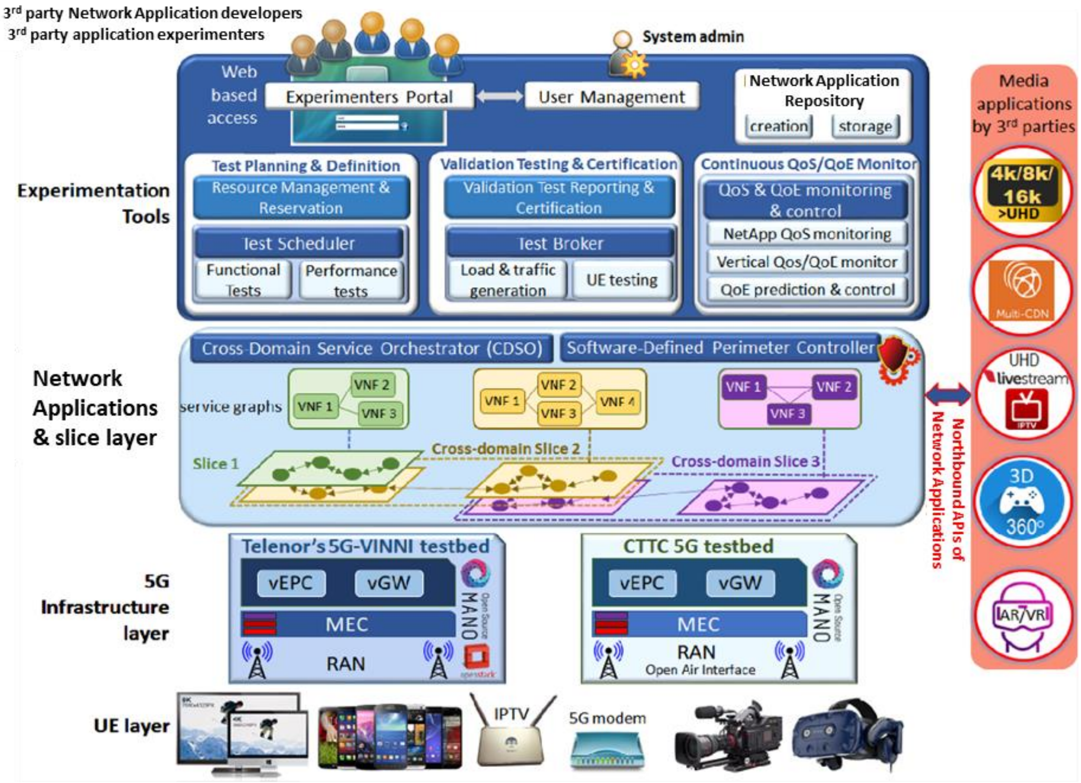

3rd party Network Application developers
3rd party application experimenters
System admin
Web based access
Experimenters Portal
User Management
Network Application Repository
creation
storage
Media applications by 3rd parties
Experimentation Tools
Test Planning & Definition
Resource Management & Reservation
Test Scheduler
Functional Tests
Performance tests
Validation Testing & Certification
Validation Test Reporting & Certification
Test Broker
Load & traffic generation
UE testing
Continuous QoS/QoE Monitor
QoS & QoE monitoring & control
NetApp QoS monitoring
Vertical QoS/QoE monitor
QoE prediction & control
4k/8k/16k >UHD
Multi-CDN
UHD livestream
IPTV
3D 360°
AR/VR
Network Applications & slice layer
Cross-Domain Service Orchestrator (CDSO)
Software-Defined Perimeter Controller
service graphs
VNF 1
VNF 2
VNF 3
VNF 4
Slice 1
Cross-domain Slice 2
Cross-domain Slice 3
Northbound APIs of Network Applications
5G Infrastructure layer
Telenor's 5G-VINNI testbed
CTTC 5G testbed
vEPC
vGW
MEC
RAN
Open Air Interface
Open Source MANO
openstack
UE layer
IPTV
5G modem

Figure 17: 5GMediaHUB Experimentation Facility high-level architecture

5.3.4.1 Application layer

The Application layer in 5GMediaHUB is implemented in accordance with new ETSI MEC Platform-as-a-Service (PaaS) extensions supporting containerized and VM-based workloads. The PaaS follows cloud-native design principles, introduced in [39] to facilitate the utilization of container technologies. Here, the consumer does not manage or control the underlying cloud infrastructure (including network, servers, operating systems, storage, or platform services), but has control over the deployed applications and possibly over application hosting environment configurations. This layer hosts media applications within VMs or Docker containers, offering PaaS capabilities and abstracting SDN and NFV complexities. Media application developers simply provide their applications as VM or container images, with their respective metadata (e.g., Helm Charts). They can leverage the Network Applications' Northbound APIs to interact with underlying Network Applications in a plug-and-play, NFVI-agnostic manner.

5.3.4.2 Experimentation layer

This layer consists of multiple applications providing the necessary functionality that allows users to rapidly build and test applications within a realistic 5G environment. It is the entry point for the majority of users, be they Network Application and/or vertical application developers. The tools provided allow the development, deployment, test and validation of applications, including any Network Applications to be included in the core 5G network. Furthermore, these experimentation tools allow rapid testing and verification of applications in the 5G environment under real-world conditions, thereby shortening delivery cycles. The tools abstract the complexities of network management and orchestration, removing the need for facility users to develop the associated domain-specific knowledge themselves, thus allowing users to concentrate on their applications and related services.

The experimentation tools layer consists of:

- Network Applications Repository: providing the user with the ability to design, build and on-board Network Applications to the CDSO.

- Experimenters Portal: managing the planning and execution of test cycles.
- User Management Module: providing user validation and verification for the various 5GMediaHUB portals and databases.
- Test Planning and Reservation Engine: responsible for setting up experiments in the testbed infrastructure.
- Validation Testing Engine: performs automatic validation and verification of Network Applications, certifying that they meet required KPIs and SLA guarantees.
- Continuous QoS/QoE Engine: responsible for analysis, monitoring and performance control of media-based applications.

The validation testing engine offers an automated way to validate the functional and non-functional specifications of Network Applications under test, producing precise and fast results. KPIs and thresholds required for validation are configured in the Jenkins file. Meeting the KPI thresholds verifies the Network Application, and the result is published in the service catalogue of the Network Applications repository.

The engine is based on an automated DevOps approach that defines a separate workflow for each Network Application. Upon triggering the engine for a particular application, the relevant workflow executes all necessary steps to complete the validation process. Workflow triggering can happen in two ways:

- automatically, when a Network Application is pushed to the repository,
- manually, either via the Experimenters' portal or the Validation Engine's portal.

The validation results (whether Passed or Failed), together with the corresponding KPIs, are propagated to the experimenters' portal, the Network Applications repository, and the service catalogue to maintain the validation status of the Network Application. Results are also accessible through the Validation Testing Engine's dedicated portal.

5.3.5 5G-IANA

5G-IANA aims to build an Automotive Open Experimental Platform (AOEP) to bring up the 5G potential of orchestrating Vertical Services based on virtualized network slices and coordinating distributed edge-to-cloud deployment for the Automotive sector.

The 5G-IANA AOEP provides Small and Medium Enterprises (SMEs) in the Automotive sector an opportunity to create, test, and deploy their services. This will be achieved by providing a set of hardware and software resources (by the AOEP), as well as computational and communication/transport infrastructure, management, and orchestration components, and a Network Applications Toolkit tailored to the Automotive sector, simplifying the design and onboarding of new Network Applications.

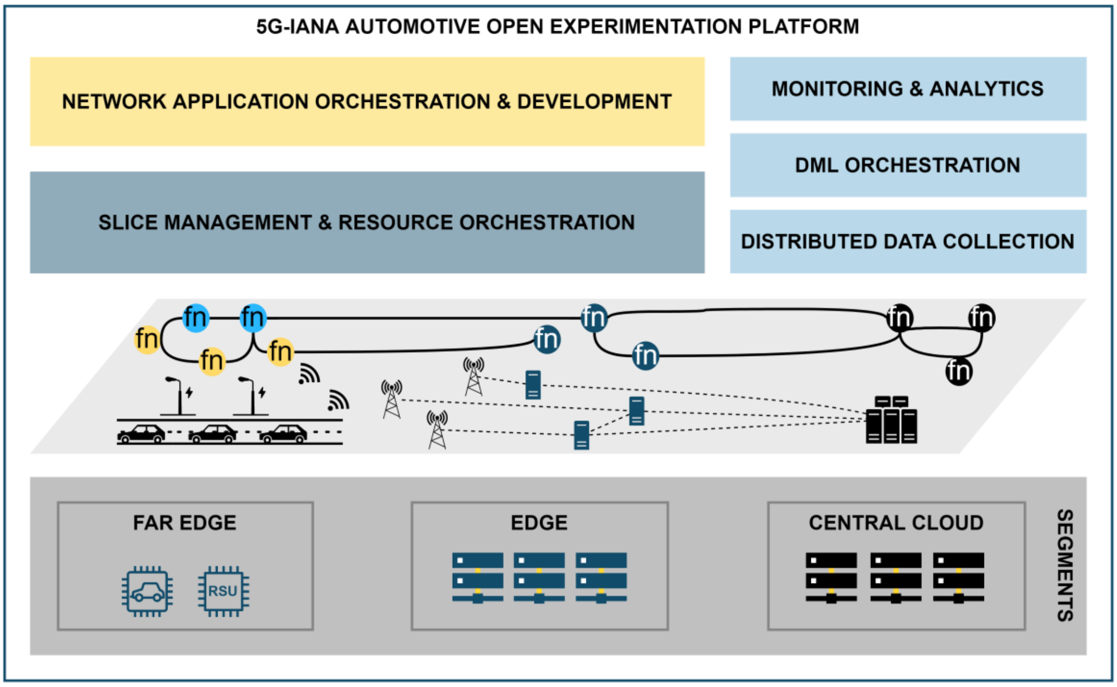

5G-IANA AUTOMOTIVE OPEN EXPERIMENTATION PLATFORM
NETWORK APPLICATION ORCHESTRATION & DEVELOPMENT
MONITORING & ANALYTICS
DML ORCHESTRATION
SLICE MANAGEMENT & RESOURCE ORCHESTRATION
DISTRIBUTED DATA COLLECTION
fn
FAR EDGE
RSU
EDGE
CENTRAL CLOUD
SEGMENTS

Figure 18: 5G-IANA Automotive Open Experimental Platform System Design

The main purpose of the 5G-IANA platform is to streamline and automate the handling of Network Applications on programmable infrastructures, particularly 5G ones. Essentially, this platform aims to simplify the complexities of programmable infrastructures and 5G environments for service developers and providers. In doing so, it is designed to make the development, deployment, and operation of 5G-ready applications (Network Applications) as easy and familiar as it is for cloud-native applications in cloud computing environments.

5.3.5.1 5G-IANA internal modules

The 5G-IANA AOEP integrates a set of functional layers to deliver all the functionalities (Figure 18). The two main layers are the Network Application Orchestration & Development layer (NOD) and the Slice Management & Resource Orchestration.

The NOD layer manages the entire lifecycle of the Network Applications (from the development to the deployment, including the orchestration). It enables the end user (who is typically an automotive industry application developer) to manage the features and deployment of applications. By doing so, it separates the management procedures for the application layer from those for the network layer. This results in an interface that connects with the underlying systems responsible for creating and managing slices, allowing compatibility with any network orchestration solution and their corresponding slice management subsystems. To aid in the re-usability of Network Applications, the 5G-IANA format includes also service-level details such as interface specifications and documentation. This enables Network Application sharing and composition with other Network Applications to form advanced Vertical Services in a distributed chain. Furthermore, each Network Application is described by a template that describes all the needed information to describe an application, such as the principal characteristics necessary for the correct functioning of the Network Application within the required 5G slice profile, which are used by the Orchestrator component of the ecosystem to deploy the Network Application using the appropriate slice for that.

The Network Application toolkit is the main entry point for the design, development, and deployment of a Network Application, leveraging a simple, yet comprehensive GUI and DevOps pipeline and mechanism to ensure the standardization, the correctness, and the stability of the Network Applications and of their components. Once a Network Application is completed, the Application Orchestration component, with the help of the Slice Management & Resource Orchestration, will negotiate the slices and deploy the Network Application over the 5G infrastructure. The Slice Management & Resource Orchestration implements the mechanisms to manage and orchestrate the available Edge and Far-Edge resources, while the Distributed Machine Learning (DML) Orchestrator is responsible for managing the functional components that determine the most suitable resources for supporting the operation of the desired Vertical Service. By integrating with the Slice Management & Resource Orchestration, the DML orchestration input can be used as a constraint for making decisions about the allocation, arbitration, and provisioning of DML compute resources throughout the 5G-IANA multi-segment compute infrastructure. Further discussion can be found in [41].

The Monitoring & Analytics and the Distributed Data Collection aim to gather and present monitoring data related to the usage of resources from the virtualized infrastructure, including on-vehicle MANO, such as compute, memory, and network, as well as the behaviour of deployed Network Applications. The core services, including Orchestration, Policies, and Profiling utilize Monitoring & Analytics, which is accountable for accumulating data based on active monitoring probes and executing data management operations (such as calculating the average values in specific timeframes). Additionally, it allows tenants access to both past and real-time monitoring data, which they can access through the 5G-IANA Dashboard and the relevant API (such as Prometheus-API). The Monitoring Engine applies a distributed approach to its metrics collection model by receiving and processing monitoring data/requests related to a particularly deployed Network Application on the overlay mesh network that links the corresponding nodes of the operational Network Application.

In terms of vertical orchestration tools, Kubernetes has been determined to be the optimal solution for orchestrating virtualized applications in the far-edge, edge, and cloud segments, satisfying the On-vehicle MANO and Edge and Cloud MANO requirements specified in the 5G-IANA project.

Kubernetes offers scalability, resilience, and portability that align with the 5G-IANA context, allowing applications to adjust the number of working nodes based on runtime requirements and continue functioning even in the event of network failure. MicroK8s, a lightweight and easy to install Kubernetes distribution certified by Canonical, was selected as the preferred distribution for 5G-IANA due to its ability to support limited hardware capabilities. The Network Application Orchestration & Development layer is responsible for deploying and managing services on the onboard/roadside units, utilizing a Microk8s API abstract interface for basic management operations. Prior to MicroK8s installation, the far-edge device must be configured to support containerized applications, and if a GPU is present, the container engine and MicroK8s must be configured accordingly.

5.3.6 EVOLVED-5G

EVOLVED-5G project [12] is focused on enabling the long-term evolution of 5G-enabled vertical industries by providing the necessary means and tools for this objective. In this regard, it aims to facilitate the realization and utilization of Network Applications that offer advanced 5G Core functionality through standardized APIs such as CAPIF [13] and MEC[4]. To achieve this objective, the project has established an open experimentation facility and developed tools for creating, verifying, validating, and certifying 5G Network Apps, as well as an online Marketplace where those Network Apps can be published and openly offered to potential users. Moreover, EVOLVED-5G project has developed a variety of Network Apps addressed to diverse use cases,

[4] https://www.etsi.org/technologies/multi-access-edge-computing

focused but not restricted to Industry and Fabric of the Future, as Network Apps under the EVOLVED-5G vision could be applied to potentially any sector or application domain.

EVOLVED-5G Network Apps consume 5G Core APIs, leveraging the SBA architecture, and provide services based on 5GC functionality via standardized Service APIs. Most of the Network Apps developed in the EVOLVED-5G project have followed the aaS model. In this regard, services are provided by exposing the offered functionality via an API that external applications can consume to make use of such services, and leverage 5G Core functionality.

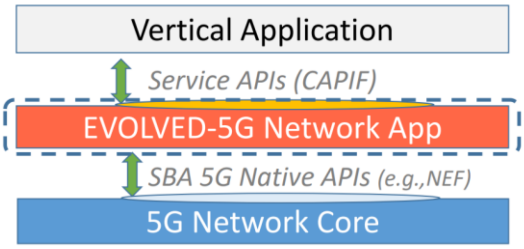


Figure 19: Scheme of EVOLVED-5G Network App aaS implementation

From an implementation perspective, EVOLVED-5G Network Apps provide RESTful Service APIs and are containerized using Docker technology, allowing for an immediate deployment regardless the hosting OS and underlying technologies. Also, as mentioned, those APIs adhere to Common API Framework (CAPIF) 3GPP specifications for 5G Networks to ensure wide adoption and high interoperability.

Table 2: Details of EVOLVED-5G Network Apps following aaS implementation approach

Network App Use Case	Native APIs	Service API
Chatbot assistant (Smart Factories: Interaction of Employees and Machines)	5G Network Core APIs from the 3GPP Network Exposure Function (NEF)	Use-case specific APIs managed by 3GPP Common API Framework (CAPIF)
Anomaly Detection (Factory of the Future operations)		
Industrial grade 5G connectivity (Factory of the Future operations)		
ID Management and Access Control (Factory of the Future operation)		
5G SIEM add on (Factory of the Future operation)		
Teleoperation (Smart Factories-Production Line Infrastructure pillar)		
Localization (Smart Factories-Production Line Infrastructure pillar)		
Smart irrigation for agriculture (Smart Agriculture)		

The EVOLVED-5G facility is based on an architecture that is composed by five separate environments, which are interconnected through a unified CI/CD framework and a shared artefact repository.

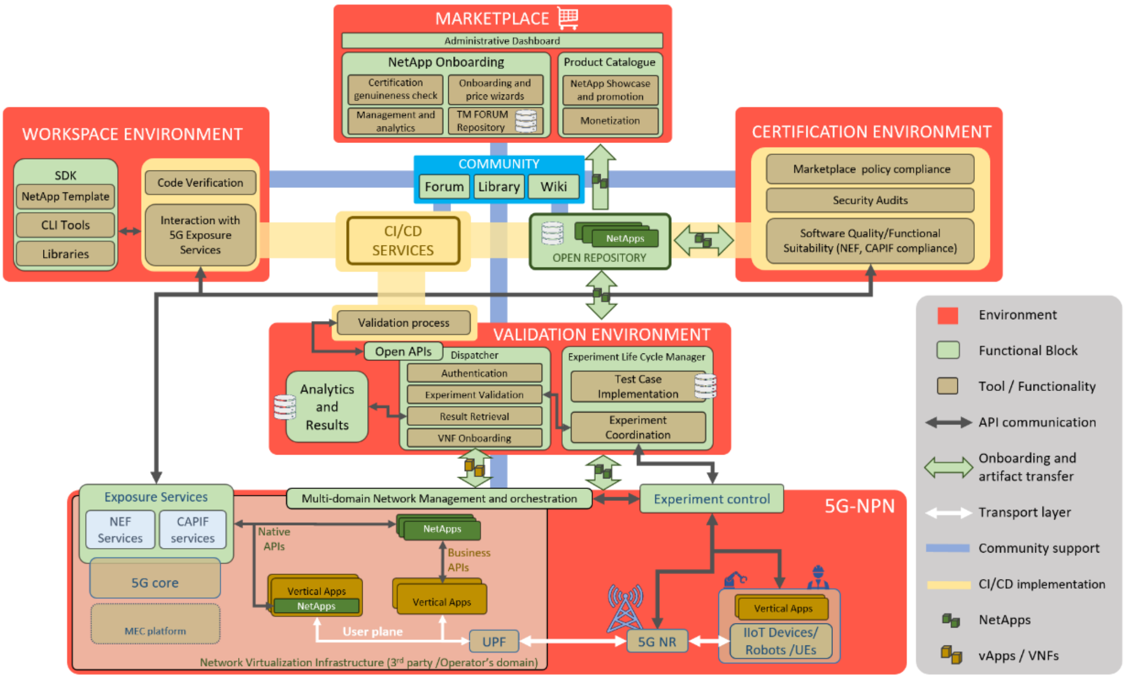


Figure 20: EVOLVED-5G system architecture

The EVOLVED-5G facility components are organized in groups, called environments, that are associated with the phases of the lifecycle of the Network Application: development and verification (Workspace), validation (Validation Environment), certification (Certification environment), and release to the Marketplace. The 5G NPN, in addition, is needed to support both the validation and certification phases of the Network Application lifecycle.

