5Genesis

5TH GENERATION END-TO-END NETWORK, EXPERIMENTATION, SYSTEM INTEGRATION, AND SHOWCASING

[H2020 - Grant Agreement No. 815178]

Deliverable D2.4

Final report on facility design and experimentation planning

Editor A. Díaz Zayas (UMA)

Contributors UMA (UNIVERSIDAD DE MALAGA), LMI (L.M. ERICSSON LIMITED), COS (COSMOTE), NCSRD (NATIONAL CENTER FOR SCIENTIFIC RESEARCH "DEMOKRITOS"), ATH (ATHONET SRL), ECM (EURECOM), KAU (KARLSTADS UNIVERSITET), SRL (SIMULA METROPOLITAN CENTER FOR DIGITAL ENGINEERING), ATOS (ATOS SPAIN SLC), SHC (Space Hellas (Cyprus) Ltd.), FhG (FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V), IHP (IHP GmBH), AVA (AVANTI HYLAS 2 CYPRUS LIMITED), REL (RunEL), TID (Telefonica I+D), FOG (FOGUS), INF (INFOLYSIS)

Version 1.0

Date July 23st, 2020

Distribution PUBLIC (PU)

List of Authors

UMA	UNIVERSIDAD DE MALAGA
A. Díaz Zayas, P. Merino, L. Panizo	
LMI	L.M. ERICSSON LIMITED
A.-M. Bosneag, E. Aumayr	
ATOS	ATOS SPAIN SA
E. Jimeno, J. Melian, Luis Gomez	
FhG	FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V.
M. Emmelmann, S.K. Rajaguru, A. Prakash	
ECM	EURECOM
P. Matzakos	
IHP	IHP GMBH - INNOVATIONS FOR HIGH PERFORMANCE MICROELECTRONICS/LEIBNIZ-INSTITUT FUER INNOVATIVE MIKROELEKTRONIK
J. Gutierrez	
SRL	SIMULA RESEARCH LABORATORY AS
O. Alay, G. Caso	
ATH	ATHONET SRL
F. Giust, D. Munaretto	
REL	RUNEL NGMT LTD
I.Koffman	
AVA	AVANTI HYLAS 2 CYPRUS LIMITED
A. Perentos, E. Hadjioannou, S. Watts	
KAU	KARLSTADS UNIVERSITET
A. Brunstrom	

SHC	SPACE HELLAS (CYPRUS) LTD
G. Gardikis	
TID	Telefonica I+D
D. Artuñedo, A. Flores	
NCSRD	National Center for Scientific Research "DEMOKRITOS"
H. Koumaras, G. Xilouris, T. Anagnostopoulos	
COS	COSMOTE
F. Setaki	
REL	RunEL
I. Koffman	
FOG	Fogus Innovations & Services P.C.
D. Tsolkas	
INF	INFOLYSiS PC
V. Koumaras	

Disclaimer

Copyright

Version History

Rev. N	Description	Author	Date
1.0	Release of D2.4	A. Díaz Zayas (UMA)	20/7/2020

LIST OF ACRONYMS

Acronym	Meaning
3GPP	3rd Generation Partnership Project
5G-PPP	5G Public-Private Partnership
ADB	Android Debug Bridge
API	Application Programming Interface
CLI	Command-Line Interface
CORD	Central Office Re-architected as a Datacenter
CQI	Channel Quality Indicator
EC	European Commission
ELCM	Experiment Life Cycle Manager
eMBB	Enhanced Mobile Broadband
EMS	Element Management System
GA	Grant Agreement
ICT	Information and Communication Technology
IM	Infrastructure Monitoring
IoT	Internet of Things
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LTE	Long Term Evolution
M&A	Monitoring and Analytics
MANO	Management & Orchestration
MCPTT	Mission Critical Push-To-Talk
MCS	Mission Critical Services
ML	Machine Learning
NBI	Northbound Interface
NEST	Network Slice Template
NFVO	Network Function Virtualization Orchestrator
NMS	Network Management System
NS	Network Service
NSA	Non-Stand-Alone
NSD	Network Service Descriptor
NSR	Network Service Record

OAI	OpenAirInterface
PM	Performance Monitoring
PSC	Primary Synchronization Code
RAN	Radio Access Network
REST	Representational State Transfer
RRM	Radio Resource Management
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
RSSI	Received Signal Strength Indicator
SA	Stand-Alone
SBI	Southbound Interface
SDN	Software Defined Network
SLM	Slicing Lifecycle Manager
SNR	Signal to Noise Ratio
SUT	System Under Test
TAP	Test Automation Platform
TS	Technical Specification
UE	User Equipment
UI	User Interface
URLLC	Ultra-Reliable Low Latency Communication
VIM	Virtual Infrastructure Manager
VM	Virtual Machine
VNF	Virtual Network Function
VNFD	Virtual Network Function Descriptor
VNFR	Virtual Network Function Record
WIM	WAN Infrastructure Manager
WP	Work Package

LIST OF FIGURES

LIST OF FIGURES

Executive Summary

This deliverable provides a complete view of the final 5GENESIS architecture, covering all the components of the experimentation chain, from the experiment control tools to the infrastructure. This deliverable can also serve as a reference point for verticals that want to use the 5GENESIS facility for experimentation purposes.

In particular, the D2.4 deliverable describes the final 5GENESIS architecture, based on the initial one, introduced in D2.2, incorporating feedback, insights, and directions received from the activities carried out in WP3, WP4 and WP6. The requirements identified in D2.2 have been refined in accordance with work done during the first two years of the project. The 5GENESIS experimentation methodology introduced in D2.3 has been also modified and the changes that applied are explained in this deliverable. Finally, the execution of distributed experiments, introduced in D2.2, has been completed in the present deliverable.

Table of Contents

1. INTRODUCTION

The purpose of this deliverable is to present in one concise document the final 5GENESIS architecture and experimentation methodology. As such, it presents the work undertaken as part of WP2 “Facility Requirements and Specifications”, Tasks T2.1, T2.2 and T2.3. The content of this deliverable represents our final view on requirements, architecture and experimentation methodology, and therefore supersedes any architecture specification and experimentation methodology content presented in our previous WP2 deliverables (D2.2 [1] and D2.3 [3], respectively).

The work performed in WP2 and included in this deliverable forms the basis for the implementations of the common architectural components in WP3, and for the deployments, testing and KPI validation work done in WPs 4, 5, and 6, while the inputs towards 5G-PPP groups are taken further by WP7.

Through a continuous feedback loop between WP2 and the implementation, deployment and testing work packages (WP3, WP4, WP5 and WP6), our view on the architecture and methodology was enhanced. This process included:

- enhancing the work done on requirements,
- adding new components such as the Slice Optimisation Module,
- specifying the east/west interface through detailing multi-platform experiment workflows,
- update the 5GENESIS reference architecture,
- refine the 5GENESIS experimentation methodology.

The key target of the 5GENESIS Experimentation Platform is to support 5G experimentation and our architecture design was conducted from this perspective. While the design accommodates for a diverse set of capabilities in each of the underlying platforms of 5GENESIS, including technological advancements in spectrum management and multi-domain end-to-end orchestration, as well as interactions between different administration and experimentation domains, it is also based on one common slicing and experimentation model, allowing for an easy interface and experimentation procedure for all experimenters. Our developed experimentation methodology is now incorporated into the 5G-PPP TMV group’s recommendations. The design and development from this perspective facilitates the validation of 5G capabilities and of the various KPIs for different vertical services run on top of our 5GENESIS Experimentation Platform, while being aligned with relevant 5G standards.

The deliverable is organised as follows:

- Section 2 includes the refinements to requirements, incorporating the feedback from WP3, WP4 & WP5.
- Section 3 presents all the components of the 5GENESIS architecture. It is structured by the three layers in our blueprint architecture (Coordination, Management & Orchestration, and Infrastructure), explaining the components of each layer, their functionality, and interactions between components and layers. This Section also explains which enhancements have been done over the architecture presented in D2.2 [2].
- Section 4 gives an overview of our experimentation methodology, including the improvements done since D2.3 [3].
- Section 5 details the east/west interface between platforms and explains how multi-platform experiments take place in 5GENESIS.

2. 5GENESIS REQUIREMENTS

The "Genesis of 5G" has entered the crucial phase of experimentation, and it faces the challenge to validate the 5G network KPIs and verify the 5G technologies with an end-to-end approach as part of 5G PPP Phase 2 and 3. 5GENESIS brings the experimentation perspective in the picture, targeting to facilitate the deployment and execution of the vertical industries' services with real-users, through the proposition of a homogeneous interface that can be supported by the various platforms. Such deployment, execution, and validation, is called an **Experiment** in the context of the project and assumes the following roles:

- **Experimenter**: Executes the experiments on behalf of the vertical industry using the experimenter interface provided by the 5GENESIS platforms. Typically, the experimenter acts also as the integrator of the vertical technologies into the service to be validated on 5GENESIS;
- **Platform Operator**: Hosts, manages and operates the platform's software and infrastructure. He is also the interface to the experimenters, and in charge of the telecommunications infrastructure, as well as the coordination, management, orchestration and monitoring systems;
- **Platform Technology Provider**: This role is acquired by vendors that support the testbed infrastructure with hardware and software components. Vendors are considered partners from the industry and even open source communities and research institutions that provide components to the 5GENESIS platforms;
- **Testers and End Users**: The users of the services deployed in the 5GENESIS platforms by the Experimenter. They can be either individuals or corporate end-users.

Building on top the 3GPP 5G architecture specifications [1], 5GENESIS proposes a 5G architecture blueprint to serve as a common architectural reference, including an openness framework, with APIs for exposing the Experimentation Framework to verticals. 5GENESIS implements an Experimentation Framework blueprint, to be implemented over platforms distributed across Europe in five cities, most of them already identified by the 5G PPP as "5G Trials Cities" [12], namely Athens, Malaga, Berlin, Limassol and Surrey.

The guiding principles, upon which the project concepts are built, are summarised below:

- 5GENESIS is comprised of various geographically dispersed platforms;
- The platforms are complementary in terms of features, nevertheless aligned to the proposed common reference architecture;
- The platforms are administratively independent, exposing open interfaces for inter-platform coordination and verticals experimentation;
- The platforms accommodate multiple experiments from various verticals with diverse requirements;

Each one of the 5GENESIS platforms has the flexibility to govern its own physical topology, architecture and particular technological features. Nevertheless, towards harmonizing the different platforms and experimentation infrastructures, elements of the common reference architecture are to be replicated across the five platforms. In this way, each platform can be administratively independent, yet interoperable with the other platforms. To implement these principles the project foresees three '**Reference Layers**' that are analysed and specified as part of the 5GENESIS architecture design:

- **Platform Coordination layer:** The Coordination layer has the primary role of interfacing with the experimenters, performing the coordination of the platform, achieving overall supervision and end-to-end configuration and execution of the experiments. This is the layer where common implementation and strict conformance among the platforms is pursuit to ensure seamless service towards the experimenters and becomes the focus of the project's main development efforts.
- **Management and Orchestration (MANO) layer:** This layer involves all the management and orchestration capabilities that are necessary to deploy for running the experiment. This layer needs to be built in accordance to the project's design, but there is no strict association with specific implementations. Even though the platforms have mainly opted for the same open source solutions, the design is open to encapsulate other solutions as well, through the development of custom plugins that realise the appropriate interactions with infrastructure layer.
- **Infrastructure layer**: It is the end-to-end network including the Mobile Core Network, Access networks (including user equipment), Transport Networks and Virtualisation Infrastructure. This layer is completely platform-dependant, and diversity among platform capabilities is encouraged.

It is noteworthy that most of the experiments are expected to be fully satisfied within a single platform realm, as they are likely to be dependant either on the capabilities match with the vertical industries requirements or location-specific characteristics. The Experimenter Interface, based on an openness framework of the project, guarantees similarity across platforms in the definition of the experiment, the description of necessary resources -such as the slice (including both the virtual network functions (VNFs) and the 3GPP features of the slice) and the computational resources to be initially assigned-, as well as the test cases and metrics to be used and collected for the validation, allowing for repetition of the experiment in various platforms easily and in a transparent manner. Moreover, the project could also supports interconnection between platforms to run experiments in a distributed way. To achieve that, the Coordination layer plays the major role for realising a distributed experiment.

Putting all these into consideration, the D2.1 deliverable [4] was published during the initial phase of the project and includes an extensive list of Requirements relevant for the initial design and implementation work. As part of the final design, the initial requirements have been refined, and become more concise, reflecting appropriately the work carried out.

A recapitulated presentation of the driving requirements is provided below. It contains the functional requirements that have impacted the design work presented in the following sections and is organised per architectural layer. As the design work has already matured to real implementations, the description of each requirement is associated with the related component(s) of the initial specification architecture as described in Deliverable 2.2 [1]. Even more, understanding that there are several stakeholders involved with disparate demands the description points also to the role that shall leverage the capability built.

2.1. Coordination Layer Requirements

The functional requirements that address the 5GENESIS coordination layer, including experimenter's interface, end-to-end experiment lifecycle management and end-user's management and monitoring, are included in Table 1.

Table 1: Coordination Layer Requirements

COORD-1	Open APIs towards the Experimenter		
Priority	Essential		
Description	The Coordination layer **shall** expose open APIs enabling the Experimenter to access the platform, define and conduct experiments as well as retrieve the results		
Affected Component(s)	Dispatcher, ELCM Portal	**Role/Stakeholder**	Experimenter
COORD-2	Platform registry		
Priority	Essential		
Description	The Coordination layer **shall** provide information about the experimental capabilities per platform, for example, the test cases supported and a description of the platform components.		
Affected Component(s)	Dispatcher, ELCM, Portal	**Role/Stakeholder**	Experimenter, Platform Owner
COORD-3	Experiment Definition		
Priority	Essential		
Description	The Coordination layer **shall** define an experiment descriptor template in order to allow experimenters to describe their experiments		
Affected Component(s)	Dispatcher, ELCM, Portal	**Role/Stakeholder**	Experimenter
COORD-4	Experiment Validation		
Priority	Essential		
Description	A systematic process and methodology verifying compliance and availability of the network services and test cases shall be implemented, allowing for experiment execution		
Affected Component(s)	Dispatcher, ELCM	**Role/Stakeholder**	Platform Owner
COORD-5	Experiment Lifecycle Management		
Priority	Essential		
Description	The Coordination layer **shall** provide control over the transition between experimentation lifecycle stages (i.e. start, stop, etc.) and of the processing at each stage.		
Affected Component(s)	Dispatcher, ELCM, Portal	**Role/Stakeholder**	Experimenter, Platform Owner
COORD-6	Inter-experiment Coordination/Scheduling		
Priority	Essential		