System components are grouped in different levels of abstraction according to the compositional and structural logic of the EVOLVED-5G architecture. From the most abstract to the most concrete, they are as follows:

- **Tier1 - Environments:** They are derived from and give support to the different phases of the lifecycle of the Network Application, or act as the foundation of other environments.
- **Tier 2 - Functional blocks:** They encapsulate related functionalities and define the main building blocks that compose each environment.
- **Tier 3 - Tools and functionalities**: Inside each *environment* and *functional block*, several components support the implementation of the different capabilities. In Figure 1, these are identified either by the name of the component or (in order to improve clarity) by stating the functionality offered.

Following these principles, the system architecture results in being composed by five environments that are interconnected by the usage of (i) **CI/CD services**, (ii) a centralized repository (the **Open Repository**), which encapsulates a source code repository and an artefacts repository, and (iii) an additional block that contains all the services that support the creation of a **Community** around the EVOLVED-5G ecosystem. In Figure 20, the tiered structural composition of the architecture is reflected by distinctively styled boxes (tier 1: red, tier 2: green, tier 3: brown). In terms of the different components' integration, different arrows are used to refer to available APIs usage, and exchange of artefacts and data.

The relation between the architectural components and the different phases of the NetApp lifecycle is illustrated in Figure 21.

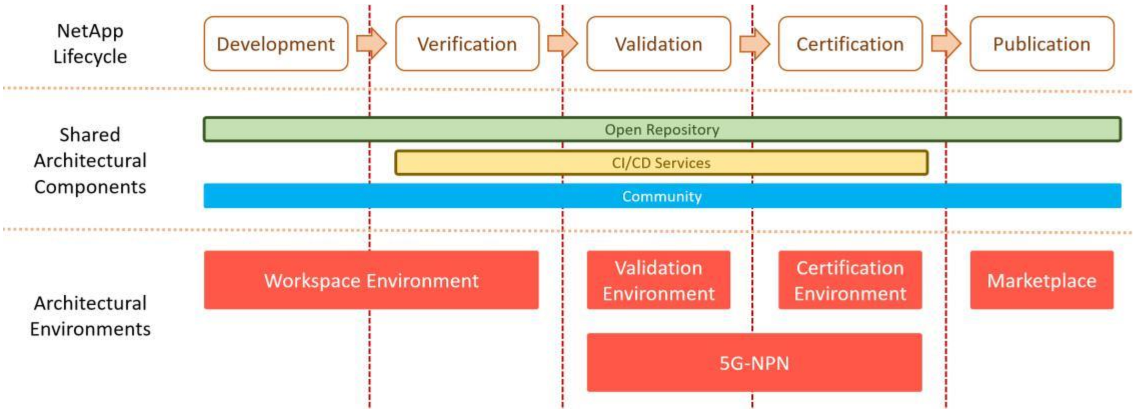


Figure 21: Network App lifecycle phases and relation with EVOLVED-5G architectural components

The **Workspace Environment** supports the development and verification of the Network Application. These two phases refer to:

- The implementation of the Network Application, including the basic functionality and compatibility with the exposure services of the 5G network. For this, the Workspace provides an SDK (Software Development Kit) to Developers, which includes documentation, libraries, pre-defined templates and a CLI (Command Line Interface) tool that ease the creation of Network Applications.
- The verification of the Network Application corresponds to the assessment of the correctness of the Network Application in terms of basic functionality and compatibility with the 5G exposure services. This is implemented as a set of tests, available to Network Application developers in the form of a verification pipeline in the CI/CD services.

Network Application developers can make use of the SDK in their local premises and can have access to the CI/CD services at any time, which allows them to work independently and without any time constraints. Once developers are confident in the functionality of their Network Application, they can proceed to the next phase in the lifecycle of the Network Application: the validation.

The **Validation Environment** aims at providing the means for testing the suitability of the Network Application for working under real network conditions, and the successful integration of the Network Application with a vertical application (vApp). For this reason, the validation is performed within the premises of the validation platforms, which have access to experimental 5G deployments, and as such they can provide a realistic but controlled environment, where KPIs (Key Performance Indicators) can be measured under different network conditions.

In order to perform validation and integration tests for Network Apps in 5G network environments, which means along with an associated Vertical Application that makes use of such a Network App, it is employed an evolved validation methodology from the H2020 5GENESIS project in 5G experimental platforms in Athens, provided by NCSR Demokritos and Cosmote, and in Malaga, provided by University of Malaga and Telefonica [45]. In order to test the provisioning of a hybrid Network App, a methodology has been defined to validate their integration and operation in a 5G network environment [46].

For the validation of a specific Network Application, developers, platform owners and vApp providers agree on a set of tests designed for computing any KPI that is of interest to the involved parties. These customized tests are closely related to the nature of the selected Network Application + vApp pair, and aim at measuring the fitness of the Network Application in supporting their specific use case. These tests are complemented by a set of pre-defined trials (similar to the ones provided as part of the verification process) that assess the correctness of the Network Application in terms of basic functionality, quality of the code and integration with the 5G exposure services.

The automated validation process begins with the developer launching the validation pipeline through the CLI tool that initiates the validation tests. The pipeline starts with an experiment and platform performance assessment. The validation team retrieves the code and performs static code analysis and a vulnerability scan of the Network App. The Network App image is then grabbed from the central open repository for Network Application in the EVOLVED-5G framework and validated using a container image certification tool.

Afterwards, the Network Application is deployed, along with CAPIF services (refer to section 9.2) and an emulated version of the 3GPP NEF that has been developed as open-source project and is publicly available by NCSR "Demokritos" [47], and tests are performed to verify the open ports and onboard the Network App using CAPIF. API discovery and call-back tests, as well as NEF services, are also performed. The Network Application is scaled up and down using ReplicaSet[5], a Kubernetes controller that ensures a specified number of identical pod replicas are running at all times, with the help of HELM [17], a package manager for Kubernetes that simplifies the deployment and management of applications. Subsequently, integration tests are conducted between the Network Application and vApp. Finally, the Network App is destroyed, and the offboarding test is performed.

Not directly related to the Network Application, but of importance in order to guarantee the repeatability and validity of the results, the validation also includes an initial set of platform assessment tests, which verify the performance of the 5G NPN infrastructure before testing the Network Application.

Then, the Validated Network Applications can be considered for the next phase of the Network Application lifecycle, that is the certification phase. In the **Certification Environment** the Network Application becomes a subject of an extensive quality assessment and conformance testing, performed in an automatic way by making use of a dedicated pipeline in the EVOLVED-5G CI/CD services. A certified Network Application is guaranteed to be interoperable with commercial 5G networks and can be released to the market.

The release to the **Marketplace** is the last stage of the Network Application lifecycle. In the Marketplace, a successfully certified Network Application can be made available to end users, who can have access to the artifacts required for the correct deployment of the Network Application in their public or private 5G networks.

The fifth environment of the EVOLVED-5G architecture, the **5G NPN**, supports the validation and certification of the Network Applications by providing the required 5G APIs (5G exposure services), as well as a 5G network where the Network Application can be tested. For this testing to be possible, the 5G NPN integrates the software (such as the Open5Genesis Suite) and hardware (computational resources, storage, measurement equipment, etc.) components, which

[5] https://kubernetes.io/es/docs/concepts/workloads/controllers/replicaset/],

are necessary for the coordination and execution of the actions described in the corresponding test cases.

Apart from the five environments, two components of critical importance act as the connecting glue of the architecture, and thus, underpin all the stages of the Network Application lifecycle. These are the CI/CD Services and the Open Repository.

The **CI/CD Services** perform an important role to all of the intermediate stages of the Network Application lifecycle (i.e., all stages except for development and the publication to the Marketplace) by implementing most of the logic required for the automated actions that constitute the verification, validation and certification processes.

The CI/CD Services implement a set of pipelines which are exposed to the Network Application developers via appropriate commands of the CLI tool of the Workspace. Each of these pipelines automatically execute a set of actions while registering the results (such as logs, measurements or any other information) for the generation of a complete report that is made available to the developer.

The **Open Repository** acts as a central storage component that is accessible to all the other environments in the architecture. In this storage, all artefacts that are related to a Network App, such as its source code, its binary images, relevant documentation or reports generated during verification, validation and certification, are saved and made available for retrieval. The role of the Open Repository is served by two different tools: GitHub acts as a code repository, given its ubiquity and familiarity among developers, while Artifactory is used for the storage of heterogeneous artefacts.

Finally, the **Community** plays an important role in the EVOLVED-5G ecosystem, centralizing the support provided to any external entities in exploiting the EVOLVED-5G framework. The Community comprises the Wiki [48], Forum [49], and the Accelerator Library [50], and a set of online training courses on an educational platform, which are open for registration and freely available.

6 Hybrid Network Applications class

This Network Applications class is related to the model where the vertical instantiates a part of its Vertical Application in the operator domain like the EDGE. The other part remains in the vertical domain.

6.1 Implementations examples

6.1.1 SMART5GRID

This section illustrates the Network Applications implementation model followed in the Smart5Grid project. The project aims to revolutionize the energy vertical industry by incorporating to 5G platforms to their environments. The Smart5Grid platform allows the vertical and third parties to design and deploy Network Applications oriented to support the production and operation environments of power grids. This platform is composed of several components that together allow them to act as a Vertical Service Providers (VSP) (see Figure 22). The components are detailed below:

- **Open Service Repository (OSR):** Allows developers to store and register Network Applications so they can be easily accessed and deployed in 5G infrastructures.
- **Validation and Verification Framework (V&V):** Used to perform automatic certification actions, definition validation and testing of Network Applications.

- **Network Application Controller (NAC):** Component in charge of provisioning and managing the lifecycle of Network Applications in network infrastructures. This component can interact with 5G network providers or directly with resources provisioned by the energy vertical customer.

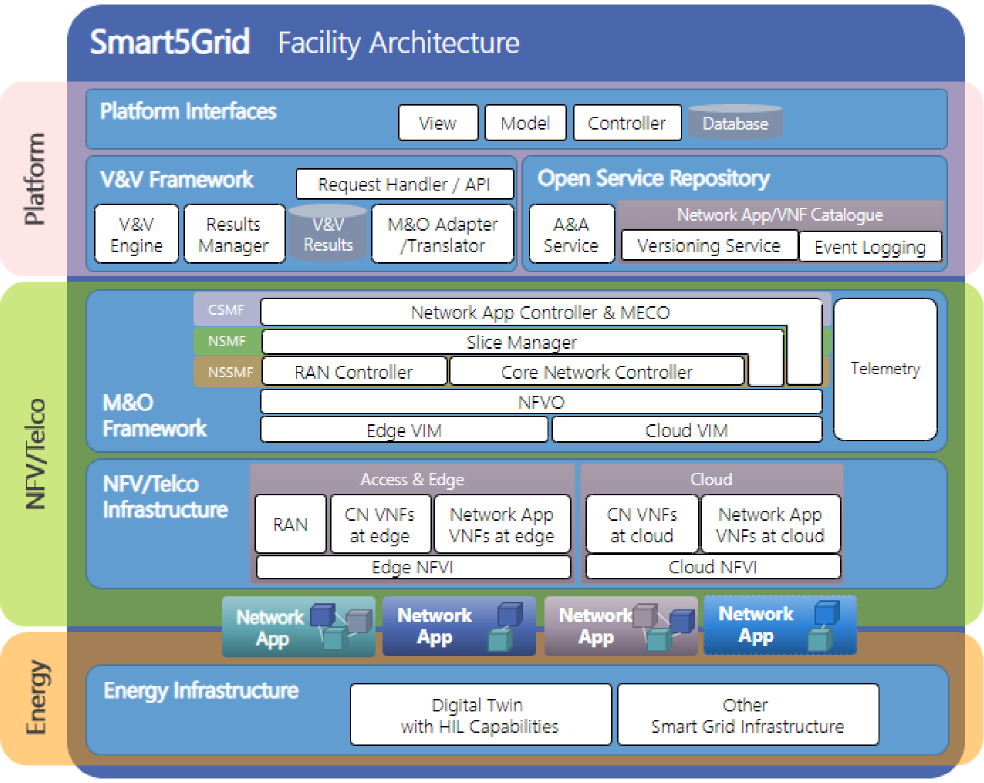


Figure 22: Smart5Grid architectural framework detailed by layers

Smart5Grid Project has defined several use cases, each with particular deployment and operation environments. For this reason, the Network Applications information model defined by the project has the flexibility to support different levels of integration with the Telco infrastructure as defined in section 3, being the Hybrid and Coupled/Delegated models, the ones supported. For example, the first model allows the deployment of Network Applications in a scenario where a VSP is integrated with a communication service provider (CSP). While the second allows Network Applications to be packaged in such a way that they can be deployed, integrated, and managed within the domains of a CSP. Here the Smart5Grid platform being a VSP could take the role of a CSP if it has full access to its 5G network infrastructure, for example, in scenarios where there are 5G private network providers.

In the context of the Smart5Grid project, several use cases have been defined that stablish their own topologies, functionalities, and deployment environments. All of them use the edge computing paradigm for the deployment of Network Applications, but the way they are managed varies according to the model followed (Hybrid or Coupled/Delegated). However, in order to cover the integration between the Smart5Grid platform and the Telco infrastructure layer, the Coupled/Delegated model for the provision of Network Applications is detailed below, which involves the integration of 5G networks and paradigms such as network slicing and Network Applications deployed on edge environments.

Integrating Smart5Grid platform with CSPs

In order to delegate the provision of Network Applications to a CSP, it is important that the VSP adapts its interfaces to the functionalities provided by the CSP. For this purpose, from the

Smart5Grid platform, the NAC component must adapt its southbound interface (SBI) to be able to communicate with the CSP, which in the case of the Smart5Grid project, is a 5G private network provider.

The 5G private provider is internally composed of the infrastructure and management layers. The physical network infrastructure layer consists of radio equipment, routers, and compute servers that host the Network Applications and virtualized 5G network and radio services. While the management layer contains the software components responsible for managing physical and virtual resources, configuring end-to-end network connectivity, and managing the lifecycle of virtualized services and Network Applications. In this context, in order to facilitate complete control of the 5G network infrastructure for vertical clients and end-users, the 5G private network provider enables a global interface (API). This interface permits external components, including end-users, to interact with the 5G system.

The Smart5Grid platform, specifically the NAC component, makes use of the interface provided by the 5G private network provider to delegate the provisioning of Network Applications (see Figure 23). For this purpose, the Smart5Grid platform must perform some previous steps:

1. Developers must generate the Network Application packages with artifacts and virtualization tools that allow the applications to be deployed in NFV infrastructures (NFVI). The packages are composed of: images containing the Network Application logic and software (images for containers or virtual machines), Network Application descriptors created in the context of the Smart5Grid project, and NSD (network service descriptor) and VNFD (VNF descriptor) descriptors based on the ETSI NFV standard.
2. The packages and repositories of the Network Applications images are placed in the OSR component where a catalog of the VNFs and Network Applications is maintained.
3. To validate the Network Application, the Network Application descriptors are reviewed by the V&V component which gives first feedback to the user if the descriptors are well defined. Subsequently, the descriptors are sent to the NAC component to initiate their onboarding and deployment operations in the Telco infrastructure by interacting with the 5G private network provider's API.

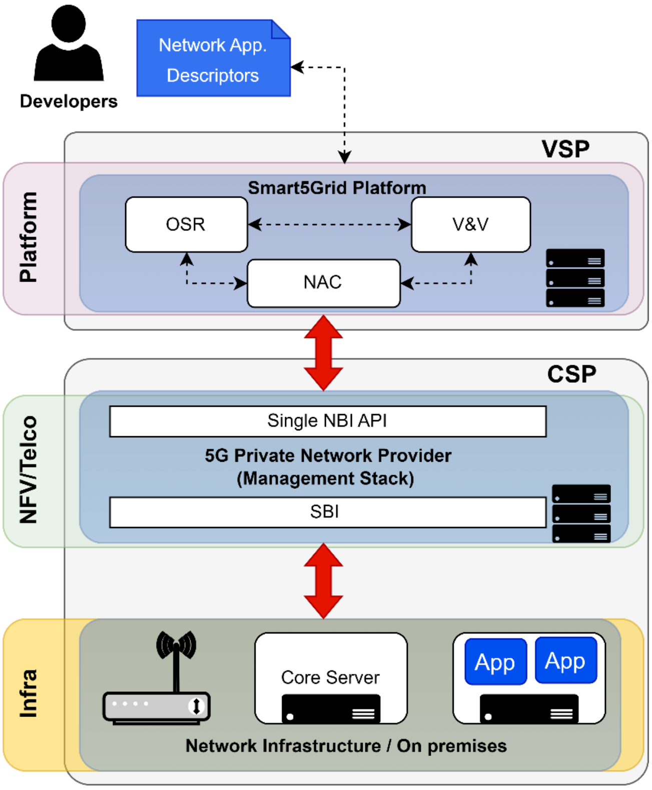


Figure 23: Smart5Grid platform and 5G Private Network Provider integration

However, the integration of a Network Application with a 5G network is not a trivial process; several previous configurations must be made at the 5G network core level to establish connectivity between UEs and Network Applications. In addition, many of the applications may require heterogeneous service level agreements (SLAs) which implies that network slices with different performance characteristics must be configured to meet the service level objectives (SLOs) of the applications. The Network Application Descriptor, as defined in the Smart5Grid project, is solely concerned with defining the Network Applications information (e.g., name, repos, endpoints, etc), their service level objectives (SLOs), and the interconnection between their services, in case they are comprised of multiple VNFs. However, it does not provide any information related to the creation of network slices or the configuration definition within the 5G network core.

These functionalities, such as network slicing and 5G network configuration allowed by the 5G private network provider are executed by the provider itself before the deployment of the applications. Finally, to test the provisioning of a Network Application, a methodology has been defined to validate their integration and operation in a 5G network environment.

Validation Methodology:

In order to perform validation and integration tests of Network Applications in 5G network environments, it is important to take into account some prerequisites and/or aspects that must be considered beforehand. For example:

- Definition of minimum computational resources required by the application. This includes the use of specialized hardware such as GPU, TPU, etc.
- Type of connectivity required. In other words, application developers must define applications following the communication model and protocols supported by 5G networks.
- In order to enable developers to debug applications without requiring access to the servers where the applications are deployed, the applications must be defined to expose their service configuration interfaces to the external network, of the applications network. By doing so, log traces can be obtained, and debugging activities can be carried out with ease.

The Smart5Grid project has defined a validation methodology for Network Application pilots, which consists of four stages:

1. **Phase 1: Create realistic conditions to replicate the operational environment of the Network Application**
   At this point the application developers, depending on their functionality, may make use of simulated data that have been created using simulators that allow the creation of digital twins that recreate the expected behavior of the scenario.

2. **Phase 2: Network Application Integration**

   Here the Network Application is prepared and packaged with all the compatible artifacts (e.g., descriptors) with the Smart5Grid platform so that it can be loaded, deployed and delegated to the CSP through the interaction of the Smart5Grid platform components and the CSP platform.