Description	The Coordination layer **shall** provide scheduling mechanisms for the most effective coordination of queued experiment requests to fulfil the experimenters' schedule efficiently, minimise idle periods and maximise the exploitation of available resources		
Affected Component(s)	ELCM	Role/Stakeholder	Platform Owner, Experimenter
COORD-8	Southbound Control APIs for Experiment Execution		
Priority	Essential		
Description	The coordination layer **shall** provide means for managing and controlling heterogeneous infrastructure elements. The implementation for each element may be enabled via plugins, wrappers or proxies, depending on the interfaces exposed by the element.		
Affected Component(s)	ELCM	Role/Stakeholder	Platform Owner, Technology Provider
COORD-9	KPIs Validation and Evaluation		
Priority	Essential		
Description	The Coordination layer **shall** gather and process all experimental data to calculate and validate the target KPIs, as well as, provide automated reporting to the experimenter		
Affected Component(s)	Analytics, ELCM	Role/Stakeholder	Experimenter
COORD-7	Experiment Monitoring		
Priority	Essential		
Description	The Coordination layer **shall** provide to the experimenter monitoring data during experiment execution.		
Affected Component(s)	Dispatcher, ELCM, Portal, Monitoring Probes	Role/Stakeholder	Experimenter
COORD-8	Access to Raw Experiment Data		
Priority	Essential		
Description	The Coordination layer **shall** support the retrieval of raw experimental data up to the level of granularity supported by the monitoring framework		
Affected Component(s)	Analytics, Dispatcher, Portal	Role/Stakeholder	Experimenter
COORD-9	Vertical Experimenter Dashboard		
Priority	Essential		
Description	The Coordination layer **shall** provide visual representation of the experiment execution results through a graphical user interface		
Affected Component(s)	Analytics, ELCM, Portal	Role/Stakeholder	Experimenter
COORD-10	Experiment Data Storage and Maintenance		
Priority	Essential		

Description	The Coordination layer **shall** provide means for storing and maintaining of experiment data for the time interval negotiated between the Platform owner and the Experimenter		
Affected Component(s)	Main DB	**Role/Stakeholder**	Platform Operator
COORD-11	Experiment Data Privacy		
Priority	Essential		
Description	The Coordination layer **shall** ensure that no experimenter is able to access other experimenters' data		
Affected Component(s)	Dispatcher, Portal	**Role/Stakeholder**	Platform Operator
COORD-12	Profiling Experimentation Configurations		
Priority	Optional		
Description	The Coordination layer **shall** expose to the experimenters predefined options for components' configuration that can be provisioned at a given time for experimentation.		
Affected Component(s)	ELCM, Portal	**Role/Stakeholder**	Platform Operator, Experimenter
COORD-13	Analytics		
Priority	Essential		
Description	The Coordination layer **shall** give to the experimenters the possibility to autonomously execute several types of analysis on their experimental data		
Affected Component(s)	Analytics	**Role/Stakeholder**	Experimenter
COORD-14	Security Analytics		
Priority	Optional		
Description	The platform **shall** provide the capability to perform near-real-time analytics on the network traffic, for the detection and classification of anomalies and/or security incidents. To the full extend, security analytics are necessary for the platform administrators to ensure the soundness of the platform and the confidentiality of the experiment's execution. Nevertheless, basic security and hazard assessment conclusions shall be shared with the experimenter as part of the experiment results		
Affected Component(s)	Security Analytics	**Role/Stakeholder**	Platform Operator
COORD-15	Adaptation for Communication with Management		
Priority	Essential		
Description	The Coordination level **shall** provide appropriate adaptation of the information received by the exposed northbound API (see COORD-1) in order to facilitate communication with the underlying southbound management entities (M&O layer components)		
Affected Component(s)	ELCM	**Role/Stakeholder**	Platform Operator, Technology Providers

Significant effort in the revised final architecture has been put to analyse the impact on Coordination Layer components in respect to supporting Cross-Platform Experimentation. The previously defined general requirement on 'Inter-platform Experimentation' referring to the existence of east-west interface among the platforms is further analysed, as presented in Table 2.

Table 2: Cross-Platform Experimentation Requirements

COORD-19	Cross-Platform Execution Request		
Priority	Essential		
Description	Platform A **shall** be able to request the execution of an experiment to Platform B. Platform B can either accept or reject		
Affected Component(s)	Dispatcher, ELCM	Role/Stakeholder	Platform Operator
COORD-20	Cross-Platform Experiment Customisation		
Priority	Essential		
Description	The experimenter **shall** be able to define a custom cross-platform experiment where he can select the deployment location of his experiments (with platform granularity at minimum)		
Affected Component(s)	Dispatcher, ELCM	Role/Stakeholder	Experimenter
COORD-21	Cross-Platform Execution Synchronisation		
Priority	Essential		
Description	Platform A should be able to ask Platform B the current experiment status on their side of the experiment. For example, current lifecycle stage, current experiment stage, execution errors etc.		
Affected Component(s)	Dispatcher, ELCM	Role/Stakeholder	Platform Operator
COORD-22	Cross-Platform Information Exchange		
Priority	Essential		
Description	Platform A **shall** be able to ask for the value of certain instantiation and provision variables to platform B. For example, the IP of a component deployed in Platform B or a configuration value		
Affected Component(s)	Dispatcher	Role/Stakeholder	Platform Operator
COORD-23	Cross-Platform Results Retrieval		
Priority	Essential		
Description	Platform A shall be able to retrieve all the results generated by Platform B while executing their part of the experiment		
Affected Component(s)	Dispatcher, ELCM	Role/Stakeholder	Platform Operator

Finally, a set of prerequisites necessary to enable the appropriate implementation of the requirements is assumed in place as follows:

- All platforms should conform to their national 5G spectrum regulation and operate on the frequency bands as allocated
- For cross-platform experimentation, the network interconnection between two or more platforms is assumed available and no other actions should be acted upon to establish or configure it.

2.2. Management & Orchestration Layer Requirements

Table 3 summarises the requirements related to the management and orchestration capabilities that are necessary within the platforms to configure and deploy the requested experiments on the underlying supporting network and infrastructure.

Table 3: MANO Layer Requirements

MANO-1	Resource Catalogue per Service		
Priority	Essential		
Description	The M&O layer shall expose an interface to list all resources allocated to a specific VNF service, including the data-centre and edge location of each service and the related infrastructure resources used		
Affected Component(s)	NFV MANO	Role/Stakeholder	Platform Operator
MANO-2	Flexible and Fast Allocation of Network Resources		
Priority	Essential		
Description	The M&O layer shall be flexible and fast in providing requested resources and the whole end-to-end process must be performed in 90 minutes or less		
Affected Component(s)	Slice Manager, NFV MANO	Role/Stakeholder	Platform Operator
MANO-3	Distributed NFVI on User or Service Demand		
Priority	Essential		
Description	The M&O layer shall support multiple NFVI geographically distributed as PoP (Point of Presence) and must be able to instantiate several VNFs per demand - for example to deploy services as close as possible to the end-users for latency optimisations		
Affected Component(s)	Slice Manager, NFV MANO	Role/Stakeholder	Platform Operator
MANO-3	Network Service Composition		
Priority	Essential		
Description	The M&O layer shall be able to deploy a network service that is composed by both virtual (i.e. VNFs) and physical (i.e. PNF) components and provide lifecycle control		
Affected Component(s)	Slice Manager, NFV MANO	Role/Stakeholder	Platform Operator, Technology Provider
MANO-4	Network Slice Definition and Blueprint		
Priority	Essential		
Description	The M&O layer shall support the definition of templates describing the key network slice baseline parameters and resources, oriented to specific		

	vertical use cases. This needs to take into account 3GPP slice definitions and 5G NR configuration capabilities		
Affected Component(s)	Slice Manager	Role/Stakeholder	Platform Operator, Technology Provider
MANO-5	Slice Management		
Priority	Essential		
Description	The M&O layer shall provide management and operation of network slice creation across technological domains (i.e. Computing, Network and Radio) based on the provided and exposed infrastructure elements capabilities		
Affected Component(s)	Slice Manager, EMS	Role/Stakeholder	Platform Operator, Technology Provider
MANO-6	Slice Isolation		
Priority	Essential		
Description	The M&O layer shall support resource isolation between resources allocated to concurrent network slices. This requirement depends on the isolation capabilities supported at the infrastructure elements and controllers		
Affected Component(s)	Slice Manager	Role/Stakeholder	Platform Operator, Technology Provider
MANO-7	Coexistence of Multiple Network Slices and/or Services		
Priority	Essential		
Description	The M&O layer shall allow the co-existence of multiple slices running concurrently over the same infrastructure		
Affected Component(s)	Slice Manager	Role/Stakeholder	Platform Operator, Technology Provider
MANO-8	Network Slice Support for User Equipment (UE)		
Priority	Essential		
Description	The M&O layer shall allow the platform operator to configure the information which associates a UE to a network slice and a service to a network slice. Based on the subscription, UE capabilities, the access technology being used by the UE, operator's policies and services provided by the network slice, the M&O layer shall also be able to move a UE from one network slice to another, as well as to remove a UE from a network slice		
Affected Component(s)	Slice Manager	Role/Stakeholder	Platform Operator, Technology Provider
MANO-9	NFV Management and Organisation		
Priority	Essential		
Description	The M&O layer shall provide means to orchestrate and manage deployment and operation of NFV Network Services on top of virtualisation capable infrastructures. The solution must follow the recent specification as laid out by ETSI NFV Industry Specification Group (NFV ISG) [20] .		
Affected Component(s)	NFV MANO	Role/Stakeholder	Platform Operator, Technology Provider
MANO-10	MEC Management and Organisation		

Priority	Optional		
Description	M&O layer may provide means to orchestrate and manage MEC applications deployment and operation on MEC enabled infrastructures at the network edge.		
Affected Component(s)	Slice Manager, NFV MANO	Role/Stakeholder	Platform Operator, Technology Provider
MANO-11	Real-time Network Monitoring		
Priority	Essential		
Description	The M&O layer shall manage and gather monitoring information across all available platform technological domains (NFV, RAN, WAN) and components (eNB, DRAN etc.) in real-time to facilitate the KPI validation objectives		
Affected Component(s)	NMS, EMS	Role/Stakeholder	Platform Operator, Technology Provider
MANO-12	Quality of Service (QoS) Mapping		
Priority	Essential		
Description	The M&O layer may be able to map quality of service levels to specific configurations within the network slice		
Affected Component(s)	NMS, EMS	Role/Stakeholder	Platform Operator, Technology Provider
MANO-13	Cross-platform Network Slicing		
Priority	Essential		
Description	Slice Managers must be responsible for the operations within their administrative domain and will not directly control or instantiate slices. Hence a Slice Manager - Slice Manager API should be available for slice operations and NS deployments		
Affected Component(s)	Slice Manager	Role/Stakeholder	Platform Operator, Technology Provider
MANO-14	Dynamic slice Optimisations		
Priority	Essential		
Description	The M&O layer shall provide support for policy management and near-real-time analytics for dynamic slice optimisations		
Affected Component(s)	Slice Manager	Role/Stakeholder	Platform Operator, Technology Provider

2.3. Infrastructure Layer Requirements

The requirements that are relevant to the Infrastructure layer components, focusing on the core network and NFVI, the backhaul network, the mobile edge platform and the radio heads, as well as the end-user equipment are presented in the Table 14.

Table 4: Infrastructure Layer Requirements

INFRA-1	4G and 5G RAN and Core Coexistence/Backward Compatibility
Priority	Essential

Description	The platform shall provide 4G and 5G RAN and core components to experiment with use cases involving inter-RAT mobility, aggregation of technologies, non-standalone mode, and other interworking capabilities		
Affected Component(s)	RAN	Role/Stakeholder	Experimenter, Platform Operator, Technology Provider
INFRA-2	5G Mode of Operation		
Priority	Essential		
Description	The 5G system (core network, RAN, and UE) shall support non-stand alone and standalone modes of 5G operation		
Affected Component(s)	RAN, Core NFVI	Role/Stakeholder	Experimenter, Platform Operator, Technology Provider
INFRA-3	Availability of User Equipment		
Priority	Essential		
Description	The platforms **shall** support commercial and experimental UEs for use as part of the experiments		
Affected Component(s)	User Equipment	Role/Stakeholder	Experimenter, Platform Operator
INFRA-4	Integration of Measurement Probes for Experiments and KPI Validation		
Priority	Essential		
Description	The Infrastructure layer **shall** provide means to instantiate and manage measurement probes as part of the end to end service deployment. The probes may allow realisation of different traffic scenarios and APIs to collect the relevant metrics for KPI validation.		
Affected Component(s)	Core/Edge NFVI, Probes	Role/Stakeholder	Platform Operator
INFRA-5	Infrastructure Control Plane		
Priority	Essential		
Description	The Infrastructure layer **shall** have a well-defined control plane and control APIs per element in order to facilitate the integration of new hardware and software components and ensure the sustainability of the platform. The control APIs would depend on the typology of the element: RAN, core network, transport network, etc.		
Affected Component(s)	Core/Edge NFVI, RAT, Transport	Role/Stakeholder	Technology Provider
INFRA-6	Resource Isolation for Slicing		
Priority	Essential		
Description	The Infrastructure layer shall provide means for resource sharing and multiple accesses support. In addition, it shall provide appropriate mechanisms for isolation of resource usage where applicable		
Affected Component(s)	Core/Edge NFVI, RAT, Transport	Role/Stakeholder	Technology Provider
INFRA-7	Virtualised Computing Environment		
Priority	Essential		

Description	The Infrastructure layer **shall** be based on virtualised computing infrastructure to support function virtualisation via virtual machines or containers		
Affected Component(s)	Core/Edge NFVI	Role/Stakeholder	Platform Operator
INFRA-8	Virtualisation Infrastructure Management		
Priority	Essential		
Description	Each virtualised infrastructure used for NFV **shall** expose interfaces and APIs to enable the resource management and their orchestration by the M&O layer		
Affected Component(s)	Core/Edge NFVI	Role/Stakeholder	Platform Operator
INFRA-9	MEC Infrastructure Deployment and Integration		
Priority	Essential		
Description	Platforms shall provide computing and networking capabilities for realising MEC capable nodes. The MEC nodes can either be based on available NFVI deployments or can be MEC specific		
Affected Component(s)	Core/Edge NFVI	Role/Stakeholder	Platform Operator
INFRA-10	Integration of 3rd Party Equipment		
Priority	Essential		
Description	The platform may support the integration of a virtual or physical infrastructure component brought by the experimenter in order to conduct the relevant experimentation and validate KPI related objectives. This may be permitted under certain conditions and safety regulations and is strongly dependent on the actual functionality provided by the component. For example, a new radio component that uses frequency that is not licenced cannot be allowed to be deployed		
Affected Component(s)	Core/Edge NFVI, RAT	Role/Stakeholder	Experimenter, Platform Operator
INFRA-12	Radio Frequency Allocation		
Priority	Essential		
Description	The Infrastructure layer must allocate the radio-frequency spectrum required to achieve 5G throughput end-to-end		
Affected Component(s)	RAT	Role/Stakeholder	Experimenter, Platform Operator
INFRA-13	5G Deployment		
Priority	Essential		
Description	The 5G RAN and UE shall support Release 15 and beyond		
Affected Component(s)	RAT	Role/Stakeholder	Experimenter, Platform Operator
INFRA-14	QoS Management Interface		
Priority	Essential		

Description	The Infrastructure layer shall provide an interface for the vertical applications to be able to control the quality of service for the execution of the experiment		
Affected Component(s)	Core/Edge NFVI	Role/Stakeholder	Technology Provider, Platform Operator
INFRA-15	Support Bearer Priority Set-up		
Priority	Essential		
Description	The Infrastructure layer shall provide the capability for pre-emption of lower priority flows and the vertical applications must be able to request the priority of a flow over another one		
Affected Component(s)	Core/Edge NFVI	Role/Stakeholder	Technology Provider, Platform Operator
INFRA-16	Support Small Cells and D-RAN		
Priority	Essential		
Description	The Infrastructure layer shall provide both small cell and distributed RAN solutions for gNB		
Affected Component(s)	RAN	Role/Stakeholder	Platform Operator
INFRA-17	Support for Software Defined Networking (SDN)		
Priority	Optional		
Description	The Infrastructure layer shall support SDN to achieve automation and virtualisation and through network programmability provide flexibility for traffic steering per experiment demand		
Affected Component(s)	Transport	Role/Stakeholder	Platform Operator
INFRA-18	Support Wifi RAT		
Priority	Optional		
Description	The Infrastructure layer shall integrate WiFi RAT according to 3GPP standards		
Affected Component(s)	RAT	Role/Stakeholder	Platform Operator
INFRA-19	Provide MCS Location Service		
Priority	Optional		
Description	The Infrastructure layer of a platform may provide 3GPP MCS [21] standard location service capabilities		
Affected Component(s)	Core/Edge NFVI	Role/Stakeholder	Platform Operator

3. 5GENESIS ARCHITECTURE

Figure 1 depicts the refined 5GENESIS architecture, indicating the Open 5GENESIS Experimentation Framework (highlighted in red lines) as well. Here we provide the full list of components and their role in the overall architecture, while implementation details and detailed description of interfaces are provided in the related deliverables on WP3. The Open 5GENESIS Experimentation Framework is fruit of a joint effort of the 5GENESIS consortium. It serves not only as a logical aggregation of architectural components, but also, as a representation of a solid software suite (Open 5GENESIS Suite) that is decoupled from the other components and can be adopted to practically transform any infrastructure to a 5GENESIS experimentation platform. The entire code of the components of this framework is released at GitHub https://github.com/5genesis under the term *Open 5GENESIS Suite.*

The most relevant changes of the architecture are in the coordination layer. In the updated version of the 5GENESIS architecture the Portal is part of the coordination layer. The features of the Portal has been also revisited and new modules have been identified: the Visualizer, which is in charge of representing the outputs provided by the Analytics module, the Experiment progress monitoring, which provides information about the status of the experiment and the Experiment registry, which contains the history of executions of the experiments defined in the Portal.