3. **Phase 3: Definition of Test Scenarios**

This stage considers the KPIs that the 5G network is expected to meet to guarantee the Network Application functionality. With the aim of ensuring the preparedness of the system for high demand scenarios, test cases are meticulously defined. These test cases are designed to cover a range of scenarios, beginning with simple end-to-end connectivity tests and progressing to basic functionality tests. Ultimately, exhaustive tests involving data traffic and computational consumption are conducted to ensure system readiness for high demand scenarios. For each of these tests, metrics are monitored and taken to generate a detailed view of the performance of the 5G network and the Network Application.

4. **Phase 4: Validation and Evaluation**
   Basically, what is done in this phase is to compare the KPIs expected by the Network Application with those obtained during the testing phase. This allows developers of Network Applications to make adjustments to the functionalities of the applications and also allows to detect the real performance of the 5G network provided by the CSP (e.g., 5G private network providers).
   .

6.1.2 The 5G-EPICENTRE Approach & Platform architecture

5G-EPICENTRE represents a 5G-PPP initiative with explicit focus on the Public Protection and Disaster Relief (PPDR) sector. The project focuses on developing a platform for PPDR 5G-enabled experiments on top of mature 5G testbed infrastructures joined in federation, where innovators addressing this vertical can stress-test their solutions and verify their robustness in simulated extreme network conditions. Such experiments are targeted at demonstrating PPDR functions over 5G networks.

5G promises to provide the vertical with the required low latency and high bandwidth, in order for PPDR agents to deliver critical life-saving functions and make decisions faster. However, special considerations are warranted, as the size of the data to be transmitted can become massive, and (particularly during catastrophic incidents) network capabilities can be stressed to the extreme. Hence, it is crucial to ensure that every data stream in an identified PPDR mobile service can be transmitted over the 5G network, within a very short time and with guaranteed quality.

To deliver on these requirements, the 5G-EPICENTRE partners have developed the concept of Network Applications with specific focus on PPDR. Eight use cases (UCs) have been defined, aimed at demonstrating how the proposed delivery model for Network Applications allows exposure of advanced 5G service operations, usability of 5G capabilities and adjusting of networks, for PPDR-specific needs. Hence in this Section, we present the 5G-EPICENTRE Network Application approach, alongside the project platform architecture to support the Network Application ecosystem hybrid delivery model.

6.1.2.1 Network Applications: Overview in 5G-EPICENTRE

Within 5G-EPICENTRE, PPDR vertical system developers are expected to utilize the offered experimentation platform to evaluate how an end-to-end (E2E) solution behaves over 5G (e.g., under stressful traffic conditions). As such, an experiment entails deployment of an E2E 5G **vertical system**, one that is a complete, global (e.g., integrating client-server parts) application to be experimented with. Insight is extracted from the experiment to demonstrate what 5G can do for PPDR end users (i.e., police forces, ambulance services, firefighting units, search and rescue, command & control, etc.).

The vertical system under test has a **vertical application** component (e.g., a front-end client on a smartphone), that is deployed in the vertical domain, and one or more **Network Application** components, which are the parts of the system that are delegated to the telco (testbed owner), and abstract the 5G network resources below in simple API calls in a way that simplifies the interaction of the vertical application with the network control plane functions. The concept is elaborated in Figure 24. Essentially, in 5G-EPICENTRE, vertical applications can be developed by any entity (Consortium partner or third party), and can connect via service APIs to our (open) Network Applications to compose a vendor-specific E2E vertical 5G system for that vendor to experiment with, on top of the provisioned platform. Eight such systems (project UCs) are being experimented with in the context of the project, to deliver a proof of concept for the project.

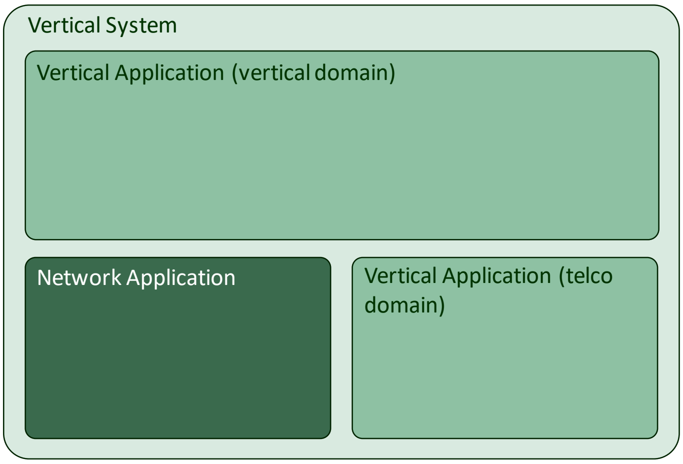

Vertical System
Vertical Application (vertical domain)
Network Application
Vertical Application (telco domain)

Figure 24: Terminology regarding Network Applications

6.1.2.2 Methodology

5G-EPICENTRE Network Applications can act on the control plane functions of the 5GC either as or via Application Functions (AFs), which *de facto* are the 'projection' of a PPDR service onto the 5GC's control plane. More precisely, the PPDR service forwards via the vertical AF a request to the 5GC directly through the Network Application service API (e.g., for a specific QoS for its media flows), which handles the complexity of the interaction with the services exposed by the 5GC (e.g., the Network Exposure Function). The request is then propagated to all other 5G network elements. Through this interaction scheme, benefits that the 5G network can bring to the PPDR sector can be exposed simply in a secure manner.

In this approach, we distinguish between two levels of trust scenarios:

- The request originates at a Network Application, that is **trusted** (e.g., it is part of the 5G-EPICENTRE offering of Network Applications, or has been delegated to the testbed owner, operating on a mutual agreement of trust). In this scenario, the Network Application has been configured to directly communicating with the 5GC functions, bypassing the NEF and making direct usage of the services of other NFs, typically those exposed by the Policy Control Function (PCF).
- The request originates at a vertical application component, that is **untrusted**. In this scenario, the Network Application has been configured to communicate with the 5GC functions, via the NEF, as foreseen by 3GPP standards for untrusted AFs.

6.1.2.3 Implementation

Since Network Applications are essentially chains of network functions, services and platform components, we opted from the very beginning to develop both platform and Network Applications exposed by the project under the microservices architectural style.

This means that we worked towards integrating Kubernetes (K8s) based management and orchestration capabilities in all of the testbeds that federate under the project. The benefits of doing so involved the promise of fast service creation. One of the core performance-related KPIs defined by the 5G-PPP, calls for average service creation time cycles to be drastically reduced ensuring to instantiate within some minutes a complete communication platform, especially for crisis management. Further, such architectures offer increased resilience and scalability: Cloud native core optimally schedules CNF/VNF to run on the available infrastructure; scale in and out in on demand; and utilize container-level isolation and health monitoring, so as to rapidly restart instances in case of failure. If a service fails, the infrastructure will immediately and automatically create a new instance of the service, so that the critical mission can be achieved, without any intervention from the user or the operator.

An interesting side effect of this is the capacity to join our testbeds in federation following multi-cluster K8s management across geographical locations using the Karmada solution. Karmada allows us to treat a multi-cluster environment as if it were a single-cluster environment, and its control plane mimics that of a K8s cluster, in that it offers an API Server, scheduler and various controllers to manage cross-cluster deployments. A K8s infrastructure is therefore deployed as an overlay on top of the NFVI in each testbed. A Karmada cluster agent can manage it from a central location. Then, the NFV-MANO underlay in each testbed can act as the resource orchestrator, allocating computational resources at K8s clusters based on Karmada policies.

6.1.2.4 Experimentation Platform Architecture

In Figure 25 we overview the project experimentation platform architecture.

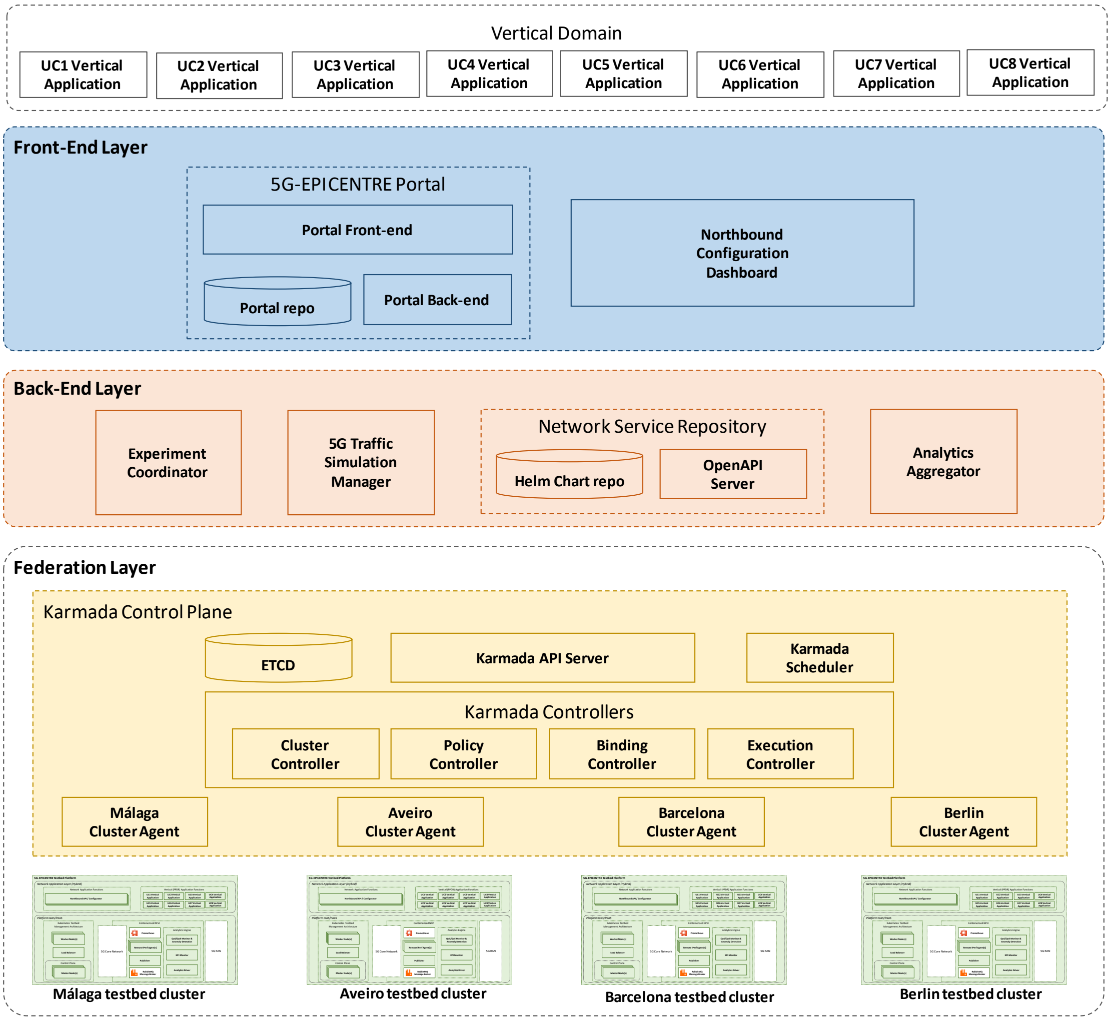


Figure 25: 5G-EPICENTRE platform overall functional architecture component diagram

A front-end layer is developed for interactions between the vertical and the testbed operators, most crucially, to delegate vertical application components (as Helm charts) to accommodate the hybrid Network Application option for interaction. In addition, through the front-end, the experiment definition and execution can be requested.

A centralized back-end infrastructure is then used to both store the Helm chart packages containing the components needed to deploy the vertical application in accordance to the experiment. A Coordinator component takes the Helm chart of the application and installs all the yaml files in the k8s cluster or clusters through Karmada.

Inside the federation, in Figure 26 we can see the augmented infrastructure reference frame. Each testbed has its own individual 5G standalone system configuration, with integrated support for a K8s management environment. Each testbed is comprised of its own NFV ecosystem, including the specific implementations of the 5G architecture (the 5GC and 5G RAN), alongside an augmented Network Functions Virtualization Infrastructure (NFVI), that integrates all virtualized components needed for a testbed virtualized infrastructure to deliver on the 5G-EPICENTRE experimentation requirements (for instance a thorough analytics pipeline). A testbed may opt to support high-availability K8s clusters (*i.e.*, multi-master setup), so as to increase robustness of the architecture in case a master node should fail.

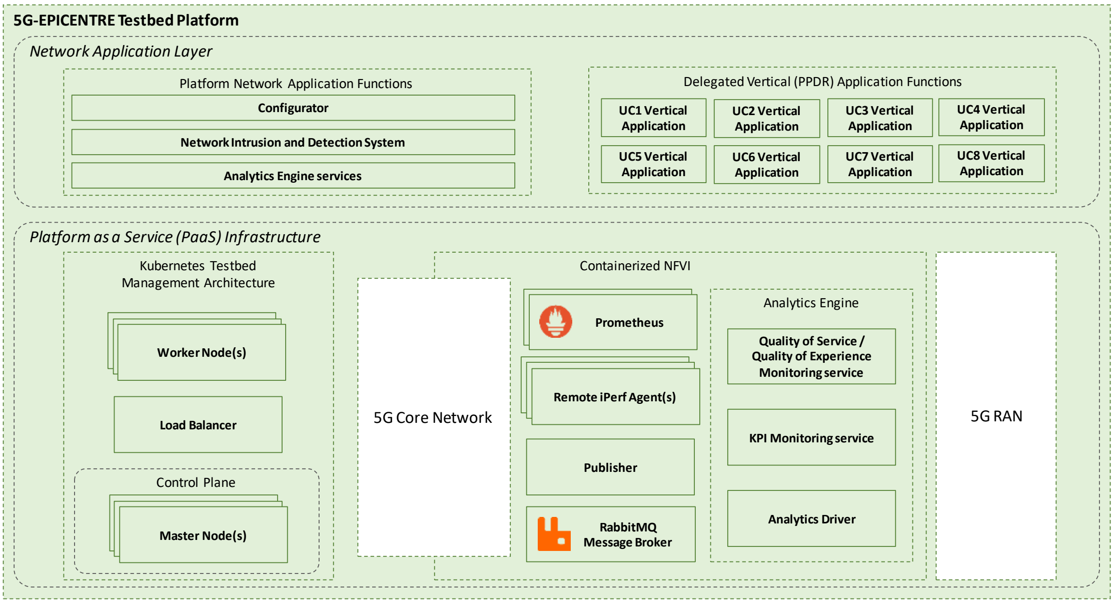


Figure 26: 5G-EPICENTRE PaaS and Network Applications Layers (delegated at testbeds)

From the Network Applications' ecosystem perspective, the aforementioned developments correspond to the PaaS layer in the previous figure. In line with the definition of Network Applications as essentially a middleware layer on top of the PaaS, the testbed reference framework is further elaborated to depict the various Network Applications contemplated in the context of the project (representing the different levels of trust and openness to third parties for the purposes of experimentation).

Network Applications are treated themselves as PaaS components in terms of deployment, since 5G-EPICENTRE treats, orchestrates, and manages Network Applications as part of the testbed K8s cluster.

Network Applications in 5G-EPICENTRE interact with the control plane functions of the 5GC network and expose features that are relevant for vertical applications, such as location or QoS according to the level of trust. To illustrate these varying levels of trust, we make a distinction between *platform Network Applications* and *vertical Network Applications*

Platform Network Applications are all the 5G-EPICENTRE functional components that are defined as Network Applications. They are considered agnostic to the vertical application and (being internal project implementations) operate under the **trusted** level of interaction scenario.

Vertical Network Applications correspond to the parts of the vertical application which is delegated to the telco/testbed operator under the ‘Delegated’, or ‘Hybrid’ options of interaction using the processes available to experimenters and function developers at the 5G-EPICENTRE Portal. Therefore, they operate under the **untrusted** level of interaction scenario

For illustrative purposes, the block incorporates vertical application components delegated by the UC owners for each of the eight foreseen UC vertical applications to be experimented with.

7 Security

The challenges posed by present-day Information and Technology (IT) markets demand the implementation of distributed and composable systems from organisations, which are being moved outside of their physical boundaries. To that extent, the cloud is contributing to the simplification of their operations and removing much of the burden efforts involved in managing and deploying traditional server infrastructure. Moreover, organizations are leveraging automation capabilities from software-driven infrastructure models, which has resulted in a cloud native approach.

The focus on a cloud-native approach for applications has led to an increased attack surface that requires the adoption of security measures throughout the software development lifecycle, from the moment the applications are designed, until the moment they are deployed and operated in production environments. Such an approach is referred to as Security-By-Design (SBD) and is one of the many security concepts considered while developing any application.

The objective of this section is to give some insights on the security and its crucial role in the Network Applications design. After summarizing some recommendations from ETSI, we introduce two innovative frameworks proposed by 5G-EPICENTRE and EVOLVED-5G. Of course, the logic of the developed frameworks could be easily applied in many other use-cases developed in ICT-41.

7.1 VNF Package Security Specification

In relation to the deployment of VNF packages, ETSI released a document where the VNF Package security requirements and procedures are defined [32]. The document addresses the security issues related to the integrity, authenticity and confidentiality of the VNF Package artifacts. During the onboarding process, it is crucial to check the VNF package security for the successful deployment. In this way, ETSI explores in this document security solutions to enhance the privacy in this process. However, only a high-level perspective is provided, without technical details or implementation guidelines.

To assure integrity, the ETSI proposes to add a cryptographic signature to each item included in the VNF Package (VNFD, images, scripts, etc.). The NFVO will be then able to verify the signatures and sign the whole VNF package when storing it, if everything is correct. The items will be signed by their creator, the VNF provider. Following this procedure, the VNF package integrity is ensured during the onboarding and the authenticity and integrity of the package during the VNF instantiation can be granted.

The proposal also explores confidentiality capabilities to store the VNFs in the corresponding catalogues. Once the integrity of the VNF package has been verified following the steps described above, the NFVO will encrypt the VNF artifacts using encryption keys supplied by the service provider. The encrypted methods are not described in the ETSI document. Once encrypted, the packages will be securely stored in the catalogue. Following this procedure, the VNF packages will acquire confidentiality protection during the instantiation. The VNF package will be decrypted just before the instantiation and then, the signatures will be verified.

7.2 Design rules and best practices

7.2.1 5G-EPICENTRE Cloud-Native Security approach

Security-wise, the Network Applications ecosystem necessitates a set of guidelines for software engineers to approach security concerns, and ensure the ecosystem remains intact in case of outside tampering. Within 5G-EPICENTRE, security is a major topic given the addressed vertical (PPDR), which becomes only more important due to the larger attack surface created as a result of adopting a cloud-native deployment option. Hence, a Holistic Security and Privacy Framework (HSPF) was designed. Through the framework, aspects such as network and container-level isolation strategies, the usage of the Service Mesh design patterns, and AI/ML techniques are encapsulated into three main abstract components: a security engine, a policy engine, and an AI engine, which combined aim to secure the overall infrastructure and monitor, mitigate and respond to security incidents (Figure 27).