The Dispatcher is the component that exposes the Open APIs and receives the requests for accessing the testbed and the requests including the services and the experiment descriptors, among others. Due to this, the authentication and the validation modules have been included in the Dispatcher. Note that the Privacy/Security Manager component has been renamed as Authenticator. Moreover, the consortium has identified new features for the Dispatcher, as a result, three new components have been included in its design. There are two components related on the Network Services (NS), the Network Service repository and the MANO wrapper. The MANO wrapper is the component in charge of the communication with the MANO system for the onboarding process of the NS, this connection between the coordination layer and the MANO system was included in the first release of the architecture, now we have included explicitly this module in the Dispatcher. Finally, the Distributor has been included to support the execution of Distributed experiments. The execution of Distributed experiments was included as part of the features supported in the initial Release of the architecture. In this deliverable we have refined the procedure for executing distributed experiments and included this new component. The final workflow for executing distributed experiments is described in Section 5.

The Analytics module has been updated in order to include explicitly the Experiment analytics module and the Security Analytic module. The component Results has been renamed as Main DB and the Experiment Registry has been allocated in the Portal. The Facility registry is called Platform Registry as it includes the description of one particular platform.

In the Management and Orchestration layer, the Slice manager includes a Slice optimization module which enables to monitor the slices and applies different policies depending on the targeted performance. Additional, monitoring probes have also been included in the Slice manager.

Finally, in the Network Management System (NMS) module the management of the resources available in the infrastructure layer have been represented by plugins, which are one of the main contributions of the 5GENESIS project.

The final components of the 5GENESIS architecture are explained in the following subsections.

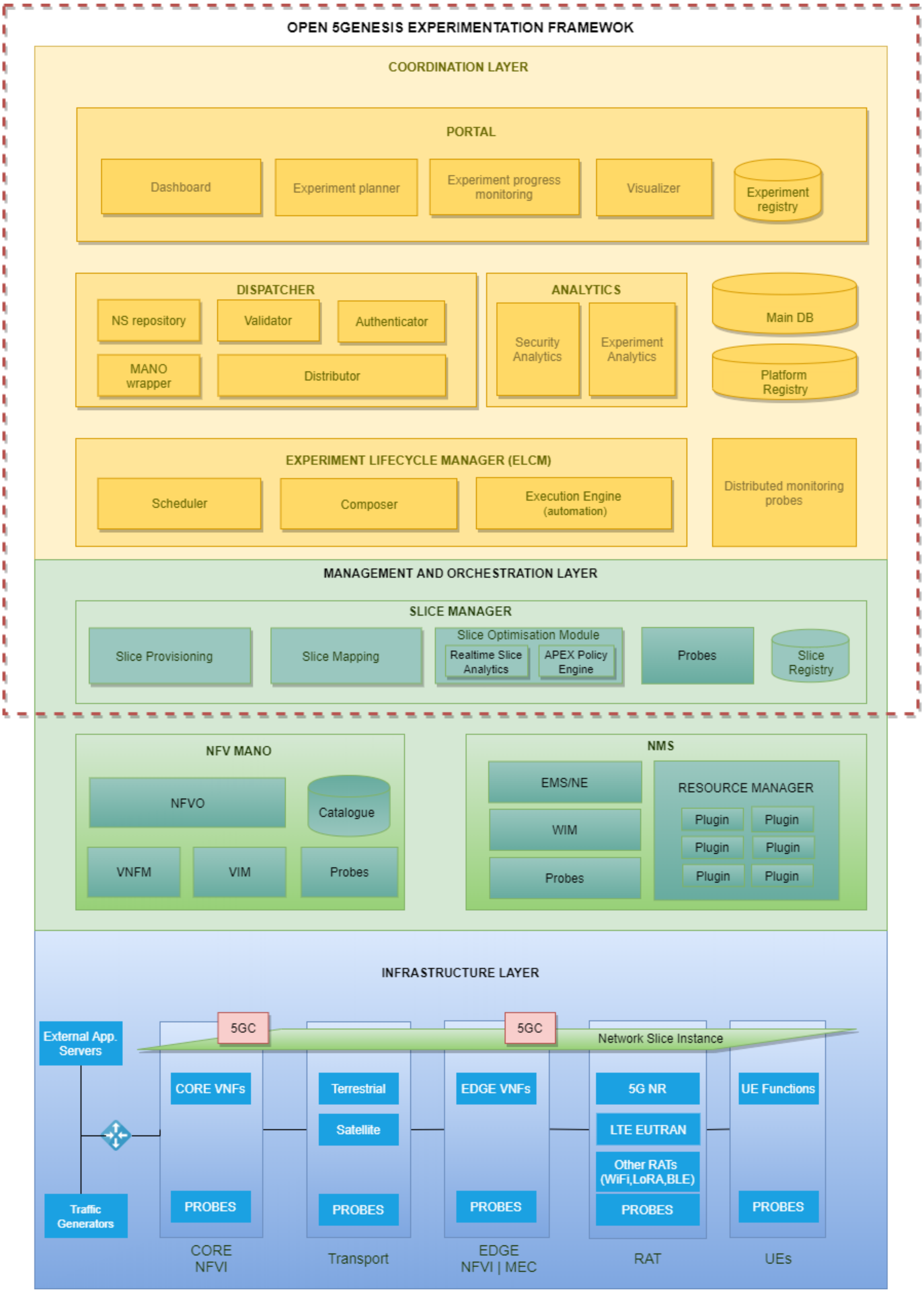

Figure 1 Final 5GENESIS reference architecture

3.1. Coordination layer

The Coordination layer is the layer that coordinates the different components that are integrated as part of the 5GENESIS platform during an experiment execution. It's also the layer through which an experimenter interfaces with the experimentation platform.

This layer contains components that handle separate parts of the coordination, from the interaction with end users to the storage of results. The Portal and Dispatcher provide two different entry points to the platforms, the platform registry and Experiment Lifecycle Manager define and oversee the execution of the different kinds of experiments supported by the platforms, the monitoring probes collect measurements from the different components, which are then used by the Analytics module to extract KPIs and aggregated measurements, and finally the main database provides long term storage of the results.

These components are described in more detail in the following sections.

3.1.1. Portal

The 5GENESIS Portal provides a Web interface that experimenters can use for defining new experiments and request their execution, providing real time information about the execution status, including execution logs. Experimenters can also use the Portal for accessing to the results generated by any previous execution, including the visualization of raw results (based on Grafana dashboards) or the download of such results as CSV files for further processing offline, and access, in general, to advanced features.

Finally, the 5GENESIS Portal provides an interface that can be used for onboarding new Network Services in the platform. After the onboarding process is completed these Network Services can be used as part of an experiment and will be deployed automatically by the Coordination layer.

3.1.2. Dispatcher

The Dispatcher is located in the Coordination Layer (see architecture Figure 1). As explained in D3.7 [5], it is responsible for receiving all the requests sent by the experimenter, either from the Portal or using any other client on top of the Dispatcher NBI. It agglutinates the whole set of the 5GENESIS Open APIs, exposing the appropriate functionalities from the lower layers that are necessary to run the platform securely from the outside. The dispatcher includes several submodules to facilitate the interaction with the underlying modules (see Figure 2).

- The MANO Wrapper (which comprehends also an internal VNF and NS repository,) manages the NFVO and VIMs in the MANO Layer, verifying the consistency among these elements.
- The Distributor is in charge of sending the validated Experiment Descriptor to the ELCM, or, in the case of distributed experiments, sending the experiment descriptors to the platforms involved in the experiment execution.

Since D2.2 [1], the Validator submodule has been moved inside the Dispatcher and split in two: NFV validation and Experiment Descriptor validation in order to avoid unnecessary requests, attaching its functionality to the MANO Wrapper and the Distributor.

The Dispatcher also offers an east-west interface for inter-platform operations, but such interface is not visible to the Experimenter as this interaction is managed by the platforms internally, automatizing this part of the process for the Experimenter.

The Dispatcher ensures that all the requests are coming from a trusted entity before fully processing them, to offer the experimenter all the capabilities of the 5GENESIS system as a single interface with all the available features in a secure way, without the need of knowing particular details of the platform.

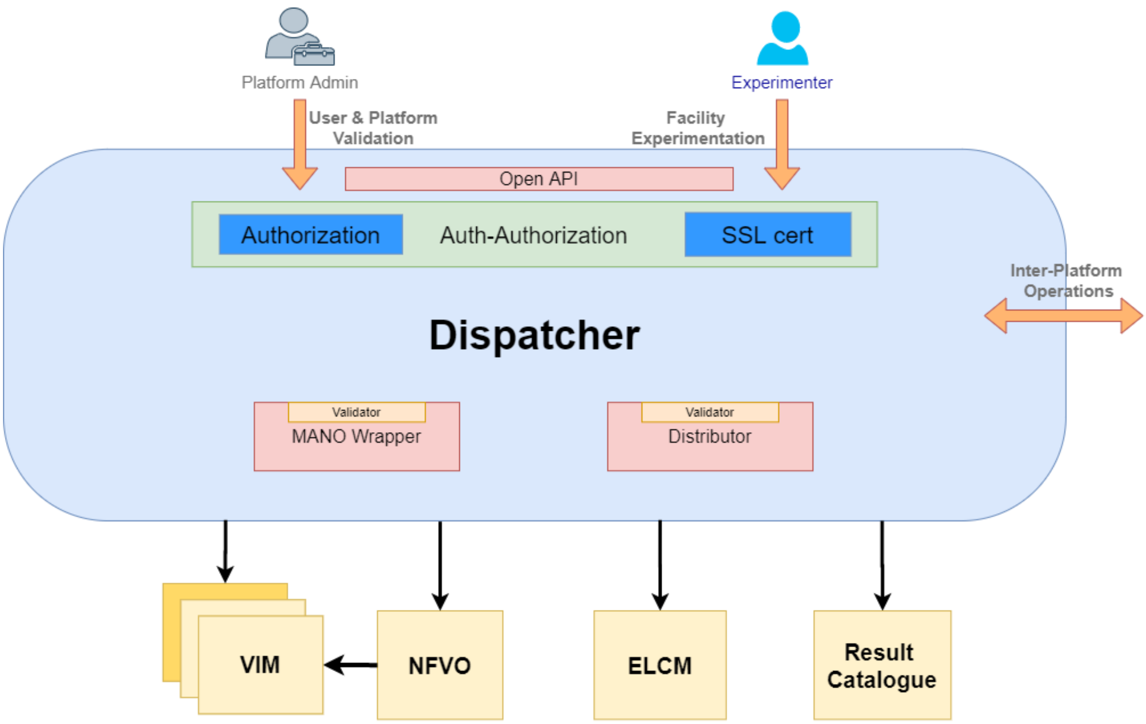

Figure 2 The Dispatcher component of the 5GENESIS coordination layer

3.1.3. Distributed monitoring probes

5GENESIS aims to develop and use a unified Monitoring and Analytics (M&A) framework [6], which includes several distributed monitoring probes. These enable the collection of heterogeneous parameters during the execution of the experiments, across all platforms and under different configurations.

Under the M&A umbrella, the Project has thus deployed and uses several probes, categorizing them in Infrastructure Monitoring (IM) and Performance Monitoring (PM) components. During an experiment execution, IM probes are devoted to the collection of parameters related to the status of architectural components, including end-user devices, radio access and networking systems, computing and storage distributed units. For example, Prometheus is used for monitoring of SDN/NFV instances, as well as RAN and Core/edge units. On parallel, PM probes focus on the active measure of end-to-end Quality of Service (QoS) and Quality of Experience (QoE) Key Performance Indicators (KPI). Here, MONROE VN is for example used as a generic PM tool that can support a variety of PM probes.

The probes interact with the 5GENESIS reference architecture via the ELCM. The ELCM is in charge of activating and configuring them at the beginning of an experiment, and also support the storage of the data they collect in the platform-specific storage utilities.

3.1.4. Main DB

The Main database provides a long-term storage solution for all the measurements generated during an experiment execution. All the values contained in the database are tagged by using a unique execution identifier, making it easy to retrieve all the results from a particular experiment at a later time, in case of multiple experiments.

Within the 5GENESIS reference architecture, the role of the main DB is to store the parameters collected by the distributed probes during an experiment execution.

A platform-specific DB instance is thus active during the platform usage and is loaded with IM and PM data once the experiment execution is terminated. Such operation is regulated by the ELCM, via a so-called Result Listener.

Within the DB, the data are organized in specific tables (e.g., IM and PM tables), in a time series format. Moreover, beside timestamps, the data are augmented by so-called metadata, including Experiment and Iteration IDs, and other parameters summarizing the configurations adopted during the experiment execution (e.g., on which physical hosts some of the virtual functions and services were deployed).

More information on the “Release-A” implementation of the DB as well as the connection with the ELCM are available in Deliverables D3.5 and D.3.15 [6] [7].

3.1.5. Analytics

5GENESIS Analytics is formed by two subcomponents, referred to as Experiment Analytics and Security Analytics.

As regards Experiment Analytics, it aims to provide a full and reliable assessment of 5G KPIs, in the form of a) statistical analysis of the KPIs, as defined in D6.1 [8], and b) Machine Learning (ML)-based analysis. This latter takes as input not only the performance KPIs (collected by the PM probes), but also the infrastructure-related parameters (monitored by the IM probes).

By doing so, Experiment Analytics targets the discovery of correlations and causalities across such parameters, ultimately leading to highlight issues causing performance losses, and possibly trigger improved configurations during next experiments.

A full-chain usage of Experiment Analytics requires the connection to the main DB, in order to query and select the data related to a particular experiment. This step is achieved via a DB-Analytics client, which thus represents the main connection point of Experiment Analytics with the 5GENESIS reference architecture. Once the data are retrieved, they are processed in order to be used for the specific required analysis (e.g., synchronize the data in time and perform a linear correlation analysis). Other functionalities currently include outlier detection, regression, prediction, and feature selection. As regards Security Analytics, the aim is to provide the capability to promptly detect and classify deviations from the normal operation of the

5GENESIS platform, which could possibly point to security incidents. For this reason, a Security Analytics platform is established, based on Big Data and Machine Learning frameworks, which analyses two types of data: i) NetFlow data from key network elements of the platform and ii) infrastructure monitoring data (see above) stemming from the core, edge and access domains. The platform is trained with datasets corresponding to the platform “normal” operation. The primary aim is to detect anomalies. The platform will also provide the means to classify anomalies to different types of security incidents, provided of course that the appropriate labelled datasets will be available for training.

More information on the “Release A” implementation of Experiment Analytics and Security Analytics are given in dedicated Deliverables D3.5 [6] and D3.13 [9].

3.1.6. Experiment lifecycle manager

The Experiment Lifecycle Manager is the entity that oversees the execution of an experiment from start to completion, coordinating the different components in the platform. The ELCM also coordinates the execution of different experiments so that the resources required by each of them are available, allowing the execution of experiments in parallel only if they cannot interfere with each other. This management is performed by the Scheduler sub-component of the ELCM. The Composer is another sub-component that uses the information contained on the Platform registry for generating the set of actions required for an experiment execution. These actions are then performed by the Executor sub-component, which communicates with the rest of the components of the platform when required, for example, by requesting the deployment and decommission of a certain Network Services to the Slice Manager at the start and end of an experiment execution.

3.1.7. Platform registry

The Platform registry is a collection of different configuration files that define the set of available Test Cases, UEs and Network Slices as well as the behaviour of the platform during the execution of an experiment. When an experiment execution request is received, ELCM matches the values requested with the set of configurations described in the Platform registry, generating a collection of actions to perform as part of the execution. The type and order of these actions depend on the selected Test Cases, UEs and Slices selected by the experimenter while defining the experiment, among other configuration parameters.

3.2. Management and orchestration layer

The Management and Orchestration layer of 5GENESIS architecture (Figure 3, has not been modified since D2.2 [1] and consists of three major components, namely:

1. Slice Manager – responsible for the management of the network slices,
2. NFV MANO – responsible for the orchestration of Network Services and lifecycle management of VNFs,

3. Network Management System (NMS) – responsible for the management of non-NFV resources such as transport network (e.g. back/front-haul, WAN) and Mobile Core/Radio elements (i.e. RAT elements and 5G/4G Core Network Functions).

The Management and Orchestration layer communicates with the Coordination layer via two reference points, one that allows the onboarding of the NS descriptors at the NFV MANO Catalogue and another that connects to Slice Manager in order to accept network slice creation configuration and service deployment requests from the ELCM.

A brief summary of the particular functionalities of the aforementioned components is provided in the following section.

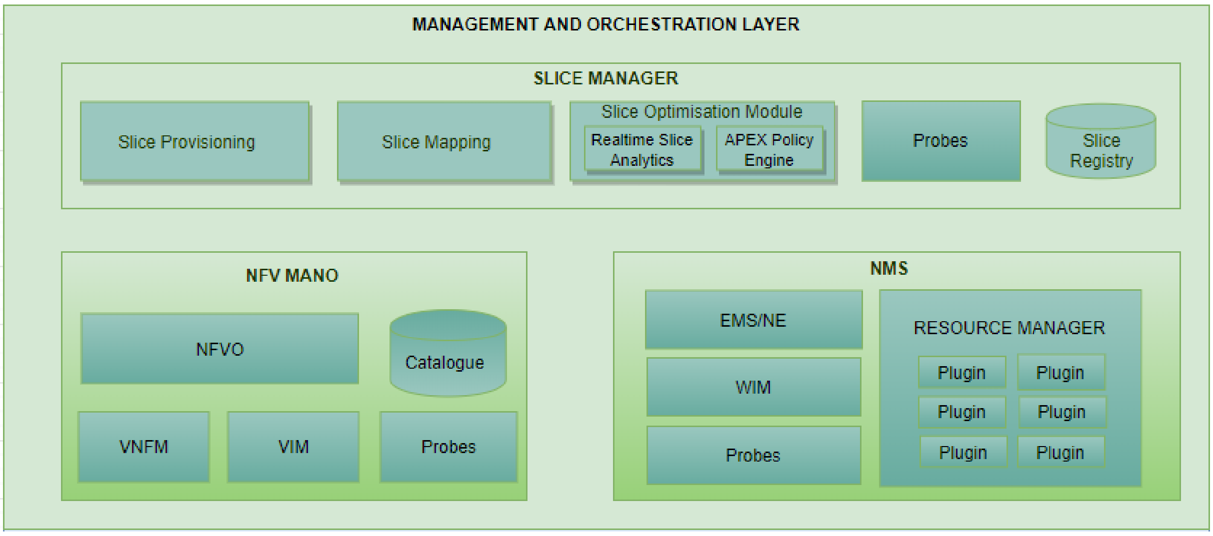

Figure 3 - MANO layer components

3.2.1. Slice manager

As discussed in D2.2 [1], the network slicing concept is one of the most prominent features of the 5G architecture. Slice Manager is a centralized software component that provides an interface for creating, modifying, monitoring, and deleting end-to-end network slices. Through the North Bound Interface (NBI), the Slice Manager interacts with a coordination layer from which receives the Network Slice Template (NEST) for creating network slices and provides the API for managing and monitoring them. Based on the on-boarded NEST, Slice Manager has to make the mapping between the available data plane resources and the described slice requirements, providing the proper placement of each Network Service. Through the South Bound Interface (SBI), Slice Manager talks to the other components of the Management and Orchestration Layer (MANO).

The 5GENESIS Slice Manager is based on a highly modular architecture, built as a mesh of microservices, each of which is running on a docker container. The key advantages of this architectural approach are that it offers simplicity in building and maintaining features, flexibility, and scalability. The Adaptation Layer module provides a level of abstraction regarding the underlaying layer technology, making it feasible for the Slice Manager to operate over any

MANO layer component without any modifications to its core functionality, as long as the proper plugin has been loaded. You can refer to D3.3 [10] for more details regarding the Slice Manager Architecture.

3.2.1.1. Slice Optimisation Module

In order to provide dynamic slice optimization from within the Slice Manager, the APEX Policy Engine and the Realtime Slice Analytics components have been added to the Slice Manager component, as highlighted in color red in Figure 4. Together, they form the Slice Optimisation Module.

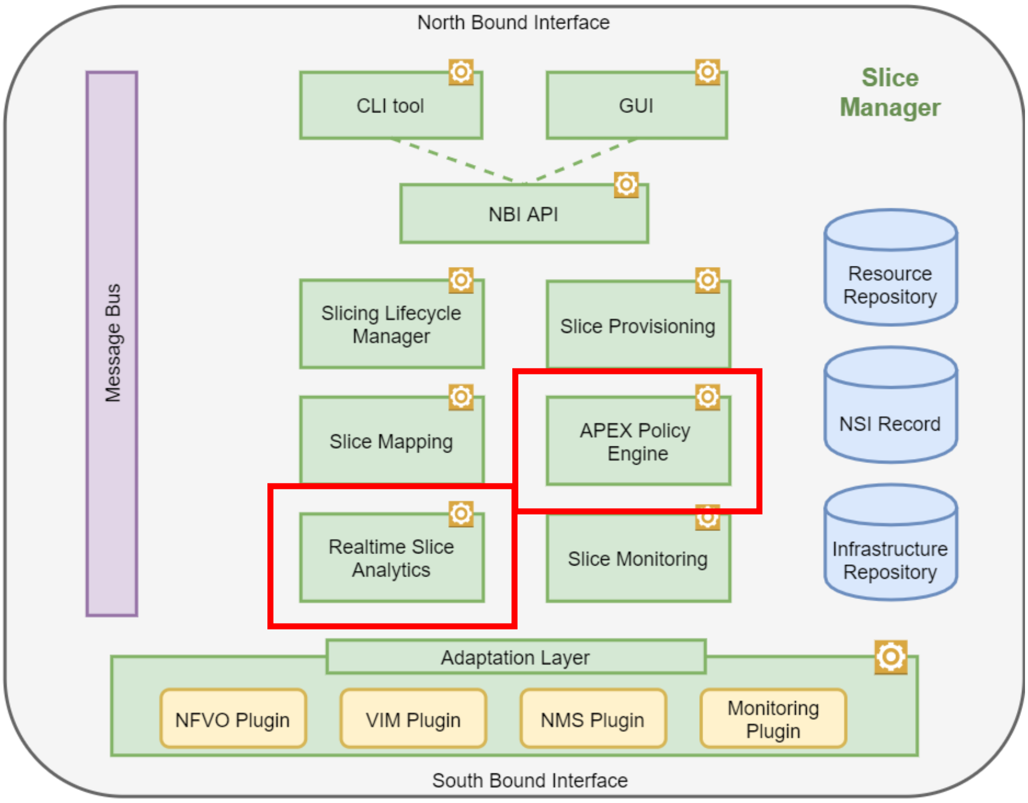

Figure 4 Slice Manager with APEX Policy Engine and Realtime Slice Analytics (highlighted in red) included for Slice Optimisation.

The role of the Realtime Slice Analytics is to process monitoring data from the Slice Monitoring component, in order to detect or predict issues in the current slice setup, e.g., slice SLA violations. Upon detection or prediction of a slice issue, the Realtime Slice Analytics component will trigger the APEX Policy Engine. The role of the APEX Policy Engine is to act upon that trigger and run specifically designed policies to remedy the issue, e.g. by sending a reconfiguration request to the Slicing Lifecycle Manager.

Like the other Slice Manager components, the Realtime Slice Analytics and the APEX Policy Engine connect through the Message Bus.

3.2.2. NFV MANO

The NFV MANO component is responsible for the instantiation of Network Services (NSs) within a virtualised infrastructure that support compute, storage and network virtualisation (NFVI).

A specific interface is exposed northbound in order to accept the requests, for NSs that have been previously onboarded in the NS catalogue. The NFV MANO comprises the following components:

- NFV Orchestrator (NFVO) - Responsible for performing the orchestration of NFVI resources across multiple VIMs with the help of the WIM component when necessary, as well as for managing the lifecycle of Network Services.
- VNF Manager (VNFM) - Responsible for the lifecycle management of VNFs under the control of the NFVO, which it achieved by sending instructions to the VIM for instantiation, scaling, updating and upgrading, as well as termination of VNFs. For the 5G architecture, the most critical issue is the interface with the EMS for each VNF/PNF.
- Virtualisation Infrastructure Manager (VIM) - Responsible for managing the virtualized infrastructure of an NFV-based solution (including keeping inventory of the allocation of virtual resources to physical resources, supporting the management of VNF forwarding graphs, managing a repository of NFVI hardware and software resources, supporting discovery of the capabilities and features to optimize the use of such resources, performing the necessary LCM operations triggered by the VNFM). This component is actually between the M&O and the Infrastructure layer. The interfaces to the infrastructure elements (i.e. NFVI, WAN) are well defined in [22].
- NS repository: The NS Repository is a distributed Network Service version control compliant to ETSI OSM 7.1. This repository of services is a solution to many problems, from uncouple the NS catalogue from the NFVO, to decentralize the network services database to be used by different NFVO at the same time.
- MANO Wrapper: This coordination layer component is in charge of manage the NS repository. It is the responsible of validate the NS packages, check their dependencies and indexing in the catalogue. Also, It is in charge of upload the VIM images, that will be required to instantiate the NS in the NFVO.

3.2.3. NMS

Network Management System (NMS) is a platform-specific entity, which is responsible for the management and monitoring of the transport network elements (e.g. SDN and SD-WAN elements, routers, firewalls, etc.) and Mobile Core/Radio elements (e.g. RAT elements and 5G/4G Core Network Functions). In each platform, NMS implements an application or a set of applications that allow network administrators to manage and monitor both software and hardware components that are part of the platform's infrastructure layer. In addition to that, NMS communicates with the North Bound components of the 5GENESIS architecture in order to allow the creation of end-to-end network slices across the platform infrastructure and enable network configuration automation.

Several software components have been developed in each platform, implementing part of the functionalities of the NMS, such as the ODL-WIM, which is responsible for configuring the transport network using the OpenDayLight SDN Controller, Amarisoft EMS, which is responsible for configuring Amarisoft physical and virtual instances that are part of the RAN, and Athonet Exporter, which is responsible for exposing useful metrics of the Athonet 5G Core to the infrastructure monitoring framework. Additionally, in order to provide a way for automating the execution of certain actions without the need of creating plugins that are specific to each of the possible testbed components a TAP plugin for controlling devices through SSH has been developed in the context of the project.

The plugin can be used for executing commands in the controlled device, both as a normal user or with administration privileges, and also for transferring files from and to the device using the SCP protocol. Using this functionality, a testbed operator may create a TAP testplan that, for example, initiate the execution of a batch script in the remote machine through SSH, and then retrieves the generated logs or results by using SCP.

In addition, this plugin can be used as a base for dedicated plugins for certain devices, if deemed necessary, easing the development process. You can refer to D5.3 [23] and WP4 deliverables for more details regarding the NMS plugins that have been developed for each platform.

3.3. Infrastructure layer

The 5GENESIS Infrastructure layer refers to all systems of any technology deployed by the platform operators to offer the communications services, including the data centers that host the VNFs that implement the end-services and applications, either in the core or edge sites. Each platform is open to select the appropriate vendor and technology, and, as long as, the proper monitoring probes are installed to support the KPIs calculation in a transparent manner, the platforms unrestrictedly set their own infrastructure architecture and evolution path. The following table, summarises the technologies deployed in the 5GENESIS platforms, and depicts the diversity of systems deployed, that are flexibly incorporated in the overall 5GENESIS experimentation model.

Table 5: Infrastructure Components Technologies

Component	Product/Technology
Data Centre (Main/Edge)	COTS servers SFF x86 PCs
Virtualisation	OpenStack VMware ESXi K8s
EPC/5GC	Athonet vEPC Athonet 5GC Open5GCore 5GIC 4G vEPC 5GIC 5GC NSA ECM OAI Core Polaris’ Rel. 15 EPC solution for Edge NextEPC Amarisoft Rel 15.5 5GC

	Amarisoft Rel 14 4G LTE
5GNR	Prototypes from RunEL and Eurecom Amarisoft NSA/SA Nokia Airscale System and 5G Small Cell (RRH) Huawei 5G Stand Alone Commercial 5G smartphones (Samsung, Xiaomi, Huawei, OnePlus)
LTE EUTRAN	Eurecom OAI / Amarisoft / Nokia AirScale / Nokia FlexiZone Commercial 4G mobile phones (Samsung A40, A90, S6, S9) USB dongles
Non-3GPP Access Networks	WiFi 802.11ac Fixed and wireless IoT devices (LoRa, BLE, PanStam and Arduino for first iteration), INTER-IoT physical/virtual network platform (FIWARE Orion for first iteration). LoRA 7 WiFi (802.11ac)
Probes	MONROE
Traffic Generator	Open-source traffic generators (e.g. Ostinato, Seagull, WARP17, TRex) IxChariot Custom scripts

In the following subsections, the design of the infrastructure components that are developed or expanded as part of 5GENESIS work are presented in more detail.