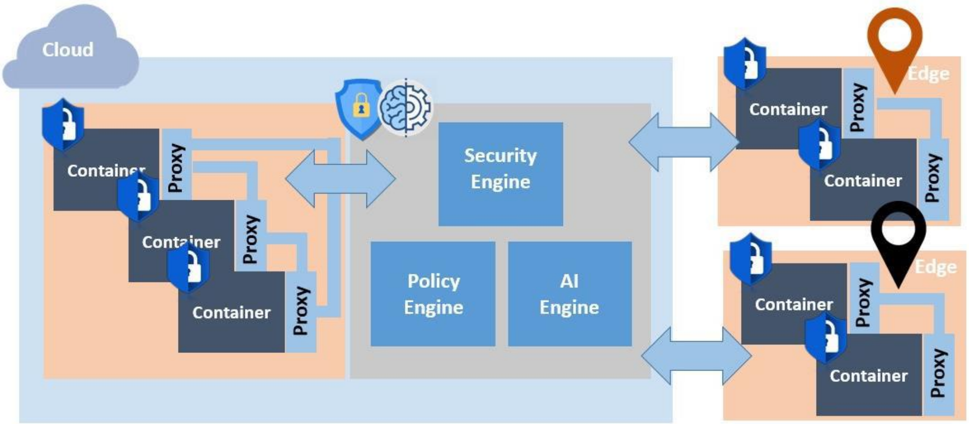


Figure 27: Reference architecture of the Holistic Security and Privacy Framework

Based on this framework, two concrete security considerations are proposed: one where concrete specification for cloud-native 5G Network Application ecosystems are addressed, and another where security functions are themselves embedded within a Network Application:

- In the former case, future Network Application ecosystem implementations based on the chaining of container-based NFs should support integration with a **service mesh** implementation, which will allow management and observability of API traffic among network services. The service mesh implementation is proposed at the vertical system level (see Section 6.1.2) and implicates the deployment of sidecars (network proxies in separate containers), that transparently integrate into a Kubernetes cluster. The sidecars do not introduce functionality to the vertical system under test, but rather enable the monitoring of incoming and outgoing traffic, thus capturing all the traffic for analysis. That is considered the first step towards the detection of malicious attacks.
- A Network Intrusion and Detection Network Application, which embeds security in the context of a Network Application component designed to agnostically integrate with vertical systems, deploy their sidecars (traffic collection agents, one for each vertical system microservice), and identify and respond to security threats (e.g., Port Scan, or Denial of Service Attack). This is achieved by means of an *Analytics, Intelligence, Control and Orchestration (AICO)* component, a concrete implementation of the HSPF that triggers security at the policy application level, e.g., blocking the origin of the anomalous traffic communication classified as an attack. The Network Application architecture derived from the HSPF is depicted in Figure 28.

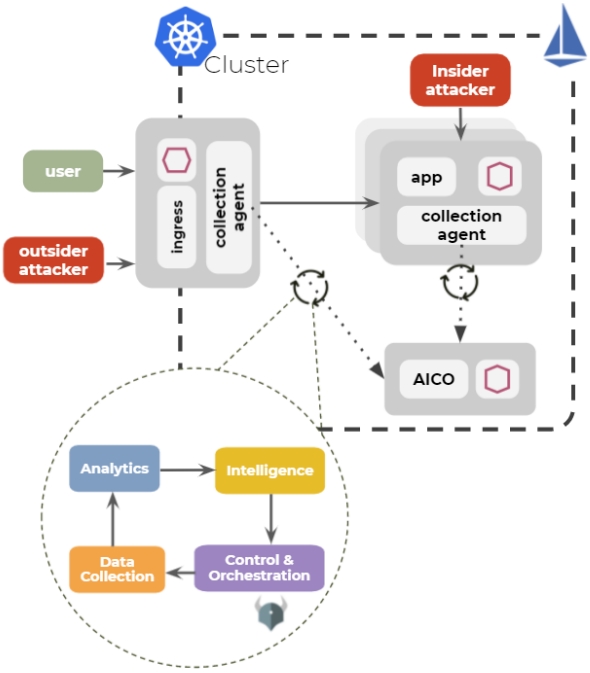


Figure 28: Network Intrusion and Detection Network Application implementation architecture

7.2.2 VITAL-5G security implementation

One of the 5G objectives is to deliver security for the envisioned Platforms and testbeds, as security is a general concern, treated in the VITAL-5G project [14] not only by following the 3GPP 5G SA WG3 security architecture and rules but also by extending and implementing the security policy is applied in general to the 5G vertical's and potential use cases, and in particular to 5G testbeds infrastructure, but also to the Network Application and 3rd party applications.

In 3GPP by design the Core Network component for Control Plane are protected by TLS (Transport Layer Security), the Air Interface is encrypted (and integrity protected) between the device and the gNB, as encrypted domains, the transport network and the services Data Networks being protected by using a suite of security methods and tools [15].

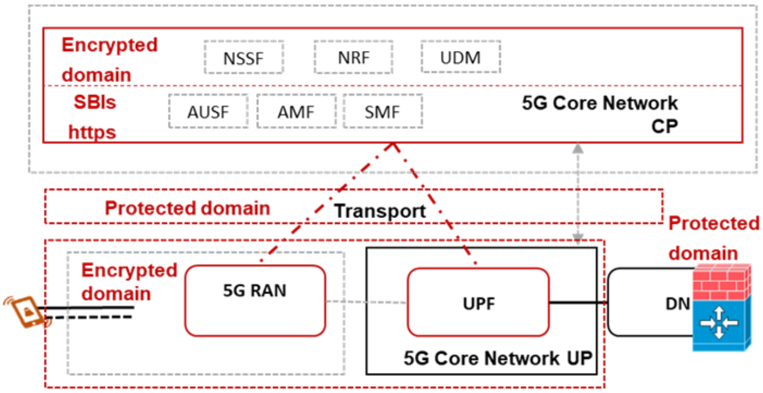


Figure 29: 5G network security landscape

VITAL-5G defines several layers of security, adapted to a multi-level security architecture, based on 3GPP security design aspects, for Security and Privacy, SA3 approach, as seen Figure 30:

- Network Access Security; Network Domain Security

- User Domain Security; Application Domain Security; SBA Domain Security
- Visibility and configurability of security

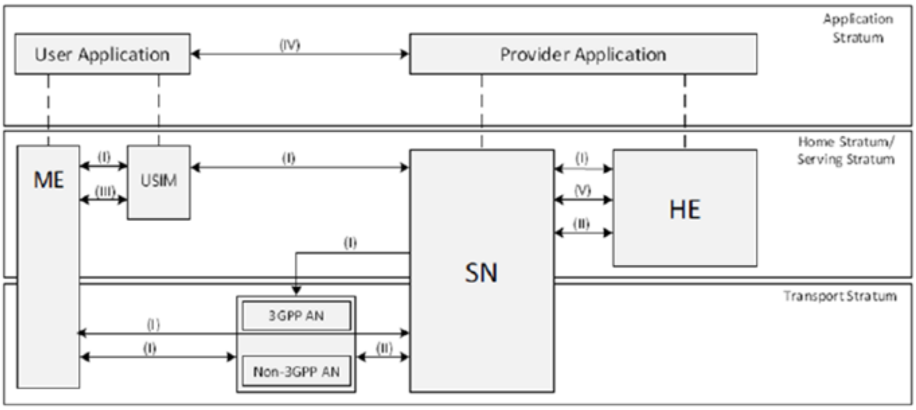


Figure 30: 5G SA3 security design, overview of security architecture

VITAL-5G infrastructures contains not only the 5G components, but also the VITAL-5G platform, exposed to (1) partners, (2) developers and further to the (3) other 3rd party experimenters. The 5G security scenarios based on the EU risk report for the 5G networks are also evaluated, analysing the different aspects as confidentiality, availability and integrity, including, but not limited to security accidental scenarios, individual hacker or group, state-backed door actor or inside a telco/5G testbed operator.

The security implementation is split in two major parts: (1) 5G network security based on 3GPP and ENISA cases and (2) VITAL-5G security implementation related to secured access to the platform's and testbeds components (e.g. developer and 3rd parties' access in the system, protection of some exposed interfaces).

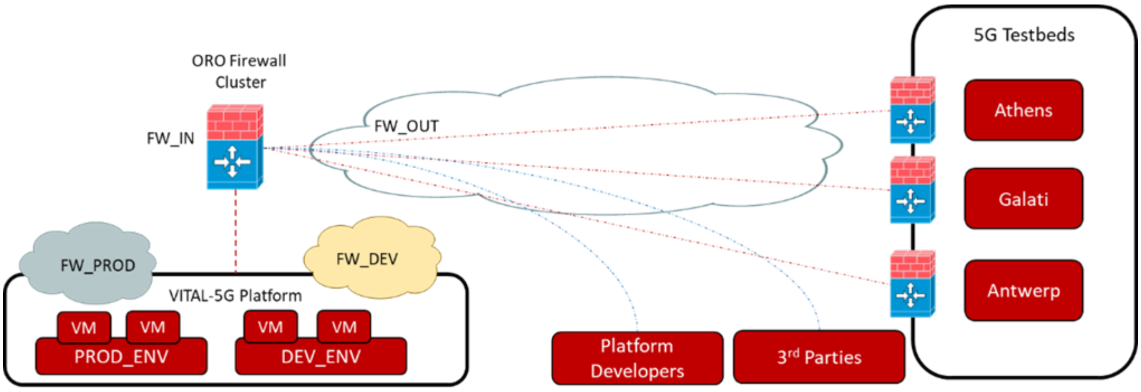


Figure 31: VITAL-5G testbeds connectivity to the platform components

The testbeds are following the 3GPP security standards and implementation, in order to avoid or mitigate as much as possible the 5G network security issues. As described by [15]we are implementing the full set of security features and the security mechanisms of the 5G system following the security procedures performed within the 5G System (implementation by design).

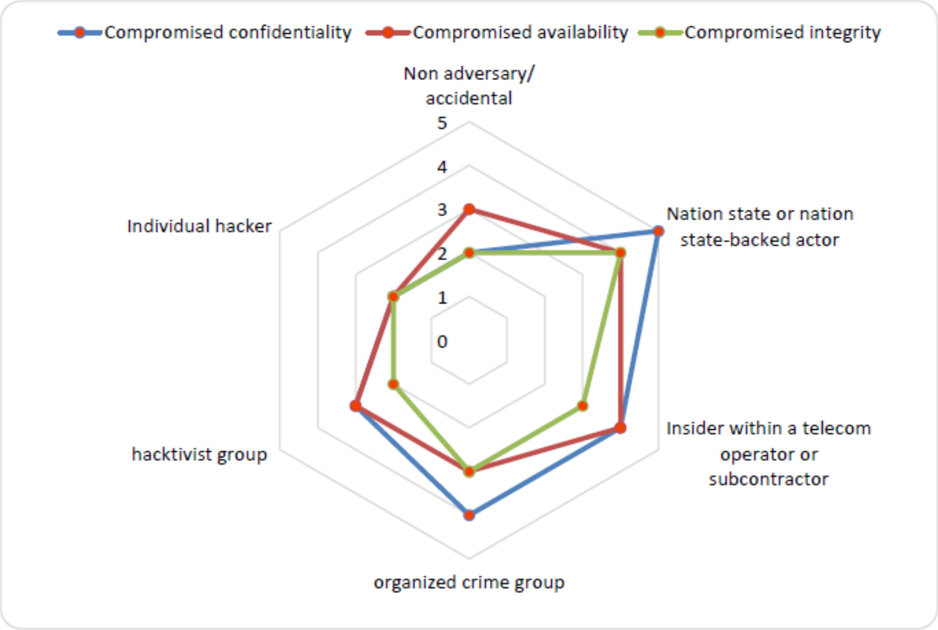


Figure 32: EU coordinated risk Threats, assets and vulnerabilities[6]

VITAL-5G security framework envision analysis and mitigation actions for the next components and functions:

- Physical Infrastructure (hardware equipment)
    - Physical hardware resources, computing, storage, network routers and switches.
- Virtualized Infrastructure (IaaS/CaaS)
    - Management systems, Hypervisor vulnerabilities, management interfaces/APIs; Northbound APIs interfaces, VNFs/VMs or CNFs attacks
- 5G Network Functions (mainly the 5G RAN and Core)
    - Network configuration, exploitation of configured data, malicious functions.
- Authentication and authorization
    - Denial of services, essential core network access (e.g., UDM)
- User data and signalling confidentiality and integrity
    - Malicious functions to exploit signalling part
- Secure storage and processing of subscription credentials
    - Subscribers privacy and data confidentiality

In VITAL-5G there were identified several few security issues, for the (1) Platform Developers, (2) Testbed owners in relation with the VITAL-5G platform, (3) Network Application developers and 3rd party experimenters that could run experiments on different VITAL-5G facilities and testbeds. There have been defined the secured flows and isolation through routing instances for proper communication access to the different components, using IPSec VPNs Site2Site model or VPNs through SSL-VPNs model, as in Figure 33.

[6] https://ec.europa.eu/commission/presscorner/detail/en/IP_19_6049

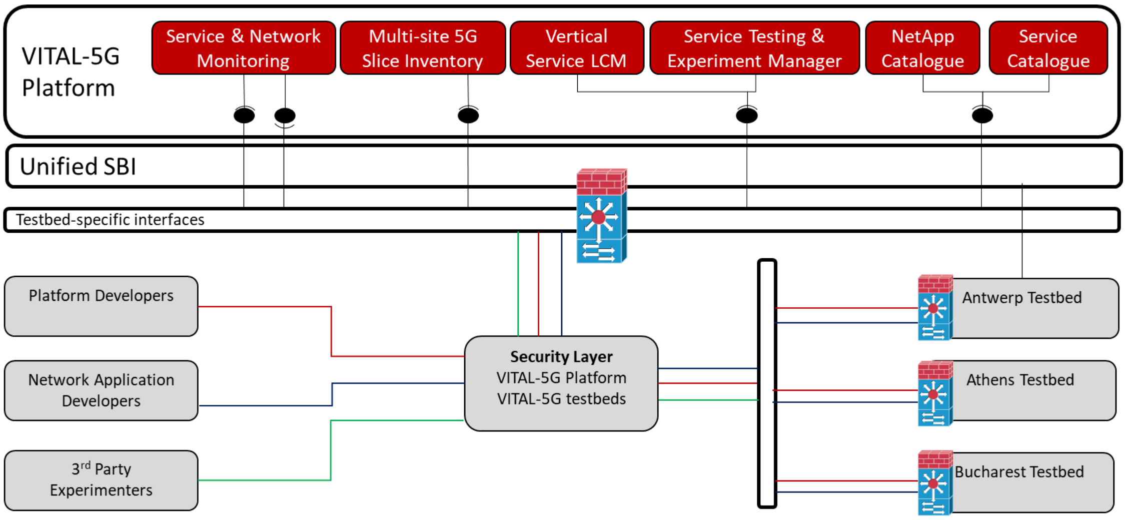


Figure 33: VITAL-5G security framework implementation

In VITAL-5G it has been created a security by design approach, providing communication flows between different entities in a controlled environment, strictly permitting only the access to the requested resources, on dedicated ports and to specific Sockets. Based on different security scenarios, all these accesses are permitted only using secured pre-shared keys for authentication, no others external entity being able to access any resources. As the Security Gateway for SSL or VPN are exposed in Internet, the Gateway and the system are protected by security and cyber security systems from different attacks, as for example the most common attacks, as DDOS, Malware, Phishing, Unauthorized access, Man in the Middle.

The VITAL-5G security access rules permit the following flows detailed in Figure 33, as:

- Platform software developers to the VITAL-5G Platform VMs (RED flow),
- Network Application developers to the VITAL-5G Platform VMs and 5G testbeds (Blue Flow),
- 3rd Party Experimenters to VITAL-5G Platform Components (Green Flow)

VITAL-5G Platform is hosting the software components in different VMs, accessed by the platform developers and implemented in the Virtualized environment, the security actions applied, providing secured services for Network Application, Services & Experiments Catalogue, Portal Web GUI, Blueprint Validation Tool, Slice Inventory.

The architecture emulates and respects a fully functional next-generation service provider and Data center networks, as this target architecture aims to solve a lot of the current technologies scalability, security and design issues that are commonly found in service provider and Data centers – among which the most important are the following: Layers scalability, Recovery time in case of failures, redundant firewall cluster for secure remote access and network security segmentation and servers running virtualization technologies. VITAL-5G implemented security domains for each of these cases, considering the public interfaces exposed to the Internet that could be a potential security or cybersecurity breach. The scenario when a remote testbed could have a potential security issue or prominent to an attack that may impact the testbed has been also evaluated. Based on the 5G security implementation, it has been avoided any potential vulnerabilities that can be faced due to illegitimate platform access from the other testbeds, based on security filtering, cryptographic algorithms, authentication procedure, monitoring and isolation within dedicated tenants with controlled limited access. There is not permitted any direct connectivity between the testbeds, but only between testbeds and VITAL-5G platform (IPSec VPNs), developers to testbeds or Platform (SSL VPNs).

8 The marketplace for Network Applications

In this competitive landscape, it has been repeatedly demonstrated throughout recent history that the mere development of high-quality technical products and technologies is insufficient for their success. To ensure the adoption of these technologies and boost their popularity, it is essential to make potential users aware of the existence of the new technical developments, and to facilitate access to these artefacts, promoting their commercialization. This situation is not different for Network Applications, which can be significantly beneficial to industry verticals by allowing the exploitation of 5G Core functionality in different services and applications. Potential stakeholders must be aware of its existence and must be able to access available Network Applications developments to make possible their adoption.

A marketplace fullfils the critical role of connecting developers of products with potential users, facilitating the commercialization process. In the context of technical innovations, a marketplace can play a vital role in bridging the gap between those who develop new technologies and those who stand to benefit from them. By providing a platform for marketing, sales, and distribution, a marketplace can help to increase the visibility of such technical products.

As a platform or a venue where buyers and sellers come together to exchange goods, services, or information, a marketplace offers a centralized location for transactions, allowing for greater efficiency and convenience in the buying and selling process. In a marketplace, buyers and sellers typically interact directly with each other, may negotiate prices and terms, and complete transactions. In the digital era, online marketplaces proliferate, allowing virtually to any person in any location with an internet-enabled device to be able to access its services and benefit from the instantaneous availability of products. These online stores often provide a significant level of trust and security for both parties, through various features such as user ratings, dispute resolution systems, and secure payment methods. Some examples of well-known online marketplaces are Amazon, eBay, Etsy or Android PlayStore, where mobile apps can be easily searched, purchased or freely acquired, and downloaded.

Various marketplaces have been developed by specific ICT-41 projects to showcase the different Network Applications developed on each of them. Furthermore, the idea of creating a global project-agnostic Marketplace for Network Applications has been proposed, which would offer a global set of network apps from the entire ICT-41 initiative.

Among the different ICT-41 marketplaces developed, the EVOLVED-5G [12] marketplace outstands as an example of a fully operative marketplace, able to support transactional online operations, and with a current full chart of diverse Network Applications for verticals for different FoF pillars.

Another venture within the same initiative is the 5GASP marketplace [43] , operating as a showcase portal of registered Network Applications offering innovative solutions for businesses. The Network Applications that successfully underwent the 5GASP autonomous testing processes, along with best practices for their implementation, are made available in the marketplace as “5GASP - Certified”.

8.1 Network Application ecosystem

In this section, we refer to 5G-IANA example to present the Network Applications ecosystem. Different roles have been identified along with their interactions creating the reference model that is presented in Figure 34.

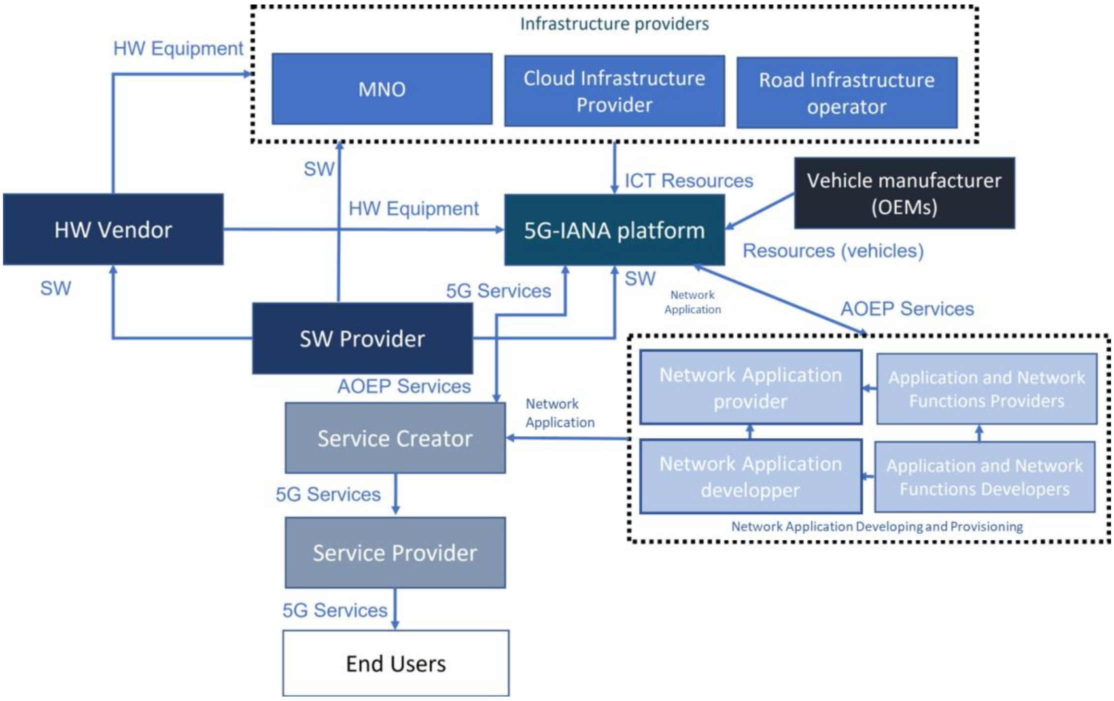

Infrastructure providers
HW Equipment
MNO
Cloud Infrastructure Provider
Road Infrastructure operator
SW
ICT Resources
Vehicle manufacturer (OEMs)
HW Vendor
HW Equipment
5G-IANA platform
Resources (vehicles)
SW
5G Services
SW
Network Application
AOEP Services
SW Provider
AOEP Services
Network Application
Network Application provider
Application and Network Functions Providers
Service Creator
Network Application developper
Application and Network Functions Developers
5G Services
Network Application Developing and Provisioning
Service Provider
5G Services
End Users

Figure 34: 5G-IANA ecosystem

In the reference model, the arrows indicate the relation among the different roles while the direction of the arrow represents the direction of the service flow. The flow of the revenue is on the opposite direction of the arrow, although in some cases there is revenue sharing agreement (in that case there are bidirectional flows and arrows). The rectangular boxes with solid line represent roles, while the ones with dotted lines grouping of roles. We have to note that there are also other relationships between the roles (for example HW vendors provide equipment to all other members of the ecosystem) but these relationships are not examined since they are not affecting the business model of the AOEP (Automotive Open Experimental Platform). More information is provided in [44].

8.2 Marketplace technical development

8.2.1 Marketplace FrontEnd

8.2.1.1 TMForum APIs

Tele Management Forum (TM Forum) is a global industry confederation actively working on evolving current Operations/ Business Support Systems (OSS/BSS), seeking solutions that facilitate their consumption by verticals and their integration into existing standards-driven architectural frameworks. In these terms, one of the TM Forum's contributions is the introduction of the Open Digital Architecture (ODA) [28], which provides scenarios for Business and Infrastructure Functions and their respective implementation through technology neutral "*flavours*". These implementations are offered in the form of Open APIs, allowing vertical customers to interact and consume offered X as a Service (XaaS), where X refers to the resource under consideration (e.g. network slice, network service, etc.).

Considering the vendor agnostic nature of the forementioned Open APIs, accompanied with their bottom-up composition across layers and broad adoption from telecommunications industry make TM Forum's Open APIs a perfect candidate for integrating components among multi-vendor environments.

TM Forum uses ODA as reference architecture to elaborate on network slicing implications. From a customer-facing viewpoint, that also affect this project, the following references are relevant:

- GB999 – ODA Production Implementation Guidelines [29]: presents slice management architectures and use cases as derived from various catalysts. Additionally, it references the set of Open APIs that could be used for slice management, including APIs for service catalogue management [4], service ordering [7], service inventory management [30].
- TMF909A – Network as a Service (NaaS) Component Suite Profile [31]: covers the operations required to be exposed in order to provide the functionality required across interworking Operational Domains.

An example of TMForum's usage is in 5GASP and the product model. This API model is utilised for publishing a Network Application to the 5GASP Network Application Marketplace and therefore making it publicly available, after the 5GASP DevOps experimentation and certification readiness lifecycle is successfully completed. TMF's Product resource model will be used to accomplish this required interconnection, as it widely observed and employed in telecommunication industry. Thus, it is expected that the most suitable resource model of the TMF's Product family to describe a Network Application Marketplace asset is Product Offering. Not only it incorporates adequate resources to describe a Network Application offering, but also ensures the interoperability among other industry implementations.

Taking into consideration the various properties of the aforementioned model, the highlighted ones (see Figure 35) will be leveraged to describe a Marketplace asset. Briefly, a Marketplace asset might contain an attachment (e.g. logo, images, certification links or files), topological information about the offered deployment, pricing, asset's specific characteristics, Service Level Agreement (SLA) reference and lastly, a reference to the actual services ordered and employed i.e. hosting network slice, Network Application and test descriptor. Notable mention should be made of the latter entity, namely Service Candidate Ref of the Product Offering resource model. This entity associates the product offering with the Service Specifications constituting the onboarding and deployment model.

To end up, utilising TMF's Product aims at:

- Consistency between the ordering and deployment model
- Introduction of business aspects, such as pricing, product options, market segment
- Imposing an abstraction layer between customer and service provider
- Effortlessly interacting with other production systems

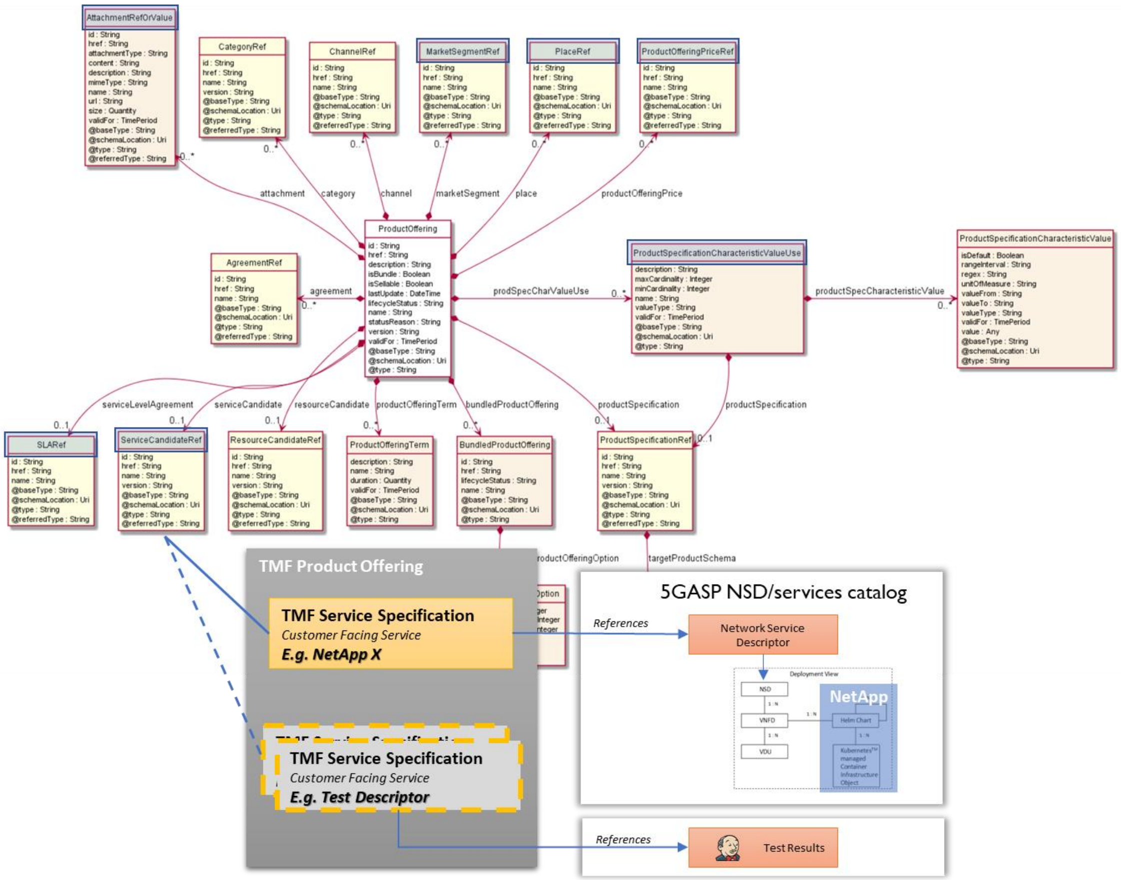


Figure 35: Product Offering resource model in 5GASP

8.2.2 MarketPlace BackEnd

8.2.2.1 Artifact Registry

The Artifact Registry is the underlaying system that stores, exposes, and provides secure and trusted access to the collection of software artifacts that comprises the Network Applications that has been part of a transaction at the marketplace.

An artifact registry is a system for storing and managing artifacts, including container images, Helm charts, language packages and other format of files [17]. For each of the artifact types supported, the registry is built as the combination of repositories, where each repository comprises a collection of related artifacts with the same name (or ID) and available with one or more different tag (or version) [19].

The granularity of the artifacts generated during the implementation of Network Applications depicted in chapter 6 highlights the need for a system such as this one. For all the inner technology layers shown in Figure 3 at least those related to the cloud-native industry, there is a wide range of options in the market to provide this service such as Jfrog Artifactory, Dist, MyGet, Google Artifact Registry [18] etc. However, the remaining telco-specific technology layers, NFV and Network Application, are still mostly managed in a manual way or using a specific solution developed by each vendor.

8.2.2.2 Network Application Repository

The repository for the artifacts corresponding to the Network Application technology layer has only been implemented in few cases out of the projects evaluated 5GASP and EVOLVED5G. Approaches such as a git-based repository and a web server repository have been used.

EVOLVED-5G has a Network App repository which is composed of two separate artifacts. The first one is GitHub on which EVOLVED-5G has created a template to generate the Network App file structure as part of the Software Development Kit (SDK) toolchain. In this repository, the developers can share their Network App code. The second repository, namely the Open Repository is used to store and manage all the Network App images (binaries) and is an extension to other repositories similar to source code repositories, i.e., GitHub. For the validation and certification stages, the Repository will be connected to a third component of the workspace, allowing CI/CD life cycle [43].

5GASP has put in place a Network App Marketplace that provides a public registry of SMEs and their registered products: reusable Network Apps, Network Function (NF) and Network Service (NS) links to open-source repositories, and useful documentation that an SME needs to know also to put in place a certification process. This portal is complemented by a Network App community that supports third parties in development [43].

The VITAL-5G Open Online Repository is one of the core components of the VITAL-5G Platform, providing the catalogue of the Network Apps developed for the project and giving the opportunity to third-party experimenters and developers to download and select them for experimentation, as well as to onboard their own Network Apps to build new vertical services. The main artifacts are 3 different catalogues that allow respectively to query and onboard Network App packages, Vertical Service Blueprints, and Descriptors, as well as Experiment Blueprints and Descriptors, together with their associated VNF packages and NFV Network Service Descriptors [43].

In 5G-MEDIA HUB there is a Network Apps Repository application that has the main purpose to enable Network Apps to be onboarded on the underlying 5G testbeds. This allows the users to design, validate and deploy their Network Apps based on the set of available VNF present in the service catalogue. The Catalogue Management and Service Ordering features from the underlying 5G testbeds allow the users to design Network Apps in a "drag and drop" modality where the constituent VNFs can be dropped onto the canvas and connecting the VNFs to generate the forwarding graphs [43].

The Smart5Grid [42] project proposes an artifact registry based on concepts used in the Helm [57] and OSM [58] repositories which manages the NFV and Network Application technology as a web server. The proposed solution improves the interoperability between the stakeholders participating in the Network Application marketplace by offering a generic telco-specific implementation that prevents vendor lock-in as it is easily extended to support new providers. Artifacts are pushed, pulled and listed using the same approach as in the Helm chart repository which is based on an index file written in yaml [57] that contains the required keys to characterize each artifact (NS, VNF, Network App).

8.2.2.3 Network Application provisioning

To create Network Applications, a developer will employ a variety of artifact types, both for their functionalities, and for the privacy of the components used. Each component of a Network Application should be developed to be a self-executable function that exposes its useful information and retrieves the information required to run with an API protocol described in section 4.3.

A generic virtualized component must agree on a predefined artifact structure, to be compliant with the orchestration system running on top of the platform. Therefore, as described in section 4.1.1, there is a layer called “Virtualization Layer” which takes the application logic as input, and its output will be a standardized version that abstracts the implementation of the components.

Through effective separation of the application from the underlying infrastructure, the Docker platform offers virtualization for various application logics.

Containers are used to package each application with all its required dependencies and libraries, thus ensuring consistency across different environments, from a test environment to a 5G enabled testbed. This isolation from the underlying operating system enhances developers' ability to deploy and manage applications efficiently, with reduced dependency on the infrastructure.

DevOps can help the developers packaging and containerizing their application logic in order to be compliant with the other components developed. Automatic pipelines can be added to the development environment in order to automatically build, analyse and package the component in a compliant format with the other virtualized component.

A DevOps pipeline is a set of processes, tools, and technologies used for continuous integration, delivery, and deployment of software applications. It is designed to automate the entire software development process, from code changes to deployment to production. The pipeline enables faster and more efficient development and deployment of applications, reducing development times and increasing the frequency of deployments.

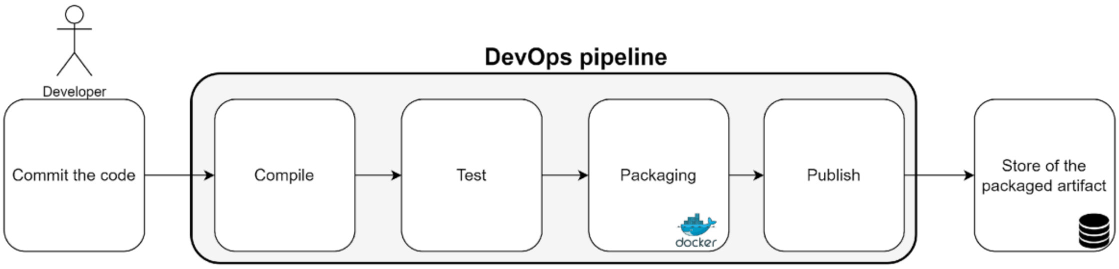


Figure 36: DevOps pipeline

A high-level DevOps pipeline for the development of a component can be viewed in Figure 3. As shown, the developers have the only task of developing their functionalities using the docker container development paradigm. Overall, Docker container development is a popular choice among developers, largely because of its portability and scalability benefits.

After the functionalities have been developed and the code committed in a Version Control System, all the operations that should be done for compiling, testing, packaging, and publishing the component to be available to be used in a Network Application, are done automatically by the DevOps pipeline that it takes care of it. After that, the artifacts, are available and are structured to be linked among them.

The management of components within the framework is facilitated through a Centralized Registry, responsible for storing and distributing component artifacts/images. This Registry can take the form for example of a Docker Registry accessible via VPN to approved third parties, such as Network Application developers and other platforms like Version Control Systems or Network Application Composition applications. These images are made available according to their respective license agreements and can be searched and retrieved by a composer for use in constructing Network Applications. For direct access, developers can use Registry APIs. The functionalities provided by this component include Component Registry Management, which allows for the retrieval and deployment of components through the Registry. For private artifacts, a private registry should be used and connected to the Network Application composer in order to not share the private artifacts/images with other actors using the same composing framework.

With this in mind, it can be possible to share components between Network Applications and it can be possible to share components also between projects. Of course, every Network Application and every component that composes a Network Application has its own logic and has been developed for a particular project/category. For reusing components, they should be integrated with the others. The integration step is key for the composition of all the Network Applications, while every component which is made available to be used in a Network Application, needs to be well documented in order to let another developer use it.

Each component can be seen as a building block to create a complete Network Application. To enable communication between the various components of a Network Application, the components that are used to build the application should communicate between them, using a previously agreed communication protocol.

8.3 Business models toward monetization

API monetization refers to the process of generating revenue by offering Application Programming Interfaces (APIs) to developers, verticals, or users. APIs allow different software systems/components to communicate and interact with each other, enabling the exchange of data and functionality. As APIs is an integral part of modern technology ecosystems and specifically the Network Applications, finding effective ways to monetize them has become crucial. Here are some common methods for API monetization:

- **Subscription Plans**: With subscription plans, the Network Applications providers offer different tiers or levels of access to their APIs for a fixed recurring fee. Each tier may offer varying levels of features, support, or usage limits. Subscription-based monetization allows Network Applications providers to establish long-term relationships with developers or customers, provide reliable support, and ensure a predictable revenue stream.
- **Developer Revenue Sharing**: In this model, Network Applications providers offer their APIs to applications developers for free and generate revenue by taking a percentage of the revenue generated by the applications developers using the API. This approach is often used in marketplaces where developers build applications or services on top of the API and share a portion of their earnings with the API provider.
- **API Usage Fees**: This approach involves charging applications developers or vertical businesses based on the usage of the API. It can be implemented through various pricing models, such as pay-as-you-go, tiered pricing, or metered billing. Developers are typically billed based on the number of API calls, data transferred, or specific features utilized. This method allows API providers to align revenue with usage and scalability.
- **Data Licensing**: Some APIs provide access to valuable datasets that can be used for analysis, research, or integration into other systems. Network Applications providers can monetize their APIs by licensing access to these datasets, either on a per-usage basis or through subscription plans, allowing customers to leverage the data for their own purposes.
- **Freemium Model**: With the freemium model, Network Application providers offer a basic version of the API for free, but charge for additional features, functionality, or higher usage limits. This allows customers or applications developers to experiment and build prototypes with the API at no cost, while providing an incentive to upgrade to a paid version for more advanced requirements.
- **Partner Programs and Add-ons**: Network Applications providers can create partner programs or offer add-ons that enhance the functionality of their APIs. This can include additional tools, plugins, integrations, or services that complement the core API offering. Network Applications providers can charge fees or royalties for these value-added offerings, generating additional revenue streams.
- **Customization and Consulting Services**: Some API providers offer customization services or provide consulting to assist customers in integrating their APIs into specific projects or systems. These services can be charged on an hourly or project basis, providing an opportunity for API providers to generate revenue beyond the API itself.

Choosing and selecting the right monetization strategy depends on various factors, including the target market, competition, value proposition, and the specific needs of the developer or user community. A combination of different methods might also be employed to maximize revenue and cater to different customer segments.