3.3.1. User equipment

3.3.1.1. UE functions

5GENESIS explores advanced functions on the UE that can be used to enhance end-to-end performance by taking advantage of policy and context information from the 5G network for protocol and algorithm selection. This allows higher layer protocols to be aligned and configured in accordance with the requested services, extending end-to-end slice management to incorporate also higher layer protocol selection and configuration at the UE.

The NEAT system and policy manager is the entity on the UE that is responsible for providing this functionality within 5GENESIS, see also Deliverable D4.11 [13]. NEAT is available as part of the MONROE VN that can be used as a tool to execute generic PM probes, see Section 3.3.1.2. The policy-based protocol selection may involve the following aspects: (i) transport protocol selection and configuration, (ii) selection between IPv6 and IPv4, and (iii) selection of interface to use. In order to perform its function the NEAT policy manger can make use of three inputs: the application requirements, the configured polices and the current network characteristics.

Besides the application requirements that are provided by the application through a TAPS compliant interface, NEAT interfaces with the ELCM and the slice manger to obtain the required information. The policies to use during an experiment can be provided by the experimenter through the portal and is passed to the MONROE VN and NEAT by the ELCM. In order to obtain information on the current network characteristics, the NEAT system makes use of the 5GENESIS slice manager. The slice identifier of the slice used for the experiment as well as the address of the slice manger is provided to MONROE VN and NEAT by the ELCM. This information allows to contact and register a NEAT instance as a policy system at the slice manager. NEAT then regularly polls the slice manager for information about the status and characteristics of the slice.

3.3.1.2. Probes

The probes running on the UE are central for experimentation and KPI validation within 5GENESIS (see also Section 3.2.3). In order to obtain performance measurements from the UEs involved in the execution of an experiment it is necessary to make use of probes. These probes monitor the status of several parameters, and save their values in a format (generally as a time series) that can later be retrieved and analyzed in order to obtain KPIs or study the behavior of the UEs in different conditions.

Several probes are available for deployment and use in the testbeds and generate results in a format that is compatible with other layers of the 5GENESIS architecture. These probes can be classified in the following groups.

(a) MONROE VN

Here, MONROE Virtual Node (VN) has been selected as a platform-agnostic performance monitoring (PM) tool. MONROE VN allows to run MONROE probes designed in the form of Docker containers, thus providing a generic tool for running a wide range of PM probes to capture 5G KPIs and application performance. The probe to run during an experiment and its parameters can be provided by the experimenter through the Portal, which provides a flexible way to run custom experiments in an automated fashion. MONROE probes for latency and throughput KPI validation are part of Release A of the Open 5GENESIS Experimentation Framework. To support controlling MONROE VN through TAP, a TAP agent is embedded into MONROE VN, in order to deploy, start, and post-process the MONROE probes. The TAP agent exposes a REST API that is used by the ELCM to provide to MONROE VN the configurations for the specific probe to run.

(b) Remote PC agents

These agents can be deployed in PCs that act as user equipment, and can be controlled through a network connection by using the exposed REST API. Two agents have been developed as part of the 5GENESIS project, and are currently available for use:

- Remote Ping agent: This agent can be deployed on Linux machines and is able to initiate round trip time test. Generated results can be retrieved via the REST API. The following parameters can be configured: Destination address, interval, packet size and time to live.

- Remote iPerf agent: The iPerf agent can be deployed on Linux and Windows machines and provides remote access to iPerf2. The agent acts as a wrapper for the iPerf executable, providing a way for controlling and retrieving results from a remote machine. For this reason, all configuration parameters that are accepted by the iPerf2 executable can be configured by using the agent.

(c) Android agents

These agents can be installed on Android devices, and can be controlled through ADB. The following agents are available:

- Resources agent: The resources agent can be used for registering the usage of hardware resources of the device (CPU, RAM), the number of packets and bytes transmitted/received and several radio parameters: operator, network, cell ID, LAC, RSSI, PSC, RSRP, SNR, CQI and RSRQ.
- Ping agent: This agent allows the execution of ping tests on Android devices. The available settings are the same as in the case of the Remote Ping agent.
- iPerf agent: As in the case of the Ping agent, this agent is very similar to its PC counterpart. The agent acts as a wrapper to the embedded iPerf2 executable, providing additional functionality for automation and result retrieval.

3.3.2. RAT

3.3.2.1. RunEL

RunEL provides a 5G Infrastructure including 5G New Radio (PHY and MAC) optimized for Ultra Reliable Low Latency Communication (URLLC). RunEL gNB is designed to support advanced features such as: 2 spectrum bands 3.5GHz and 28 GHz, Beam Forming, MIMO, flexible frames, 200MHz BW and more.

The basic architecture of the RunEL solution is depicted in Figure 5.

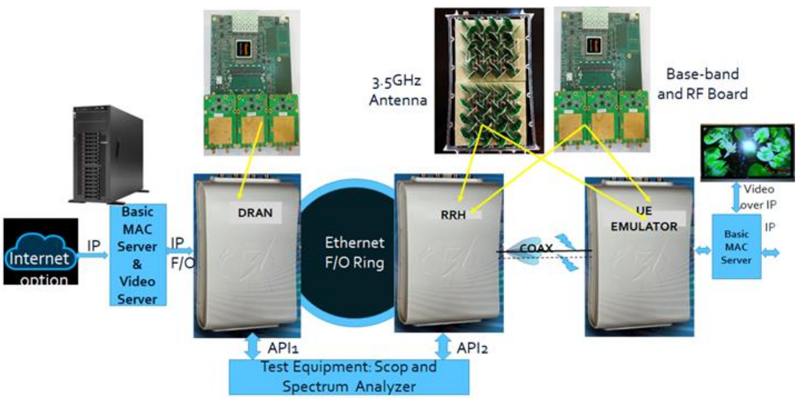

Figure 5 Architecture of the RunEL RAT solution.

The NR fabric includes two units which comprise the physical layer. These include the DRAN (Distributed RAN) unit that includes the High PHY and the RRH (Remote Radio Head) unit that includes the Low PHY as depicted below for Downlink and Uplink.

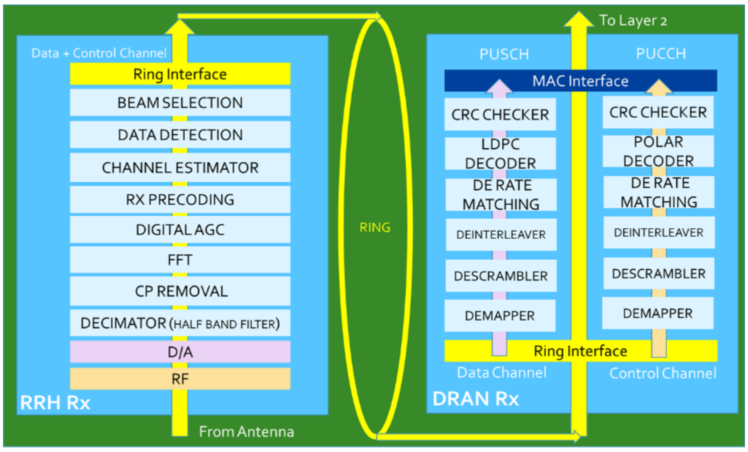

Figure 6 PHY Layer Downlink

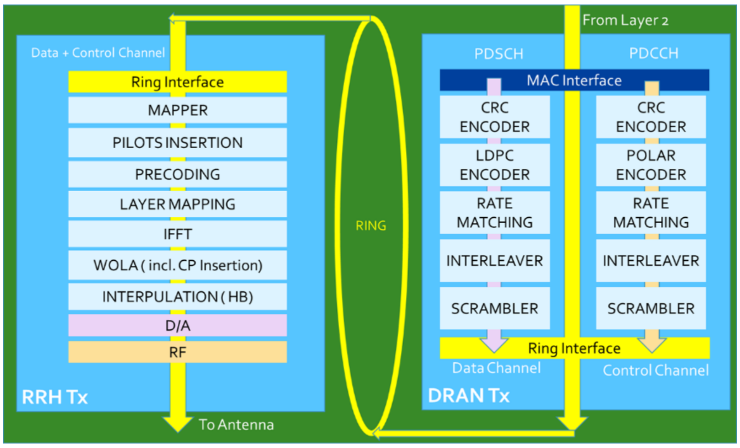

Figure 7 PHY Layer Uplink

Figure 6 and Figure 7 depict the RAN Physical layer processing which is implemented and running at the DRAN and the RRH units. Different processing is accomplished for the data and the control channels. Encoding (LDPC and Polar), rate matching, interleaving, and scrambling are accomplished at the DRAN. Mapping, RS (DMRS) insertion, precoding, layers mapping, iFFT, WOLA, interpolation D/A and RF are implemented at the RRH.

The DRAN connects with several and more RRH units. Connection is accomplished over a Nx10Gbps F/O Ethernet ring (at present N=2).

To enable early 5G tests before off the shelf UE units are available, a special UE emulator was developed by RunEL. This UE emulator connects with the base (RRH and DRAN) over the air (OTA) or via a coaxial cable.

Network servers (a video server was presented) can connect via a Lite RAN Protocols Stack Processor to the DRAN unit. Connection is via an IP over an Ethernet link. The protocol stack includes a lite version of the 3GPP RAN SDAP, PDCP, RLC and MAC protocol layers.

To run the early tests an option for either transparent or with CRC (to enable online measurements) transport was enabled. Transparent mode can handle a transport rate of over

50Mbps. When CRC is included within the data payload as of the protocol stack Windows early platform the maximal data rate is limited to 10Mbps.

3.3.2.2. OAI gNB/UE setup without connection to the Core Network

As an intermediary step until the 5g ENDC NSA solution becomes available for integration in the 5Genesis platforms, ECM has provided a special mode (noS1) which allows to perform basic Downlink and Uplink IP traffic tests (e.g., ping, iperf) over the 5G RAN stack (PDCP, RLC, MAC, PHY) of the gNB and nrUE. In this mode, there is no LTE eNB and Core Network and the 5G signaling that would normally take place between these two entities and the UE to establish a 5G connection in a NSA implementation is bypassed, by preconfiguring the required parameters at the gNB and OAI UE. As shown in Figure 8, traffic can be injected/ extracted directly at the level of the PDCP using virtual TUN interfaces at gNB and UE sides.

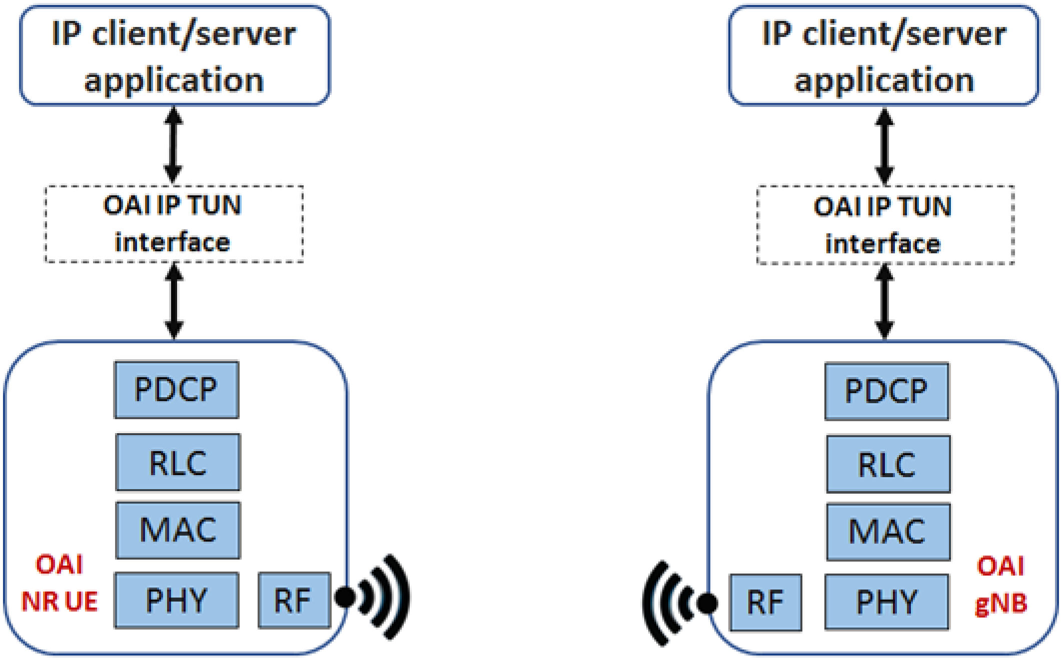

Figure 8 OAI noS1 mode architecture supporting IP traffic flow

The RAN protocol stack implementation for both 5G-NR UE and gNB have become gradually available throughout phases 1 and 2. Starting from the NR physical layer (phase 1), the noS1 mode was initially reusing the LTE functionality (3GPP Rel.14) of the upper layers to allow for IP traffic tests. As the corresponding NR functionality (3GPP Rel.15) for the upper layers and the interfaces with NR PHY progressively became available, they were integrated for the noS1 mode as well (phase 2).

3.3.2.3. ENDC NSA setup with OAI gNB and commercial UE

During phase 2, ECM has been working on the integration of the ENDC NSA mode in OAI, targeting a setup with commercial UEs which will be available for the 5Genesis platforms in phase 3.

The OAI NSA architecture to support this setup follows configuration option 3a as defined in 3GPP Rel. 15 and is depicted in Figure 9. In this context, the OAI eNB and gNB are connected over the **X2-C** interface to exchange all the required 5G configuration concerning the target UE. Thus, the eNB conveys the UE NR capabilities to the gNB and the corresponding 5G cell configuration originating from the gNB is transferred back to the UE through LTE RRC siganling. Once this configuration is exchanged and applied successfully on both sides, **S1-C** signaling between the eNB and the MME of the Core Network triggers the patch switch procedures at the SGW, so that the IP traffic can be transferred now via the gNB (**S1-U** interface) upon the completion of the random access procedures and the connection of the UE to the 5G cell (**user plane**).