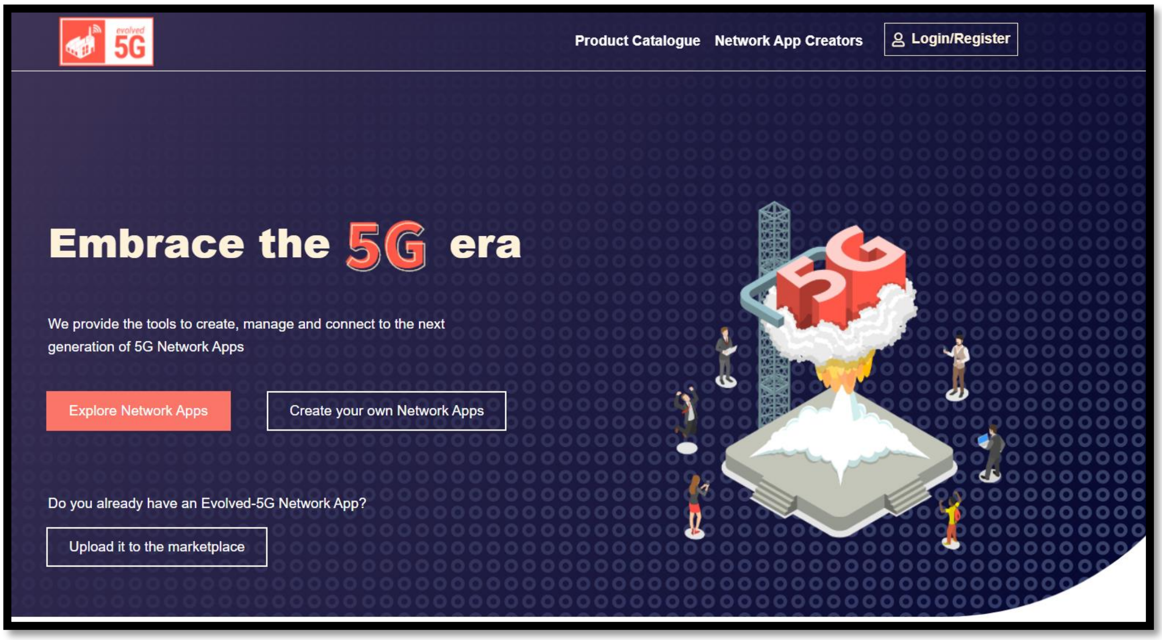
8.4 Example of marketplaces

8.4.1 EVOLVED-5G

The EVOLVED-5G Marketplace [12] is a fully operational online store with fully transactional capabilities where Network Applications developed in the EVOLVED-5G project can be downloaded. Users are able to search, filter, purchase –or obtain for free-, and download these Network Applications, and developers are able to upload their own Network Applications to the marketplace for sale. Moreover, developers can monitor statistics regarding the Network Application key historical data (visits, purchases) in a visual and interactive manner.

It represents a secure commercial environment, where secure transactions along with standard security measures have been implemented.

Currently, the EVOLVED-5G Marketplace is available at https://marketplace.evolved-5g.eu/, see Figure 37, and provides a wide set of Network Applications for different vertical use cases, with a notable variety on its product catalogue.



Embrace the 5G era

We provide the tools to create, manage and connect to the next generation of 5G Network Apps

Do you already have an Evolved-5G Network App?

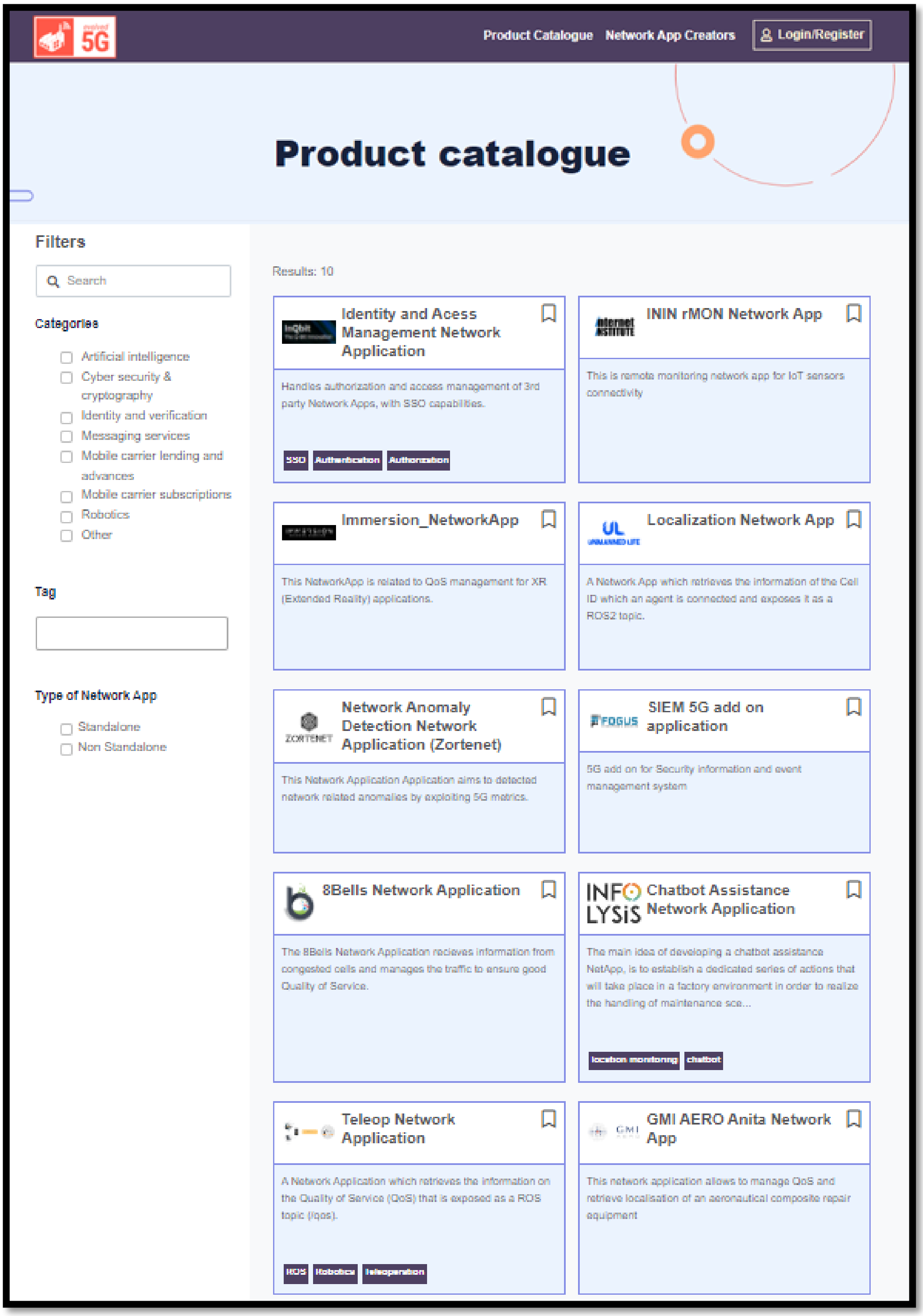
Figure 37: EVOLVED-5G Marketplace, available at https://marketplace.evolved-5g.eu/

This EVOLVED-5G transactional platform presents important features for a marketplace such as:

- Blockchain support for transaction record, with the possibility of performing Ethereum purchases. Each transaction is verified and recorded through a complex cryptographic process, and the resulting information is stored across a distributed network of computers. This means that all participants in the network have access to the same information, which can be verified and audited in real-time. It provides a unique and tamper-proof record of transactions.
- User-friendly interface both for end-users, Network Applications developers, and administrators.
- Automated upload, with immediate purchase availability of Network Applications. Network Applications can be manually or automatically uploaded to the marketplace

platform, via a set of pipelines connected to the Open Repository in the EVOLVED-5G framework where Network Applications are internally stored.

- Security: transactions are secure. The Marketplace implements solid security and privacy measures. Access to the Marketplace is only available via secure connection, employing security certificates and requiring user authentication.
- User management and secure identification.
- Excellent positioning: The EVOLVED-5G marketplace is the first option in search engines, facilitating immediate access. Users do not need to search the marketplace along different resources and can access immediately through the first link available.
- Online availability – Any user, at any time and at any place with just an internet-enabled device can access the marketplace. Virtually, that provides potential universal access.
- High usability
- API for marketplace control and monitoring, including Ethereum transactions using Blockchain technology.
- Open source: it is aligned with the open-source vision.
- Dockerized – Marketplace is containerized using Docker technology; thus it presents the ability to run instantly and seamlessly on any OS, regardless of the underlying technology. This feature allows for seamless portability, server migration, and the future extension and diversification of marketplaces.
- Administrative dashboard: Graphical interface of high usability that allows for efficient management of the marketplace.
- Support for Marketing campaigns. A set of landing pages have been created targeting the Marketplace users. These pages explain the value propositions of the Marketplace in simple terms and can be used to a) attract users organically by search engines or b) create paid marketing campaigns.



Product catalogue

Results: 10

Identity and Acess Management Network Application
Handles authorization and access management of 3rd party Network Apps, with SSO capabilities.
SSO | Authentication | Authorization

ININ rMON Network App
This is remote monitoring network app for IoT sensors connectivity

Immersion_NetworkApp
This NetworkApp is related to QoS management for XR (Extended Reality) applications.

Localization Network App
A Network App which retrieves the information of the Cell ID which an agent is connected and exposes it as a ROS2 topic.

Network Anomaly Detection Network Application (Zortenet)
This Network Application Application aims to detected network related anomalies by exploiting 5G metrics.

SIEM 5G add on application
5G add on for Security information and event management system

8Bells Network Application
The 8Bells Network Application recieves information from congested cells and manages the traffic to ensure good Quality of Service.

Chatbot Assistance Network Application
The main idea of developing a chatbot assistance NetApp, is to establish a dedicated series of actions that will take place in a factory environment in order to realize the handling of maintenance sce...
location monitoring | chatbot

Teleop Network Application
A Network Application which retrieves the information on the Quality of Service (QoS) that is exposed as a ROS topic (/qos).
ROS | Robotics | Teleoperation

GMI AERO Anita Network App
This network application allows to manage QoS and retrieve localisation of an aeronautical composite repair equipment

Figure 38: Network Applications catalogue of the EVOLVED-5G marketplace

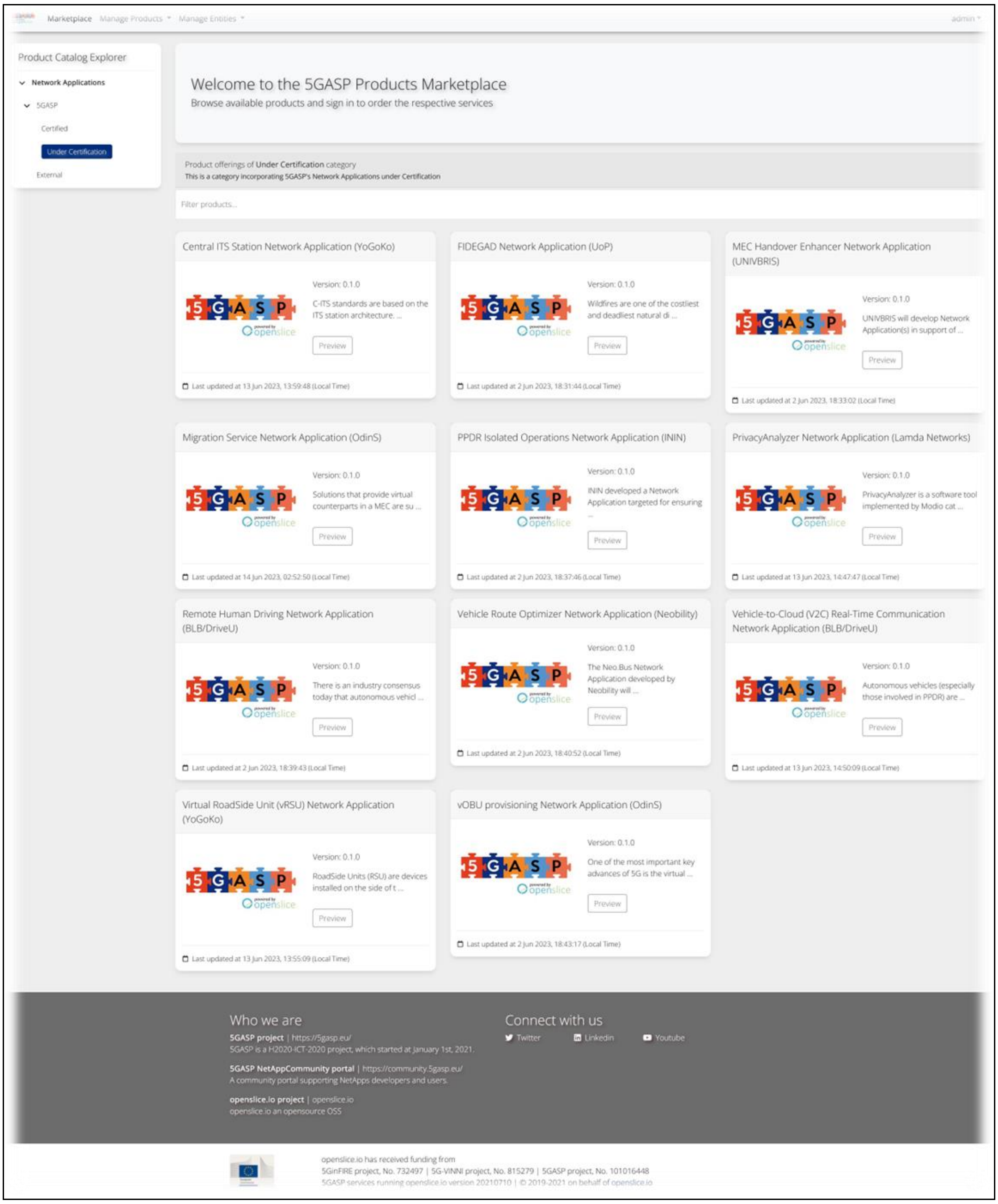
Furthermore, the existence of a Marketplace also has implications for the creation and fostering of a community around these developments, as it brings significant visibility to them, and allows to potential developers and other stakeholders their further adoption and usage. In this sense, other elements from the EVOLVED-5G framework complement this action:

- EVOLVED-5G Forum [12] provides a space where Network Applications users, developers and other stakeholders can publish their inquiries and support answers regarding EVOLVED-5G elements. There is a specific category to provide support on purchased Network Apps, retrieved through the marketplace.
- EVOLVED-5G Wiki [12] provides how-to technical documentation regarding Network Applications usage and deployment.
- EVOLVED-5G GitHub [12] provides the open-source code of the released Network Applications.

8.4.2 5GASP Marketplace

The 5GASP marketplace [43] is a showcase portal of registered Network Applications that provide innovative solutions for business, validated through independent testing on the 5GASP platform along with the respective operation information collected during the process.

Currently, the 5GASP marketplace operates at https://portal.5gasp.eu/products (Figure 39), offering a variety of Network Applications from miscellaneous verticals through its set of publicly available product catalogues, ready to be seamlessly extended due to the employed standardized interfaces.



Welcome to the 5GASP Products Marketplace

Browse available products and sign in to order the respective services

Product offerings of **Under Certification** category

This is a category incorporating 5GASP's Network Applications under Certification

Central ITS Station Network Application (YoGoKo)

Version: 0.1.0

C-ITS standards are based on the ITS station architecture. ...

Last updated at 13 Jun 2023, 13:59:48 (Local Time)

FIDEGAD Network Application (UoP)

Version: 0.1.0

Wildfires are one of the costliest and deadliest natural di ...

Last updated at 2 Jun 2023, 18:31:44 (Local Time)

MEC Handover Enhancer Network Application (UNIVBRIS)

Version: 0.1.0

UNIVBRIS will develop Network Application(s) in support of ...

Last updated at 2 Jun 2023, 18:33:02 (Local Time)

Migration Service Network Application (OdinS)

Version: 0.1.0

Solutions that provide virtual counterparts in a MEC are su ...

Last updated at 14 Jun 2023, 02:52:50 (Local Time)

PPDR Isolated Operations Network Application (ININ)

Version: 0.1.0

ININ developed a Network Application targeted for ensuring ...

Last updated at 2 Jun 2023, 18:37:46 (Local Time)

PrivacyAnalyzer Network Application (Lamda Networks)

Version: 0.1.0

PrivacyAnalyzer is a software tool implemented by Modio cat ...

Last updated at 13 Jun 2023, 14:47:47 (Local Time)

Remote Human Driving Network Application (BLB/DriveU)

Version: 0.1.0

There is an industry consensus today that autonomous vehicl ...

Last updated at 2 Jun 2023, 18:39:43 (Local Time)

Vehicle Route Optimizer Network Application (Neobility)

Version: 0.1.0

The Neo.Bus Network Application developed by Neobility will ...

Last updated at 2 Jun 2023, 18:40:52 (Local Time)

Vehicle-to-Cloud (V2C) Real-Time Communication Network Application (BLB/DriveU)

Version: 0.1.0

Autonomous vehicles (especially those involved in PPDR) are ...

Last updated at 13 Jun 2023, 14:50:09 (Local Time)

Virtual RoadSide Unit (vRSU) Network Application (YoGoKo)

Version: 0.1.0

RoadSide Units (RSU) are devices installed on the side of t ...

Last updated at 13 Jun 2023, 13:55:09 (Local Time)

vOBU provisioning Network Application (OdinS)

Version: 0.1.0

One of the most important key advances of 5G is the virtual ...

Last updated at 2 Jun 2023, 18:43:17 (Local Time)

Who we are

5GASP project | https://5gasp.eu/
5GASP is a H2020-ICT-2020 project, which started at January 1st, 2021.

5GASP NetAppCommunity portal | https://community.5gasp.eu/
A community portal supporting NetApps developers and users.

openslice.io project | openslice.io
openslice.io an opensource OSS

Connect with us

Twitter Linkedin Youtube

openslice.io has received funding from
5GinFIRE project, No. 732497 | 5G-VINNI project, No. 815279 | 5GASP project, No. 101016448
5GASP services running openslice.io version 20210710 | © 2019-2021 on behalf of openslice.io

Figure 39: 5GASP marketplace

Specifically, the 5GASP marketplace operates similarly to the mobile application stores, where i) selected Network Applications are provided for a fee; ii) some other Network Applications are provided for free; and iii) third parties either gain visibility and/or make revenue when they publish their Network Application, under the condition that it had been deployed, tested and certified by the 5GASP platform.

To publish a Network Application, the developers first need to create their own account on the portal. Then, the respective Network Application is submitted to the 5GASP portal for autonomous testing, validation, and certification employing the state-of-the-art 5GASP facilities and testing tools. After the DevOps experimentation and certification readiness lifecycle is completed, the Network Application is automatically published at the 5GASP marketplace, therefore rendering it publicly available, where customers can purchase it and operate it tailored to their needs.

The interface employed for publishing the Network Application to the 5GASP marketplace is based on TMF's Product-resource model, as it is widely observed and extensively used in the telecommunication industry. Specifically, it is expected that the most suitable resource model of the TMF's Product family to describe a Network Application marketplace asset is Product Offering. Not only it incorporates adequate resources to describe any given Network Application offering, but also ensures interoperability among other industry implementations.

Taking into consideration the various properties of the aforementioned model, a Marketplace asset might comprise an attachment (e.g. logo, images, certification links, or files), topological information about the offered deployment, pricing, asset's specific characteristics, Service Level Agreement (SLA) reference and lastly, a reference to the actual services ordered and employed i.e. the hosting network slice, the Network Application, and the respective test descriptor.