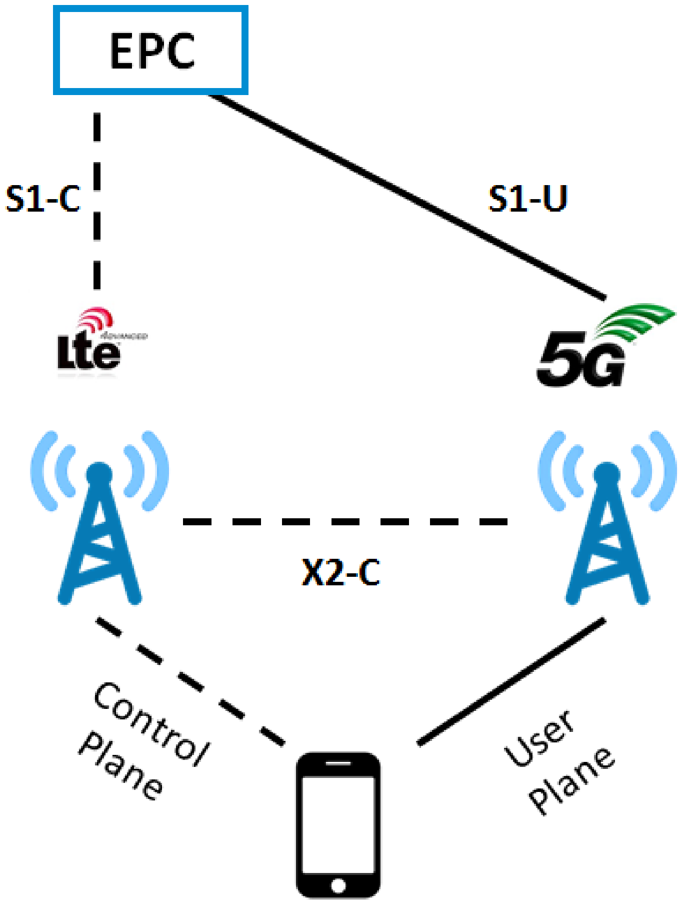

Figure 9 OAI ENDC NSA architecture

3.3.3. Edge

Traditionally, Telecommunications Central Offices or Telephone Exchanges require significant efforts for configuration and control. Meanwhile, users continuously demand higher capacities, lower latencies and more customized services. Central Office Re-architected as a Datacenter (CORD) [14] initiative, plans on breaking this paradigm, bringing the benefits of cloud computing to the Central Offices. CORD re-architects the Central Office as a data center. The basic approach centers on unifying the following three related but distinct technology trends:

- The first is SDN, which is about separating the network's control and data planes. This makes the control plane programmable, and that can lead to increased innovation. It also allows for simplification of forwarding devices that can be built using merchant silicon, resulting in less expensive white-box switches.
- The second is NFV, which is about moving the data plane from hardware devices to virtual machines. This reduces CAPEX costs (through server consolidation and replacing high margin devices with commodity hardware) and OPEX costs (through software based orchestration). It also has the potential to improve operator agility and increase the opportunity for innovation.
- The third is the Cloud, which defines the state of the art in building scalable services—leveraging software based solutions, microservice architecture, virtualized commodity platforms, elastic scaling, and service composition, to enable network operators to rapidly innovate.

While it is easy to see that all three threads (SDN, NFV, Cloud) play a role in reducing costs, it is just as important to recognize that all three are also sources of innovative (and revenue generating) services that Telcos can offer subscribers. These include control plane services (e.g., content centric networking, virtual networks on demand, cloud network binding), data plane services (e.g., Parental Control, NAT, WAN Acceleration), and global cloud services (e.g., CDN, Storage, Analytics, Internet of Things).

The goal of CORD is not only to replace today's purpose-built hardware devices with their more agile software based counterparts, but also to make the Central Office an integral part of every Telco's larger cloud strategy, enabling them to offer more valuable services. This means CORD's software architecture must be general enough to support a wide range of services. This includes both access services (e.g., Fiber to the Home) and scalable cloud services (SaaS); services implemented in the data plane (NFV) and services implemented in the control plane (SDN); trusted operator provided services and untrusted third party services; and bundled legacy services and disaggregated greenfield services.

M-CORD (Mobile CORD) is focused on addressing the needs of the mobile networks. It has been influenced by the emerging 5G use cases and is programmatically applicable to a range of performance targets on the same platform. M-CORD transforms the mobile network so that SDN control and data planes are decoupled, SDN control plane is logically centralized, cellular network functions as well as operator specific services are disaggregated and virtualized, virtualized functions and services are composed as scalable services and the overall cellular network is orchestrated so that use case-specific set of services are on-boarded and dynamically scaled.

M-CORD has a natural fit for Mobile Edge Computing (MEC). With emerging vertical sectors such as mission critical IoT, virtual reality, and advanced gaming which are all delay sensitive and can require high bandwidth, enabling use case specific services at the mobile edge can become a necessity. M-CORD is an ideal platform for providing dynamically programmable, orchestral and scalable mobile edge services.

Traditionally, Telecommunications Central Offices (CO) or Telephone Exchanges requires significant efforts for configuration and control. Meanwhile, users continuously demand higher capacities, lower latencies and more customized services. Inspired by the Central Office Re-architected as a Datacenter (CORD) initiative, plans on breaking this paradigm, bringing the benefits of cloud computing to the Central Offices. Accordingly, it introduces a new architecture for Central Offices based on edge computing that allows the virtualization of its access network and offers third-party application developers and content providers cloud-computing capabilities at the edge of the network. An innovative design based on NFV, SDN and cloud computing paradigms is proposed. Built on some CORD principles, taking its disruptive approach a step further by simplifying the implementation and Telefónica is introducing new elements, including a native IPv6-only fabric and an innovative rack design for its hardware infrastructure.

The architecture bases on CORD. Figure 6 illustrates the architecture. As shown, upon OpenNebula, ONOS [15] is the selected SDN platform for the deployment, as most of CORD functionality relies on applications developed in this platform. All network applications are similar to those in CORD but have been redesigned. Service management is coordinated by OneFlow [16]which rely on OpenNebula and Open Network Operating System (ONOS) respectively for infrastructure and network management as shown in Figure 6.

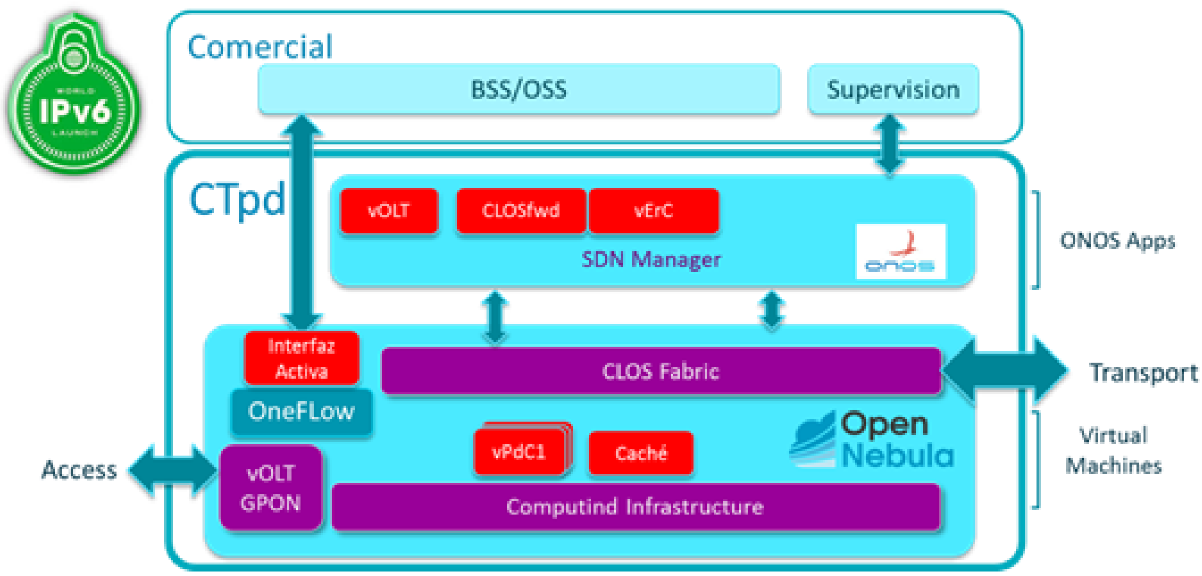

Figure 10 Edge logical architecture and components

Figure 10 shows an overview of the system where the server function as the hosts of the VNFs.

One of the main challenges is to open the CO to third-party edge computing applications, similarly to the Infrastructure as a Service (IaaS) model, which opens the data center to external workloads. The ability to provide this edge computing platform in a pay-as-you-go model, a la IaaS, opens up avenues in both innovative use cases and business models. However, given the specific characteristics of the CO in terms of computational and storage resources, and the security constraints of the environment, it is required a well-defined framework to develop such edge applications. Edge computing application should have the following characteristics:

- Stateless: In order to fast reallocate an application, or to migrate it when the user moves across the access network (e.g. from home to the office), the edge application cannot store state within the CO. Any state persistence cannot be stored at the edge, so it is proposed to be stored within the BSS through well-defined interfaces in dedicated storage services.
- Autoconfiguration: The application should be able to autoconfigure itself. This process is performed using specific information (context) passed to the edge application upon boot. The context may include user data, configuration parameters or additional resources to install the application. In this phase, the edge application will retrieve any state data needed from the BSS storage services.
- Composition: There are some complex applications that require the deployment of multiple VMs. An edge application captures this nature and includes also deployment dependencies between the VMs. The inter-connection of the VMs of each edge application happens in a separate private network.
- Elasticity: Considering application specific performance metrics, the number of VMs or application components can increase (or decrease). An elasticity rule may require for example to add more VMs at specific times and dates (e.g. with an advertisement campaign) or when the number of requests are above a given threshold.

Apart from the above characteristics, a well-defined API to manage the edge application is provided. This API resembles the classical IaaS API to control the life-cycle of a VM. The functionality exposed by OpenNebula and ONOS is used to deploy the edge application and provide it with the features mentioned above. For this OSM will interact with that API in order to orchestrate the edge infrastructure through this APIs. This is shown at Figure 8.

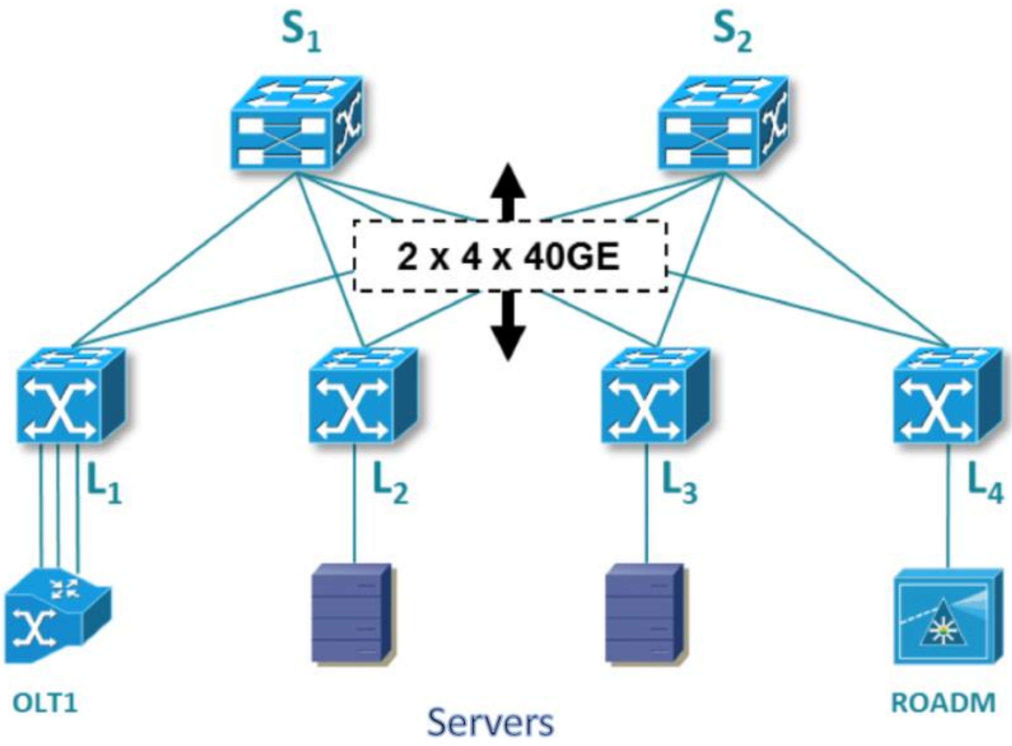

Figure 11 Edge Infrastructure

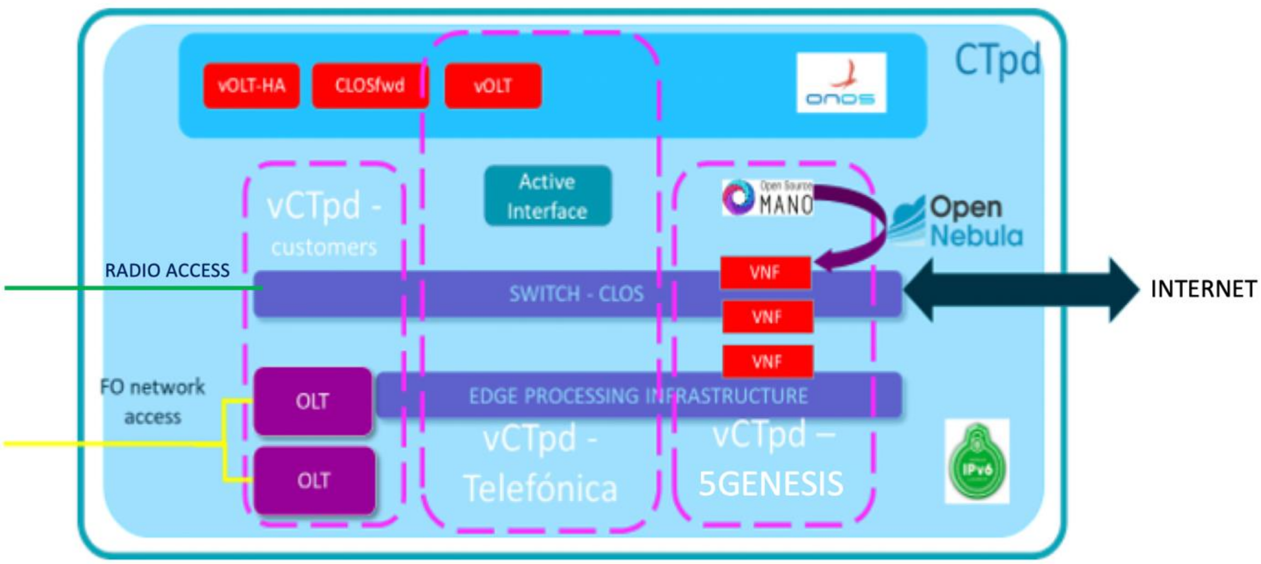

Figure 12 Diagram of VNF deployment at the edge infrastructure through OSM

3.3.4. Transport

3.3.4.1. Terrestrial backhaul

5GENESIS makes use of a terrestrial mmWave backhaul solution at 60 GHz that is suitable for both indoor and outdoor point-to-multipoint wireless communication, targeting transmission distances of up to 200 meters. The solution is based on Software Defined Radio (SDR) equipment and features beam search capabilities to ease and speed up the alignment process between the mmWave units, reaping the benefits of the use of pencil beams. Moreover, this entails extensive frequency reuse while simplifying the interference management.

The solution is envisioned for small cell deployment in dense urban areas, where low cost and flexible high data rate connections allow extending the 5G connectivity towards remote edge deployments.

3.3.4.2. Satellite backhaul

5GENESIS will also employ satellite communications for the backhaul links, thus giving the possibility to extend 5G coverage to areas and use cases far beyond the typical coverage of terrestrial radio network architectures (e.g. underserved areas, long-haul transportation media etc.) The focus will be on multi-spot geostationary (GEO) satellites, operating at Ka-band for increased throughput. 5GENESIS will exploit and properly adapt satellite networks with technical enablers for more seamless integration with 5G, such as management APIs for network slicing and QoS control and SDN/NFV capabilities at both the satellite gateway and the terminal (edge) to allow end-to-end service management through the satellite backhaul.

Within the project we will look to build a real-world implementation of the QI/QoS adaptation presented in figure 3-7 and table 3-5 in the H2020 SaT5G deliverable [1].

The terminals and antennas to be employed will support both stationary/portable use as well as mobile use, allowing the ad-hoc deployment of 5G “hot-spots” anywhere and anytime.