Eventually, employing such a broadly operated resource model, as the TMF's Product, aims at:

- Consistency between the ordering and deployment model
- Introduction of business aspects, such as pricing, product options, market segment
- Imposing an abstraction layer between the customer and service provider
- Effortlessly interacting with other production systems

9 Standard status

In this section, we outline the standard status on the 5G API requirements for services definition, common API framework proposed in 3GPP and CAMARA initiative between the GSMA (reference telco association) and the Linux Foundation (reference cloud association).

9.1 5G Telco API requirements for services definition

5G system architecture is the new communication service enabling model that leverages service-based interactions between 5G Core control plane and the network functions through secured and exposed platform's APIs. In principle, it is based on the Service Based Architecture (SBA) framework concept, creating the modularized services deployment, on-demand networks implementation, fast deployment cycles, dynamic services launch in the network. The main API requirements are focused on the 5G Core functions, interaction between components in SBA and on the management and orchestration capabilities of the 5G network. Thus, a Service Based Interface (SBI) represents how a given NF exposes a set of services for 5G Core interfaces specified in 3GPP TS 23.501. A sketch of SBA framework in 5G system control plane can be found in Figure 40. Among these NFs, both Network Repository Function (NRF) and Network Exposure Function (NEF) are of concern.

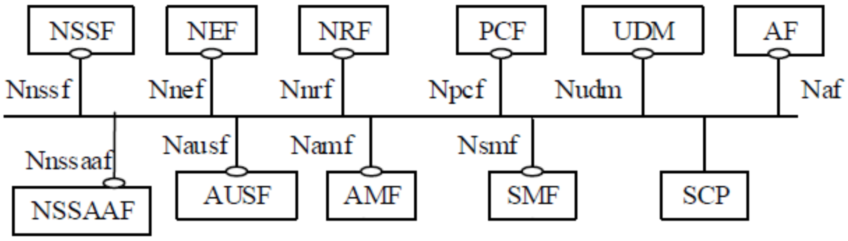


Figure 40: 5G Core Control Plane Architecture

The NRF is the entity responsible for selecting the network functions for the PDU session based on services profiles, network states, policies, or other goals, viewed as an evolution of the existing 3GPP DNS systems combined with the selection logics in MME in 4G, integrated with additional intelligence for NF policy selection. Note that the NF Service Framework should include service registration, update and deregistration in order to make the NRF aware of the available NF instances and supported services. This can enable a NF service consumer to discover the NF Service Producer that can provide the expected NF service (with access authorization), through three main services offered by NRF: (1) Nnrf_NFManagement (2) Nnrf_NFDiscovery and (3) Nnrf_AccessToken. A depict of the three main service is in Figure 41, and more services can be found in 3GPP TS 29.510.

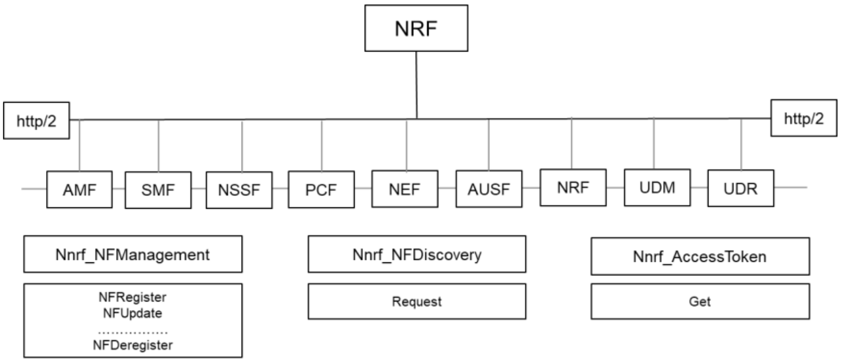


Figure 41: NRF Service Based Interfaces

The main tasks of NRF are to maintain the NF profiles of available NF instances, enable new NF subscription and registration, support service discovery functions, and interact with every NF in the 5G Core. The services offered by NRF to the NF (e.g., the three aforementioned service in Figure 41) are using the open API, which is defined in 3GPP TS 29.510, e.g., nnrf-nfm, and nnrf-disc. The design details for the SBIs specified by 3GPP includes: (i) API purpose, (ii) URIs of resources, (iii) HTTPs supported methods and supported representation (e.g., JSON), and (iv) Request/response body schema. A depict of 5G NRF service APIs can be found in Figure 42.

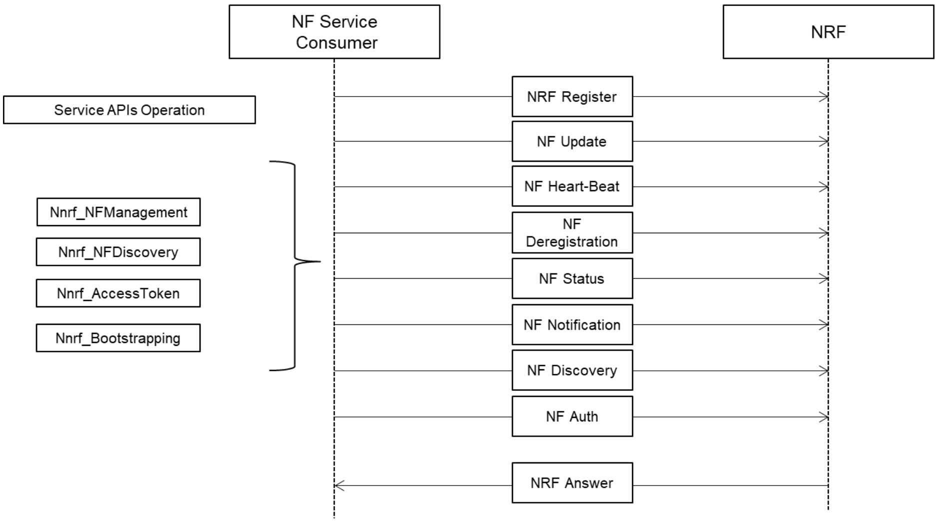


Figure 42: NRF Services APIs

The NEF is the entity responsible for securely exposing different network capabilities, services and functions provided by 3GPP NF to the 5G customers (e.g., 3rd party, Application Function (AF), Edge Apps). And it leverages the exposed APIs to expose the required network information, as the architecture depicted in Figure 44.

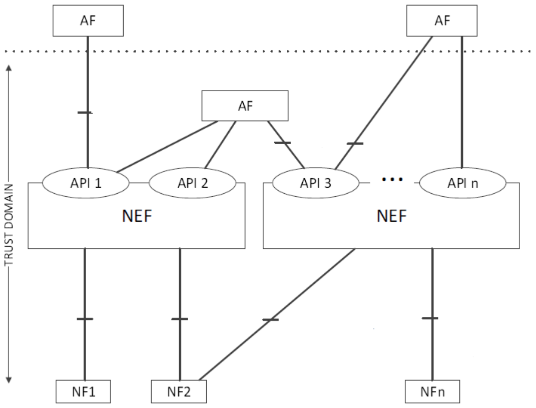


Figure 43: 3GPP Architecture for Network Exposure

NEF comprises two main exposure function services: NEF northbound APIs and NEF southbound services. The NEF Northbound Interface is the RESTful API interface between the NEF and AF that supporting several procedures, such as monitoring, triggering, provisioning, traffic influence and AF session QoS, corresponding to the supported NEF servicers, i.e., Nnef_Interfaces between NEF and AF, as described in TS 23.502 and represented in Figure 44. Moreover, it supports functionalities related to secured network capabilities exposure, AF to 3GPP network authentication, and authorization information. The security is offered by using secured communication between NEF and AF over the NEF northbound Interface, accessing the SCEF authorized APIs (OAuth2 protocol).

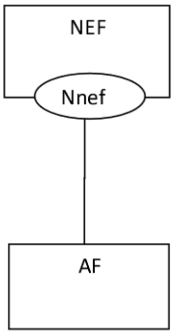


Figure 44: NEF Services SBI representation

In addition, NEF Northbound APIs supports the common API framework (will be tailored in section 9.2) and is based on the service-based interaction, as presented in Figure 45. It relies on a series of services operation activities between NEF and the consumers in place, such as Service Subscribe, Service Unsubscribe, Notify, Update through the REST APIs.

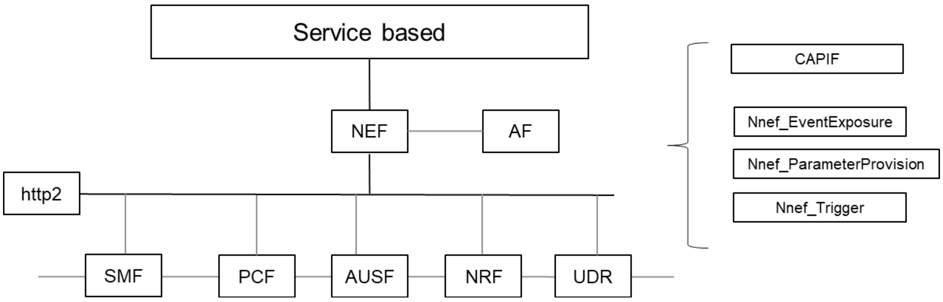


Figure 45: NEF Services Interaction

To enable flexible communication between UPF and other 5G Core NFs, the UPF can expose network information to NFs through UPF event exposure service.

In Release 17, TS 23.548 [23.548] has supported that UPF can expose QoS monitoring results to local NEF. In Release 18, the UPF can also expose other useful information to other NFs to optimize the network performance.

For that the service-based interface Nupf is introduced in the 5G system architecture to support registration, deregistration and discovery via NRF.

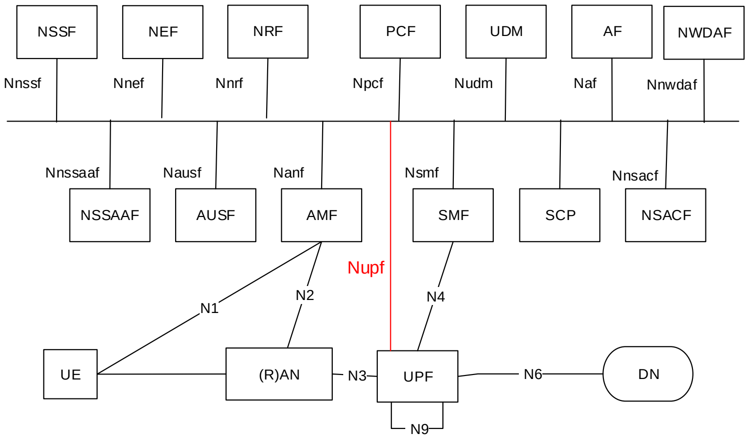


Figure 46: 5G Core architecture with service based UPF

The UPF registers directly with the NRF and does not use N4 for registering to NRF.

9.2 Common API Framework

The use of Application Programming Interfaces (APIs) has served for many years as a bridge between mobile operators and start-ups in emerging markets [33]. Operators have begun to consider whether to open their APIs, starting form APIs related to mobile messaging, operator billing etc. In addition, the recently witnessed convergence of IT and Telecom worlds have contributed a lot on putting APIs in the epicentre of network programming and service provisioning. Representative examples that prove this statement are: the 5G Service Based Architecture (SBA), which has been designed based on the flexibility that HTTP/2 Restful APIs to provide interaction among 3GPP network functions; the intense work on API specification and development in open project and fora (e.g., the OpenAPI project of TM Forum); and the wide adoption of the microservice software pattern/architecture which is based on API interacting software pieces.

In the same direction, Network Applications [1] provide network- or vertical- oriented services, meaning that they can assist/enhance either the network operation/management[7] or the vertical application[8]. As third-party applications, the Network Applications should interact with network functions/nodes though open and standardised interfaces/APIs that can reside at any plane (user, control, management) or any domain (core, radio, transport). Thus, a widely accepted and standardised API framework is needed, in order to guarantee interoperability and security in that interaction.

[7] For instance, in the EVOLVED-5G project a related contribution to 3GPP SA6 work has emerged (3GPP/TSG SA2/eNA_Ph2 Rel.17: Contribution "Support of DN performance analytics by NWDAF" - S2-2101388) under the scope of extending the NWDAF analytics APIs so that Network Applications can retrieve data from vertical apps, and the NWDAF build performance analytics & predictions by using inputs from Network Applications.

[8] The ICT-41 projects (5GPPP, phase 3, part 6 projects), work towards providing Network Applications that fulfil needs and requests from various vertical industries, e.g., automotive (5GIANA, 5GASP), Industry 4.0/manufacturing (5GINDUCE, EVOLVED5G, 5GERA), transport & logistics (VITAL5G, 5GERA), media (5GMediaHub), public protection and disaster relief (5G-EPICENTRE, 5GERA, 5GGASP), healthcare (5GERA).

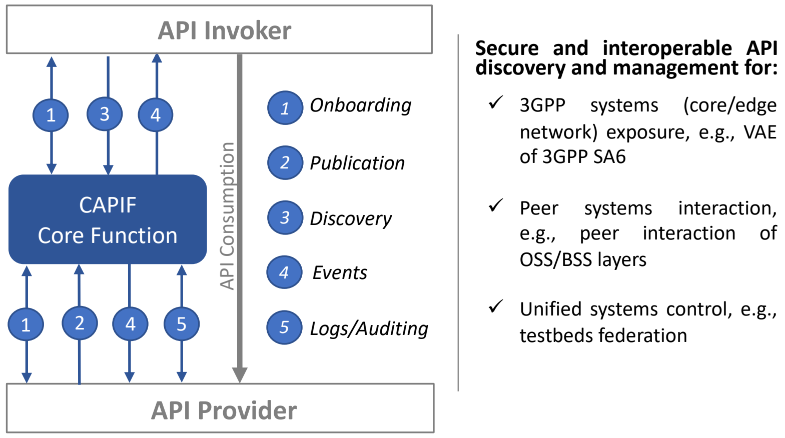

API Invoker
1
3
4
CAPIF
Core Function
1
2
4
5
API Consumption
API Provider
1 Onboarding
2 Publication
3 Discovery
4 Events
5 Logs/Auditing
Secure and interoperable API discovery and management for:
✓ 3GPP systems (core/edge network) exposure, e.g., VAE of 3GPP SA6
✓ Peer systems interaction, e.g., peer interaction of OSS/BSS layers
✓ Unified systems control, e.g., testbeds federation

Figure 47: Abstract illustration of the CAPIF functionality, with representative examples of its applicability

The 3GPP Common API Framework (CAPIF) is adopted as the main candidate framework for that purpose. CAPIF has been an integral part of the 3GPP SA6 specifications since Rel. 15 [11] and continues to evolve with new features and reference points, added in any following Release. Its objective is to provide a unified and standardized northbound API framework across several 3GPP functions. Practically, it has been designed to facilitate the network core exposure, towards new application enablers of various vertical industries (including, Unmanned aerial systems, Edge data networks, Factories of the future, and Vehicular communication systems). Beyond this initial target, CAPIF has been used as a key standardized API-management framework for secure and interoperable interaction among any API providers and API consumers Figure 47.

Already, Telefonica (Spain) and FOGUS Innovations & Services P.C. (Greece) co-developed the CAPIF services and provide them for the first time as an open-source solution [22][23]. The implementation that they provide is fully compliant with the related specifications (specifically with the 3GPP Rel. 17 [9]), and it is also accompanied by test plans and ready to use templates.

9.2.1 CAPIF functional architecture and services

Three main entities have been defined in the CAPIF architecture, namely, the API invoker, the CAPIF Core Function (CCF), and the API provider. The API invoker is typically provided by a third-party application that needs to consume APIs from an API provider. The CAPIF core function is the main entity of CAPIF which is in charge of: i) authenticate an API invoker based on identity and/or other information; ii) authorise an API invoker prior to accessing service APIs, iii) onboard/offboard API invokers, iv) monitor service API invocations, and v) store policy configurations related to CAPIF and service APIs. The API provider is an entity that provides API Exposing, Publishing and Management functions. More precisely, the API exposing function (AEF) is the provider of the service APIs; the API publishing function (APF) enables the API provider to publish service APIs to CCF in order to enable the discovery of those APIs by an API invoker; and the API management function (AMF) enables the API provider to perform administration of the service APIs.

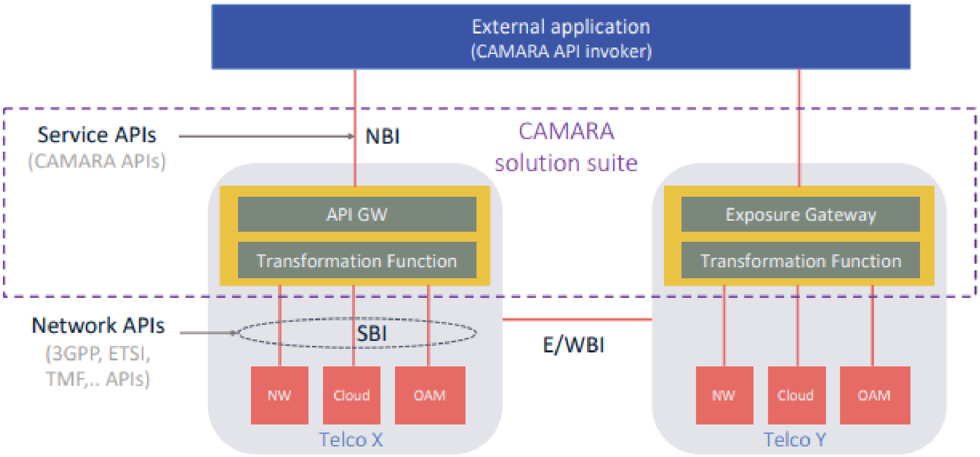
9.2.2 CAPIF for secure API-based interaction

While the interoperability factor of CAPIF is given "by design", since CAPIF specifications are standardized, the security aspects that it brings in Network Application interaction with underlay network functions or nodes is further justified here. More precisely, mutual authentication based on client and server certificates is performed between the CCF and the API Provider/Invoker, using TLS. It is noted that TLS provides integrity protection, replay protection and confidentiality protection for all the reference points of the CAPIF architecture. Certificates to be used shall follow the profiles given in 3GPP TS 33.310 [24]. At the application/user layer, CAPIF shall be able to authenticate and authorize each user asking for API services and also, shall coordinate authentication and authorization between API Invoker and AEF. It is noted that, the authentication process refers to the process of verifying the identity of a user by obtaining some sort of credentials (e.g., username, password). Authorization, on the contrary, is the process of allowing an authenticated user to access resources by checking whether the user has access rights to those resources. For those processes PSK, PKI, or OAuth methods can be used. Overall, based on the specifications (TS 33.310 [24]), the following methods have been defined:

- Method 1 – Use of TLS with PSK. A Pre-Shared Key (PSK) is used for the API Invoker and AEF interaction.
- Method 2 – Use of TLS with PKI. Mutual authentication is provided. It is assumed that both API invoker and AEF are pre-provisioned with certificates created by a relates engine.
- Method 3 – Use of TLS with OAuth token. The CCF shall perform the functionalities of the Authorization and token protocol endpoints, the API invoker shall perform the functions of the resource owner, client and redirection endpoints functionalities, while the AEF shall perform the resource server functions.

9.3 CAMARA

The text of this section is extracted from the following reference [25]. CAMARA is a joint initiative between the GSMA (reference telco association) and the Linux Foundation (reference cloud association). Its mission is to foster the definition, development and validation of APIs enabling NaaS, promoting their usage and de-facto adoption by using an Apache 2.0 license. In this endeavour, CAMARA counts on an open and ever-growing community which gathers frontline industry stakeholders, including vendors, tier-1 operators, hyperscalers and solution integrators, among others. The up-to-date list of companies participating in CAMARA can be found in [26].