3.3.5. Core network

The core network functionalities developed in 5Genesis can be deployed virtualized as VNFs. The mobile core network solutions are being upgraded from 4G to offer 5G core network functionalities following the 3GPP specifications. 5Geneis deploys 4/5G network cores of different vendors in the platforms: Fraunhofer FOKUS Open5GCore [24] is deployed at the Berlin platform; the Athonet core is deployed at Málaga, Athens and Limassol. The architecture takes already into account the needs of 5G. For example, the Athonet core features 5G NSA support whereas the Open5GCore is already based on a Service Based Architecture (SBA) and focuses on supporting 5G SA.

In the 5G NSA mode, both the control plane and data plane interfaces between the RAN and the core network leverage the EPC architecture (3GPP Rel.15): the control plane interfaces are based on S1-AP and NAS signaling; in the data plane, GTP-U is still the encapsulation protocol chosen to transport the user protocol data units.

The most dramatic change introduced by the 5G core network architecture is the so-called Service Based Architecture, by which the binary protocols traditionally used in the telco industry are replaced by text-based protocols based on RESTful APIs.

3.3.5.1. Athonet 4/5G Core

Athonet 5G core network brings, starting already in 4G technology along with the NSA architecture and then in 5G SA technology, the separation of the user and control planes to ease flexible and agile deployments as required by specific 5G use cases (e.g., URLLC). Furthermore, virtualisation and distribution to the edge of core network functionalities, such as the user plane functions, allows running applications as close as possible to the users, improving service delivery and quality of experience. As defined in 3GPP TS 23.501, control plane and user planes are architecturally separated from the (R)AN between the interfaces N2 towards the Access and Mobility Management Function (AMF) and N3 towards the User Plane Function (UPF). This eases the deployment of the 5G network starting from the edge. In addition, the already rich set of APIs exposed is planned to be enhanced and harmonized in order to reflect Common API Framework (CAPIF) envisioned by Rel. 16 of the 5G specifications.

Figure 13 illustrates a combined 4G and 5G deployment as envisioned with Athonet core network technology. The diagram depicts the logical architecture, but it is worth noting that in terms of implementation, similar functions, like the AMF and MME, can be actually part of the same software instance.

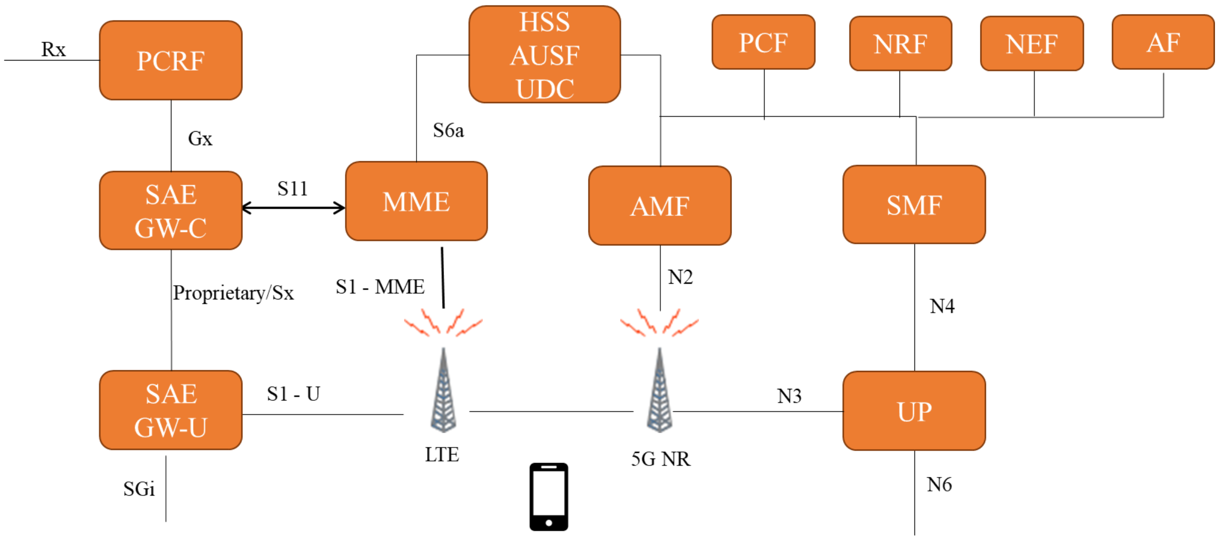

Figure 13 Combined 4G and 5G technology

3.3.5.2. Fraunhofer FOKUS Open5GCore

The Open5GCore aims at providing support and speeding-up research, by facilitating know-how transfer from Fraunhofer FOKUS towards partners. It implements the new 5G components as standalone, independent of the previous 4G EPC functionality. Through this, Open5GCore enables a fast and targeted 5G innovation, hands-on fast implementation and realistic evaluation and demonstration of new concepts and use case opportunities. The most recent release (Rel. 5) includes a large level of newly implemented functions developed on top of an accelerated software platform:

- Integration with 5G NR SA (N1, N2, N3)
- Implementing control-user plane split – PFCP (N4)
- Service-Based Architecture (HTTP/2, OpenAPI, REST)
- Local offloading and backhaul control
- Highly customizable for vertical use cases and dedicated networks
- Benchmarking tool for the 5G Core network
- Basic end-to-end support for non-3GPP access
- Support for multi-slice environments

An architecture overview of the Open5GCore is show in the following Figure 14.

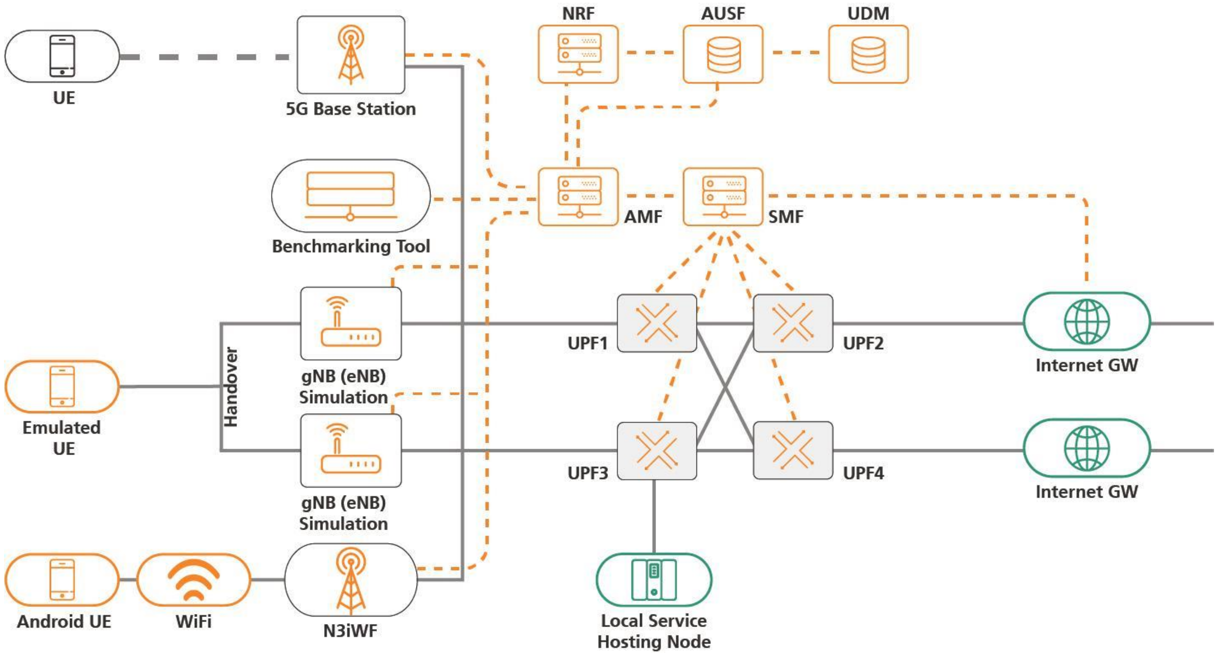

Figure 14 Open5GCore Architecture © Fraunhofer FOKUS

Open5GCore Rel. 5 integrates with 5G New Radio Stand-Alone (SA) and has successfully passed integration tests with several commercial off-the-shelf RANs. The core runs on top of common hardware platforms and can be deployed with containers or virtual machines on top of a large number of virtualization environments.

4. 5GENESIS EXPERIMENTATION METHODOLOGY

4.1. Experimentation methodology

5GENESIS has defined an experimentation methodology based on a modular approach composed of three major logical components, namely the test cases, the scenarios, and the slices. This approach allows for supporting heterogeneous experiment requirements coming from a variety of vertical industries. All the information required for running an experiment is formalized in three parts or fields, one to one mapped to the three logical components mentioned above. Thus, an experiment descriptor is defined including input information for the test case to run, the scenario under which this test case will be executed and the slices that need to be set. The use of test cases isolates from the experiment the definition of the target KPI and the procedure to compute it. The scenarios incorporate all the information related to the conditions in which the experiment will be executed trying to reproduce the real conditions in which the system under test is operated. For example, a scenario can order to tune the system to provide extremely low signal strength conditions, as an effort to extract performance measurement on the robustness of the system when channel conditions deteriorate. Finally, the slice covers the configuration and deployment of the resources assigned by the network to the experiment/system under test. Based on this approach an experiment descriptor template was introduced in D2.3 [3] and has been updated in this deliverable to make it more general, flexible and sustainable. The final 5GENESIS Experiment Descriptor template is described in Section 4.4.

4.1.1. Types of experiments

Initially, 5GENESIS methodology defined two type of experiments, the standards and the customs. The standard experiments are the experiments based on the test cases specified by the 5GENESIS consortium. These experiments enable the comparison and the benchmarking of different variants (i.e. A/B testing). To adapt the experimentation to particular requirements, the methodology also supports the definition of custom experiments in which the measurement points, among other parameters, can be specified/configured for the solution under test. In addition, and to enable the sustainability of the methodology we have opened the methodology to support new types of experiments. This is the case of MONROE experiments, defined in the MONROE project and also adopted in the 5GENESIS project. MONROE experiments are containerized experiments that can be executed on MONROE nodes [17] [18], so the testbeds equipped with these probes can use the 5GENESIS experimentation framework for running MONROE experiments .

In D2.3 the concepts of “attended” and ”unattended” experiments were introduced as well, to establish a differentiation between fully automated experiments and experiments that need human intervention during its execution. The second ones were included in the initial methodology to support the executions of experiments with real user and manual operation of the system under test. The fully automated experiments are oriented towards the exhaustive testing for detecting underperformance issues. In the final description of the 5GENESIS experimentation methodology both concepts are applicable to the different types of experiments and have been renamed to "Automated" and "Non-automated" experiments to better capture the meaning that the terms are meant to represent.

4.2. Test cases

The test case specifies the conditions of the System Under Test (SUT), the procedure to execute the tests, collect the measurements and compute the KPIs.

The test case template introduced in deliverable D2.3 has been updated by renaming some of the fields, adjusting their content and adding new fields. In particular, the field "**Test procedure**" has been renamed to "**Methodology**" and the sequence of actions to be ran during the execution of the test case has been moved to a new field named "**Test case sequence**". The "**Methodology**" includes the declaration of the required number of iterations, the monitoring time, the monitoring frequency, etc. The field "KPI computation procedure" and the field "Test case output" has been merged in a new field called "Calculation process and output". Finally, three new fields have been added:

- **Complementary measurements**. The measurements specified in this field are not the main target of the test case but can be useful when interpreting of the outputs of the test case.
- **Pre-conditions**. To ensure that the test cases are executed in the same conditions, this field specifies the conditions that need to be met by the SUT before the execution of the test case.
- **Applicability**. To verify whether the test case is applicable to the SUT, this field includes the list of features and capabilities that should be supported by the SUT when executing the test case.

Table 6 provides the 5GENESIS test case template.

Table 6 Test case template

Test Case Template		*-ID number-*	*-Related Metric ID-*
#	Description of the fields to be completed		
1	Description of the target KPI *Here goes the definition of the target KPI. Each test case targets only one KPI (main KPI). However, secondary measurements from complementary KPIs can be added as well (see field 4 in this template). The definition of the main KPI specializes the related target metric (the ID of the related target metric is declared in the first row of this template). More precisely, the definition of the main KPI declares at least the reference points from which the measurement(s) will be performed, the underlay system, the reference protocol stack level etc...*		
2	Methodology *Here the acceptable values for the monitoring time, the iterations required, the monitoring frequency, etc., are declared. The reference to the calibration test is taken from the test case. This is to facilitate the comparison between measurements.*		

3	*Calculation process and output* *Here goes information related to the calculation process required. This is information may include details related to the underlay system. Here goes also the Units of the metric, and potentially a request for first order statistics (Min, Max, etc.)*
4	Complementary measurements *A secondary list of KPIs useful to interpret the values of the target KPI. Getting these measurements is not mandatory for the test case.*
5	Pre-conditions Any requirement that needs to be done before *execution* of this test case. A list of test specific pre-conditions that need to be met by the SUT including information about **equipment configuration**, **traffic descriptor** i.e., precise description of the initial state of the SUT required to start executing the test sequence
6	Applicability A list of features and capabilities which are required to be supported by the SUT in order to execute this test (e.g., if this list contains an optional feature to be supported, then the test is optional)
7	Test Case Sequence *Specializes the measurement process (methodology) of the metric for the selected underlay system. Measurements points and measurement procedure specification.*

4.3. Scenarios and slices

The "Scenario" concept was introduced in deliverable D2.3 [3]. The parameters that are part of the definition of the scenario are different from those specified by the slice. The parameters defined in the scenario establish the working point of the network and the location and mobility conditions of the UE.

The scenario is meant to be a guideline for the definition of network conditions to reproduce realistic situations in which to perform the experiments.

The definition of the scenarios is very dependent on the infrastructure layer. Table 7 provides the final template for the description of the scenarios which includes two new fields for the definition of background traffic and available computational resources.

The slice specifies the end-to-end resources specifically allocated in the network for the system under test in order to fulfil the performance requirements of the application under test. A full description of the slicing mechanisms supported in 5GENESIS project can be found in D3.3 [10] and in its GitHub repository [19].

Table 7 Scenario template

Scenario Description Template		*-ID number-*
#	Description of the fields to be completed	
1	Radio access technology *4G,5G*	
2	Standalone / Non-Standalone (if applicable)	
3	Cell Power	
4	Frequency band: *Sub-6 GHz* *mmWave*	
5	Maximum bandwidth per component carrier *50 MHz, 100 MHz, 200 MHz, 400 MHz*	
6	Sub-carrier spacing *Sub 6 GHz: 15 kHz, 30 kHz, 60 kHz* *mmWave: 60 kHz, 120 kHz, 240 kHz, 480 kHz*	
7	Number of component carriers *Maximum number of CC = 16 (5G)* *Maximum number of CC = 5 (4G)*	
8	CP *Cyclic Prefix: normal, extended*	
9	Massive MIMO *Number of antennas on NodeB*	
10	MIMO schemes (codeword and number of layers) *The number of codewords per PDSCH assignment per UE* o *1 codeword for 1 to 4-layer transmission* o *2 codewords for 5 to 8-layer transmission.* *DL DMRS based spatial multiplexing (SU-MIMO/MU-MIMO) is supported* o *At least, the 8 orthogonal DL DMRS ports are supported for SU-MIMO* o *Maximum 12 orthogonal DL DMRS ports are supported for MU-MIMO*	
11	Modulation schemes *Downlink: QPSK, 16 QAM, 64 QAM, 256 QAM* *UplinK: QPSK, 16 QAM, 64 QAM, 256 QAM*	
12	Duplex mode *FDD, TDD*	
13	TDD uplink/downlink pattern (if applicable) 0.5 ms, 0.625 ms, 1 ms, 1.25 ms, 2 ms, 2.5ms, 5 ms, 10 ms	
14	Contention based random access procedure/contention free (if applicable)	
15	User location and speed	
16	Background traffic	
17	Computational resources available	

4.4. Experiment Descriptor

The final version of the experiment descriptor is described in this section. The template has been simplified and includes support for the execution of distributed experiments and MONROE experiments. Moreover, additional field has been included to support new features and experiments in order to ensure the sustainability of the 5GENESIS experimentation methodology.