NOTE: E/WBI stands for East/Westbound Interface, and calls for federation capabilities across telco operators. This is transparent to 3rd parties, and therefore out of scope of CAMARA project

Figure 48: CAMARA architectural framework

Figure 48 depicts the reference architectural framework of CAMARA. As seen, it includes the following components:

- Network APIs: These are the APIs which are implemented in telco assets, including network resources (core, access, transport functions), cloud resources (virtualized and cloud-native workload hosting infrastructures) and IT resources (OSS and orchestration tools). These APIs are typically defined in standard bodies or industry fora and tied to the underlying technology. Examples of these APIs include the ones defined by 3GPP, ETSI or TMForum, among others.

- Service APIs: These are the NaaS APIs, the ones to be made available for consumption to 3rd party applications, a.k.a. external applications. They are designed according to the principles listed in Section 1: open (API source code is Apache 2.0), global reach (they are offered by different telco operators) and user-friendly (service APIs result from the abstraction of network API semantics, hiding telco complexity).
- Transformation function: It keeps the information on correspondences between service APIs and network APIs and executes workflows to enforce these mappings. This component can be deployed as a microservice provisioned with a workflow engine.
- API Gateway: It provides all the capabilities that are needed to policy the interaction between the operator and the external applications, in relation to service API invocation. These capabilities include service API publication & discovery, access control (authentication & authorization of applications), auditing, accounting and logging. As seen, the focus of CAMARA work is on service APIs, and how they can build atop existing network APIs with the help of transformation function and API Gateway.

As seen in Figure 48, the architectural components building up the CAMARA solution suite are two: the transformation function and the API Gateway. The first component can be implemented using a workflow engine, while for the API Gateway the only mandate is that it needs to support OAuth2.0 with client credentials grant [27]. There exists a wide variety of API Gateway solutions, both open-source and commercial, which are OAuth2.0 compliant. However, for the sake of consistency when offering service APIs, it is desirable to agree on a common, standards-based solution for this gateway. In this regard, CAMARA proposes to use 3GPP Common API Framework (CAPIF) [13]. The reason for this proposal is twofold. On the one hand, CAPIF is a normative solution with wide acceptance at industry. The other main advantage of CAPIF is that though specified by the 3GPP, it is not tied to 3GPP APIs; indeed, CAPIF can be used as an API Gateway for any API, regardless of their semantics. Therefore, it is an ideal candidate to publish CAMARA APIs, and policy their exposure to the 3rd party applications.

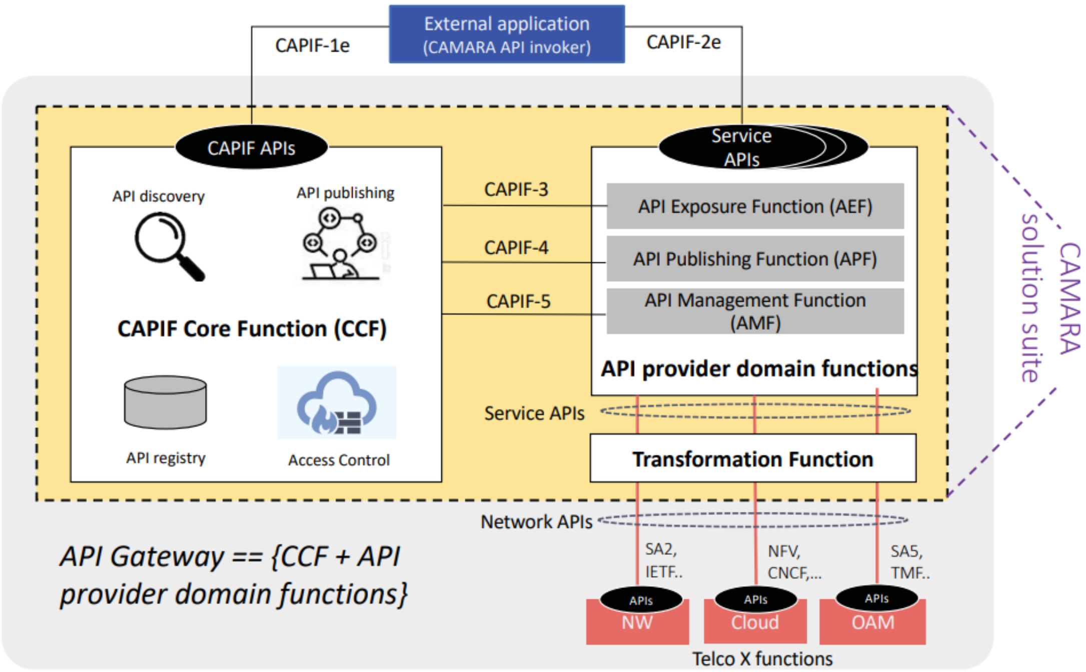


Figure 49: CAPIF as reference API GW for CAMARA

10 Toward 6G

Telecom research activities are a continuous work that addresses the challenges of the future networks but the 3GPP well granted approach in each “G” generation standardisation leads to approximately ten-year cycles. In the 5G to 6G transition, there are plenty of new functionalities under so-called ‘beyond 5G’ with 5G-Advance specially provided and hopefully monetised in vertical B2B businesses. We refer to new functionalities arriving in post release 16 [8], with releases 17 [9] and 18 for 5G-Advance in terms of better slicing management for vertical traffic QoE discrimination that is essential to 5G support for instance vertical critical communication services (i.e. CCAM or FRMCS evolution), adoption of extreme-edge and sensing elements with edge computing seamless support that is important for enabling XR (Extended Reality) and media services, accurate indoor localisation services and deterministic networks (e.g. TSN) that are key to support Industry 4.0 cases or network exposure functions (via NEF entity) and associated data analytics for more automation of network operation and enable vertical application services to take advantage of access to network data. These functionalities request updated software stacks and evolved hardware which normally takes at least two additional years after closing a release for vendors to become a reality in the market and be fully exploitable for vertical service industries.

The general technical trends in ‘beyond 5G’ networks do not radically change in next 6G. 6G research towards 2030+ will continue deep diving in trendy topics such as the disaggregation of the network elements from core functions to radio elements (e.g. O-RAN), the complete virtualisation of network (i.e. NFV/SDN) deeply embracing cloud-native principles and flexible orchestration in the network transformation, the usage of telco-Edge computing capabilities as part of the ‘continuum’, the AI-native adoption at multiple network levels to enable smart control loops (i.e. more autonomous network) while addressing both the increasing attention to next ‘G’ energy consumption (sustainability) and permanent concern on cybersecurity and privacy dimension. While 6G concept is being consolidated with the network of networks and multiple CSPs, a new challenge for exploiting it will be the management of the complexity in which AI will also play an essential role. 6G research should not only focus on more performance (more bandwidth, new frequency bands, lower air-latency, etc) but on bringing added value to stakeholders and end users of vertical industries.

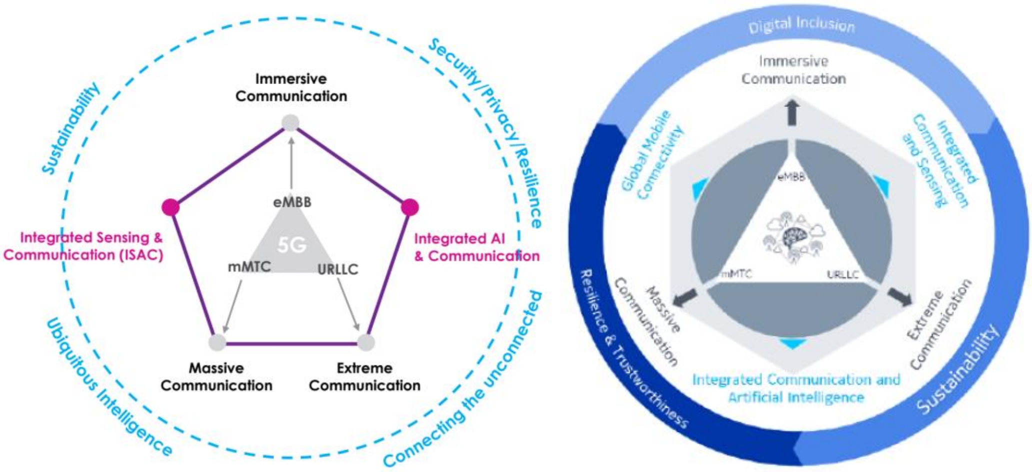

Immersive Communication
Security/Privacy/Resilience
Sustainability
eMBB
5G
mMTC
URLLC
Integrated Sensing & Communication (ISAC)
Integrated AI & Communication
Massive Communication
Extreme Communication
Ubiquitous Intelligence
Connecting the unconnected
Digital Inclusion
Immersive Communication
Global Mobile Connectivity
Integrated Communication and Sensing
eMBB
mMTC
URLLC
Massive Communication
Extreme Communication
Resilience & Trustworthiness
Integrated Communication and Artificial Intelligence
Sustainability

Figure 50: IMT-2030 of 6G / ITU-R vision for IMT-2030

On the other hand, initiatives such the GSMA telco operator platform [33] and open-source communities under Linux Foundation [34] run in parallel to the new ‘G’ and will definitively be the mechanism to increase the impact and accelerate development cycles of the network towards vertical industries (Health, Transport and Logistics, Energy, Manufacturing, etc). The network is a critical enabler for driving innovation in vertical industries. Vertical industries are also impacted by the similar IT evolution to cloud ‘nativeness’ and AI rise and need to understand what can extract and how can easily use the network. This drives the importance of open mechanisms and common API-based framework (e.g. CAPIF) since there is a network-cloud convergence in many segments. CAPIF functionality is a fundamental piece for vertical monetisation since enables third parties utilizing network programmability to its full extension and secure accessing to 5G core APIs to third parties. In this way, vertical application can meet and match the different entry points they need.

This network exposure capability as a northbound service API is the mechanism in which a CSP offers an entry point to get data from network domain and receive management commands. It is expected that 6G expands even more these exposure capabilities to a multi-stakeholder (several CSPs case as it was started in the federated platform CAMARA project) and across multiple network domains, not only the Core part but also RAN for instance to SMO and nRT- RIC (RAN Intelligent Controller) elements in O-RAN terminology and optical transport segment.

Recent events about future networks such as 6G Symposium [35] ‘Beyond the hype’ (London April’23) and ETSI event “Evolving NFV towards the Next Decade” (Sophia Antipolis March’23) [36] or 6G Summit in Berlin April’23 [37] pointed out common need to look at vertical industries, increase the network openness and expand the network exposure via APIs to enable new end-to-end scenarios. 6G will bring growing application environments and the need to address better network data and control in enterprise and industrial environments.

11 Conclusion

This paper is prepared by 5G-PPP software network Working Group, and it complements a series of white paper published by the WG related to the cloud native transformation and the role of the vertical. The aim of this white paper is to demystify the concept of the Network Application (or Network App). The important point of the concept is related to how the telco exposes the capabilities of the 5G/B5G platform, to others business platforms owned by the verticals to enhance their services. This mix and match between services from the telco and services from the verticals is the main driver of the concept of the Network Application. Open platform, API, service exposure, abstraction etc are different topics on which different projects, mainly ICT-41, worked out through diverse vertical use-cases. It results different mode of interaction between verticals and 5G/B5G system leading to three main categories: as a service, hybrid and coupled/integrated models.

12 References

[1] 5G-PPP White Paper, "Network Applications: Opening up 5G and beyond networks", Sept 2022 [online]. Available: https://5g-ppp.eu/wp-content/uploads/2022/10/Software-Network-WG-Network-Applications-2022.pdf

[2] 5G-PPP White Paper, "Service performance measurement methods over 5G experimental networks", May 2021 [online]. Available: https://5g-ppp.eu/wp-content/uploads/2021/06/Service-performance-measurement-methods-over-5G-experimental-networks_short_version_08052021-Final.pdf

[3] ETSI GS NFV-SOL 006, V3.5.1, "Protocols and Data Models; NFV descriptors based on YANG Specification" , https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/006/03.05.01_60/gs_NFV-SOL006v030501p.pdf

[4] Malin Eriksson and V. Hallberg , "Comparison between JSON and YAML for Data Serialization", Economics, 2011

[5] TM Forum, "TMF 633 – Service Catalog Management"

[6] ETSI NFV ISG, "GR NFV-IFA 029, Architecture; Report on the Enhancements of the NFV architecture towards

[7] TM Forum, "TMF 641 – Service Ordering Management"

[8] 3GPP release 16, available online: https://www.3gpp.org/release-16

[9] 3GPP release 17, available online: https://www.3gpp.org/release-17

[10] 3GPP SEAL "Service Enabler Architecture Layer for Verticals", available online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3587

[11] 3GPP CAPIF "Common API Framework", available online: https://www.3gpp.org/common-api-framework-capif

[12] EVOLVED-5G Project: https://evolved-5g.eu/

[13] Common API Framework (CAPIF): https://www.3gpp.org/common-api-framework-capif

[14] VITAL-5G project: https://www.vital5g.eu/

[15] VITAL-5G D1.2 deliverable, "System Specifications and Architecture", [online] available: https://www.vital5g.eu/wp-content/uploads/2022/01/VITAL5G-D1.2-5G-system-specifications-and-architecture_Final.pdf

[16] What is Kubernetes? https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

[17] Helm https://helm.sh

[18] https://cloud.google.com/container-registry/docs/overview

[19] https://www.plutora.com/ci-cd-tools/artifacts-management-tools

[20] https://helm.sh/docs/topics/chart_repository/

[21] https://osm.etsi.org/docs/user-guide/latest/06-osm-platform-configuration.html#osm-repositories

[22] [Online], https://www.innoradar.eu/innovation/47486

[23] [online] APIs: A bridge between mobile operators and start-ups in emerging markets https://www.gsma.com/mobilefordevelopment/wp-content/uploads/2016/07/GSMA_Mobile-operators-start-ups-in-emerging-markets.pdf

[24] 3GPP TS.33.310, “Network Domain Security (NDS); Authentication Framework (AF)”

[25] Jose Ordonez-Lucena, Felix Dsouza, “Pathways towards Network-as-a-Service: the CAMARA project”, NAI '22: Proceedings of the ACM SIGCOMM Workshop on Network-Application Integration, August 2022, Pages 53–59

[26] CAMARA. 2023. Consortium members. Retrieved June 25, 2023, from https://github.com/camaraproject/Governance/blob/main/PARTICIPANTS.MD

[27] CAMARA. 2023. Authentication and Authorization Concept for Service APIs, Retrieved June 25, 2023, from https://github.com/camaraproject/WorkingGroups/

[28] [TMF IF1167] TM Forum, “TMF IF1167 – ODA Functional Architecture”, 2020

[29] [GB999] TM Forum, “GB999 – ODA Production Implementation Guidelines”

[30] [TMF 638] TM Forum, “TMF 638 - Service Inventory Management”

[31] [TMF909A] TM Forum, “TMF909A – API Suite Specification for NaaS”

[32] [online] https://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/021/02.06.01_60/gs_NFV-SEC021v020601p.pdf

[33] [online] https://www.gsma.com/futurenetworks/operator-platform-hp/

[34] [online] https://www.linuxfoundation.org/

[35] [online] https://www.6gworld.com/6gsymposium-spring-2023/

[36] [online] https://www.etsi.org/events/2154-evolving-nfv-towards-the-next-decade

[37] [online] https://tmt.knect365.com/6g-summit/

[38] Deliverable 1.7: Architecture design and technical specifications - Initial, 5GMediaHUB.

[39] ETSI GR NFV-IFA 029 V3.3.1 (2019-11). [Online]. Available: https://www.etsi.org/deliver/etsi_gr/NFVIFA/001_099/029/03.03.01_60/gr_NFV-IFA029v030301p.pdf

[40] D3.1, 5GASP experimentation services, middleware and multi-domain facilities continuous integration, 2021, [online] https://www.5gasp.eu/assets/documents/deliverables/D3.1%20Experimentation%20Services,%20Middleware%20and%20Multi-Domain%20Facilities%20Continuous%20Integration.pdf

[41] https://www.5g-iana.eu/

[42] https://smart5grid.eu/

[43] https://portal.5gasp.eu/products

[44] 5G-IANA – D6.1 – Market analysis and initial business models, available online at https://zenodo.org/record/7858420#.ZEj3QXZBxD8

[45] Díaz Zayas, Almudena, Giuseppe Caso, Özgü Alay, Pedro Merino, Anna Brunstrom, Dimitris Tsolkas, and Harilaos Koumaras. 2020. "A Modular Experimentation Methodology for 5G Deployments: The 5GENESIS Approach" Sensors 20, no. 22: 6652. https://doi.org/10.3390/s20226652

[46] Koumaras, D. Tsolkas, J. Garcia, D. Artunedo, B. Garcia, R. Marco, A. Salkintzis, D. Fragkos, G. Makropoulos, F. Setaki, A. Diaz, P. Merino, V. Koumaras, P. Encinar, Y. Karadimas, ” A network programmability framework for vertical applications in the beyond 5G era”, 2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), 2022, pp. 375-380, doi: 10.1109/EuCNC/6GSummit54941.2022.9815790.

[47] D. Fragkos, G. Makropoulos, A. Gogos, H. Koumaras and A. Kaloxylos, “NEFSim: An open experimentation framework utilizing 3GPP’s exposure services”, 2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), 2022, pp. 303-308, doi: 10.1109/EuCNC/6GSummit54941.2022.9815829.

[48] https://wiki.evolved-5g.eu/

[49] https://forum.evolved-5g.eu/t/welcome-to-the-evolved-5g-forum/14

[50] https://forum.evolved-5g.eu/c/library/7

13 List of Contributors

Name	Company / Institute / University	Country	Project
Editorial Team			
Overall Editor			
Bessem Sayadi	NOKIA Bell-Labs 5G-PPP Software Network WG Chairman	France	
Contributors			
Chia-Yu Chang	NOKIA Bell-Labs	Belgium	DAEMON
Christos Tranoris	University of Patras	Greece	5GASP
Marius Iordache	Orange	Romania	VITAL-5G
Eirini Liotou	Institute of Communication and Computer Systems	Greece	5G-IANA
Matteo Andolfi	Nextworks	Italy	5G-IANA
Theodoros Rokkas	INCITES Consulting	Luxembourg	5G-IANA
Hamzeh Khalili	CTTC	Spain	5G-MediaHUB
Ioannis Tomkos	University of Patras	GR	5G-Induce
Nikolaos Kanakaris	University of Patras	GR	5G-Induce
Josep Martrat	ATOS	Spain	Affordable5G
Thanos Xirofotos	UBITECH	Greece	5GIANA, 5GINDUCE
Qi Wang	University of the West of Scotland	UK	5GINDUCE
Jose M. Alcaraz Calero	University of the West of Scotland	UK	5GINDUCE
Dimitrios Klonidis	UBITECH	GR	5GINDUCE
Guillermo Gomez	ATOS	Spain	Smart5Grid
Andres Cardenas Cordova	I2CAT	Spain	Smart5Grid
Regel G. Usach	Polytechnic University of Valencia	Spain	EVOLVED-5G
David Artunedo	Telefonica	Spain	EVOLVED-5G
Harilaos Koumaras	NCSR “Demokritos”	Greece	EVOLVED-5G
George Makropoulos	NCSR “Demokritos”	Greece	EVOLVED-5G
Dimitris Tsolkas	Fogus Innovations & Services P.C.	Greece	EVOLVED-5G

Acknowledgment

We would like to thank all the project contributors that are indirectly involved in this White Paper and not cited directly in the list above.