The Experiment Descriptor is a data structure that includes all the values that are required for defining an experiment execution. The Experiment Descriptor is designed to be easy to edit, store and transfer, due to the use of the JSON format.

Since experimenters must be able to manually create new the Experiment Descriptors, the amount of data included in them has been condensed as much as possible. For example, when compared to the Experiment Descriptors used in Release A, we can see that fields devoted to user information have been removed and authentication is now handled by the upper components of the architecture (the Portal and the Open APIs).

The Experiment Descriptor template can be seen below:

```
{
  ExperimentType: Standard/Custom/MONROE
  Automated: <bool>
  TestCases: <List[str]>
  UEs: <List[str]>  UEs IDs

  Slice: <str>
  NSs: <List[Tuple[str, str]]> (NSD Id, Location)
  Scenario: <string>

  ExclusiveExecution: <bool>
  ReservationTime: <int> (Minutes)

  Application: <str>
  Parameters: <Dict[str,obj]>

  Remote: <str> Remote platform Id
  RemoteDescriptor: <Experiment Descriptor>

  Version: <str>
  Extra: <Dict[str,obj]>
}
```

The first two sets of values are the most important for the definition of the experiment. The first group includes the type of experiment, the test cases to execute and the UEs to use, while the second define the slice, network services and scenario to configure and deploy.

The third group is used to control the scheduling of the experiment. An '*Exclusive*' experiment will not be run at the same time as other experiments in the testbed, while the '*ReservationTime*' is used to define the duration of the experiment when automation is not enabled.

The fourth group is used to define the configuration of experiments that use MONROE nodes: '*Application*' defines the container to deploy in the node, and '*Parameters*' includes the configuration of the container. The '*Parameters'* field is also used for specifying customized parameters in the case of a 'Custom' experiment.

The fifth group is expected to provide the fields necessary to support the execution of distributed experiments, thought is subject to change depending on the issues and refinements detected during the development of this feature. The 'Remote' field is used to identify the secondary platform that will be part of the distributed experiment, while '*RemoteDescriptor'* contains a JSON object in the same format as the main descriptor, but excluding the 'Remote' and '*RemoteDescriptor'* fields. This secondary descriptor contains the values required to configure the experiment execution in the remote platform.

In order to ease the addition of new functionality in the future two fields have been included: The '*Version'* field can be used to specify the exact version of the Experiment Descriptor, so that the lower layers can customize the handling of the descriptor according to any future modification while keeping compatibility with older descriptors. The '*Extra'* field can be used to add any kind of information. This can be useful, for example, for adding debug or tracing information, or as an easy way to support extra functionality without changing the format of the Experiment Descriptors.

Two examples of real experiment descriptors can be seen below. The first corresponding to a Standard experiment, and the second configured for using a MONROE node. Please note that these examples do not contain fields related to the execution of distributed experiments, since development of this functionality has not started at the time of writing.

```
{
	'Version': '2.0.0',
	'ExperimentType': 'Standard',
	'TestCases': ['Throughput'],
	'UEs': ['Note10'],
	'Slice': None,
	'NSs': [['123e4567-e89b-12d3-a456-426655440000', 'Edge']],
	'ExclusiveExecution': False,
	'Scenario': None,
	'Automated': True,
	'ReservationTime': None,
	'Application': None,
	'Parameters': {},
	'Extra': {}
}

{
	'Version': '2.0.0',
	'ExperimentType': 'MONROE',
	'TestCases': [],
	'UEs': [],
	'Slice': None,
	'NSs': [],
	'ExclusiveExecution': False,
	'Scenario': None,
	'Automated': True,
	'ReservationTime': None,
	'Application': 'monroe/ping:virt',
	'Parameters': {'server': '8.8.8.8'},
	'Extra': {}
}
```

5. Distributed Experiments

5.1. Types of distributed experiments

5.1.1. Common testbed operator

One way of orchestrating the workflow can achieved through common testbed operator use case. This is illustrated in Figure 15. In this case, the two platform sites or platform shares one testbed operator. The platforms are connected through VPN. The components of layers, such as coordination layer, management and orchestration layer as well as infrastructure layer, are distributed among these platforms. That means; the East-West as well as the North-South bound interfaces are distributed among platforms. Here, east-west and north-south interfaces need to know the locations of the deployed components to communicate. This enables the platforms to act as one to perform the experiments or handling requests from UE.

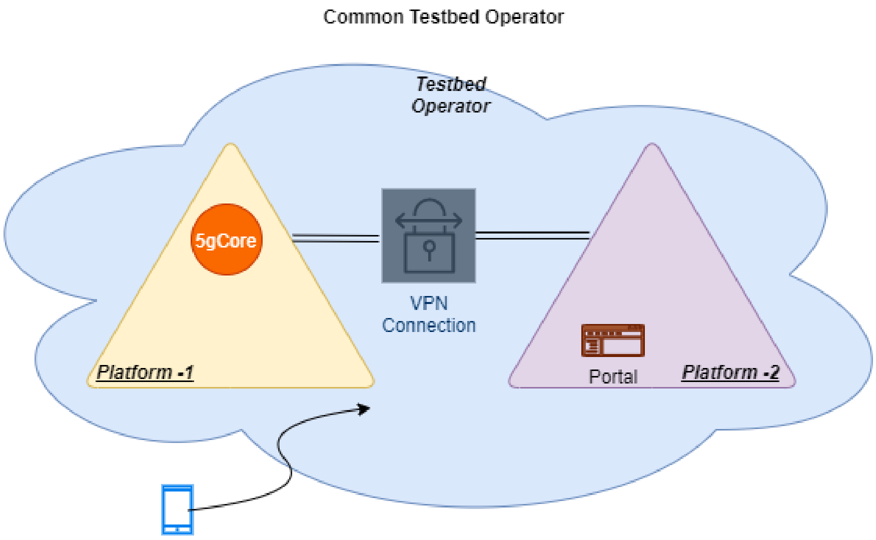

Figure 15 : Common Testbed Operator

5.1.2. Individual Testbed Operators

Figure 16 illustrates the use case where the platforms are connected to different testbed operators and the communication between them. Each platform would deploy and manage their own components layers. To execute the inter platform request or experiments these platforms need to connect through VPN. Additionally, only the east-west interfaces need to be configured to be exposed among the platforms. This also enforces the platforms to focus on their own deployed components. In order to display the communication; an example is shown where an UE at Operator 2 wants to communicate to the server hosted at operator 1 or internet. The server can be considered as a web application that the user wants to access through the user device. In the case of application hosted in the operator 1 test bed, then Operator1 will provide or expose channel from the application/server to the internet or VPN connection . Similarly, Operator 2 will provide or expose channel from the UE to the VPN/internet connection. This is to emphasize that the platforms need to focus on their own deployment.

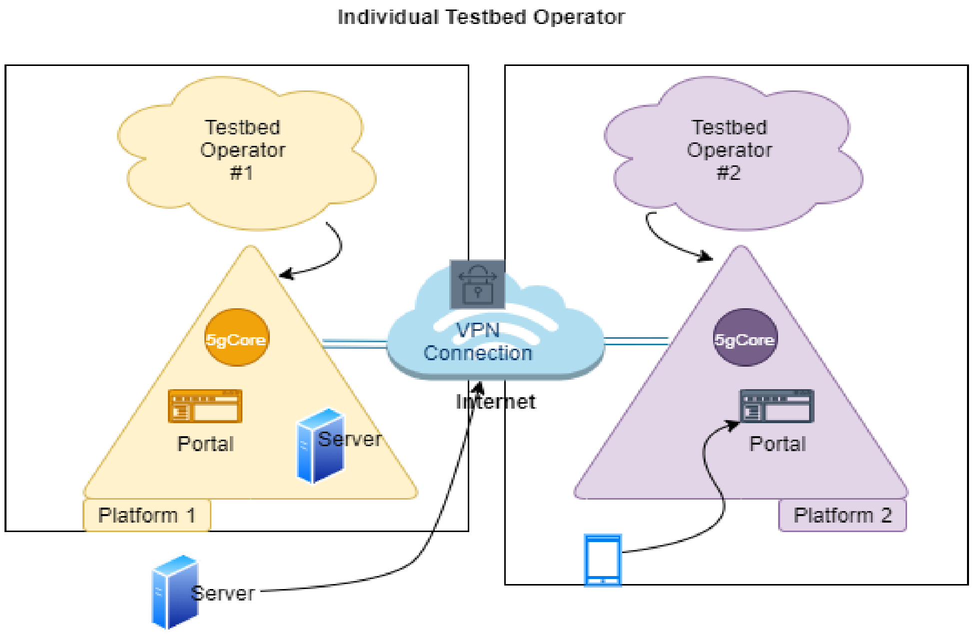

Figure 16 : Individual Testbed Operators

5.2. General workflow of distributed experiments

The execution of distributed experiments that involve multiple 5GENESIS platforms is based on the synchronization and exchange of information between the platforms as they execute their respective actions. This means that each platform is responsible for the execution of part of the experiment, while coordinating with the other platforms to guarantee the correct execution of actions that need to be performed in a certain order, or to obtain required information from other platforms when required. Figure 17 shows an example of the action flow between two platforms during an experiment execution. It is important to note that some details in this example may change depending on issues or improvements identified during the implementation of the support for distributed experiments, but, in general, the final workflow should be very close to the one presented in this document.

In this example, which is based on a Mission Critical Service call between two UEs in different platforms, a complete experiment descriptor is received by the dispatcher of the Host platform. Since this descriptor defines the execution of a distributed experiment, the experiment descriptor is divided in parts that are sent to the two platforms involved in the experiment.

Once both platforms received the request for execution, the experiment starts. Platform 1 is in charge of deploying the Mission Critical Service and registering the first UE. Platform 2 must know the "Domain" of the service in order to register the second UE. For this reason, Platform 2 waits until Platform 1 finish the deployment stage (synchronization) and then asks for the value of the "Domain" variable (information exchange).

At this point, the UE in Platform 1 is in charge of calling to the user in Platform 2, however, in order to perform this action, Platform 2 must have finished the registration of the second UE, and a user name must be known at the MCS application (for example, "Officer James"). For this reason, Platform 1 waits until Platform 2 finish “Registering”, and then retrieves the “User” identifier that will receive the call.

This sequence of synchronization and/or information exchange is repeated as needed until all the steps in the experiment are completed. Once the execution ends, Platform 1, which received the original execution request, retrieves all the results generated by Platform 2 during the experiment in order to consolidate them in a single results repository.

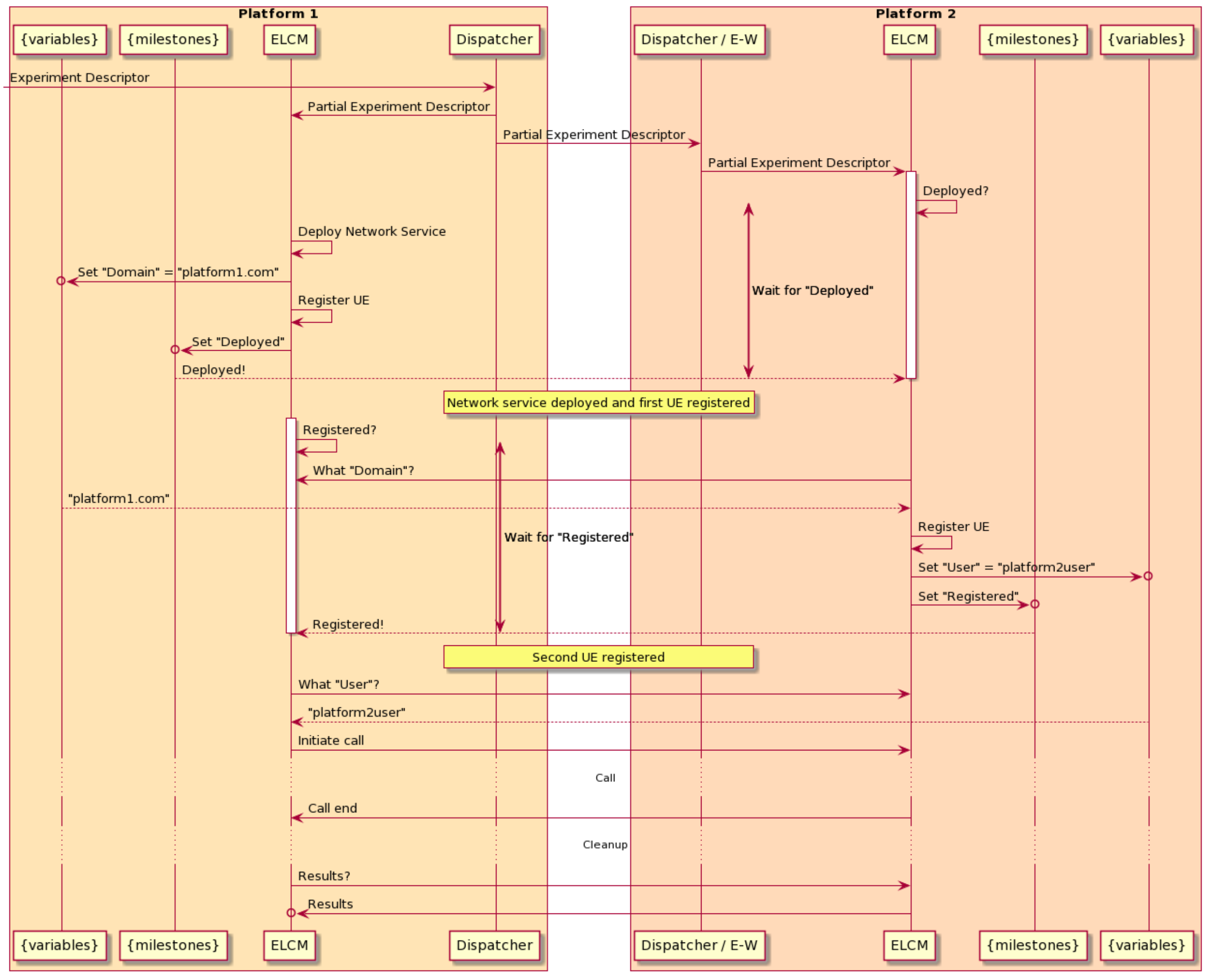

Figure 17. Example of a distributed experiment’s execution flow

Prior to the execution of the Experiment in a distributed environment, some agreements need to be fulfilled by the platform’s administrators, in order to allow remote platforms to experiment with the host platform.

The following diagram explains the interaction between the two platforms to enable the communication for the distributed experiment.

Once the remote platform has been successfully registered, the Dispatcher of the host platform will be able to communicate with the Dispatcher of the remote platform.

It is expected that most of the east-west interface implementation will reside in the ELCM, and that there will be direct communication between the ELCM of the platforms once the experiment execution starts. The execution request, however, will be most probably implemented in the Dispatcher, since authentication and authorization is handled in this component.

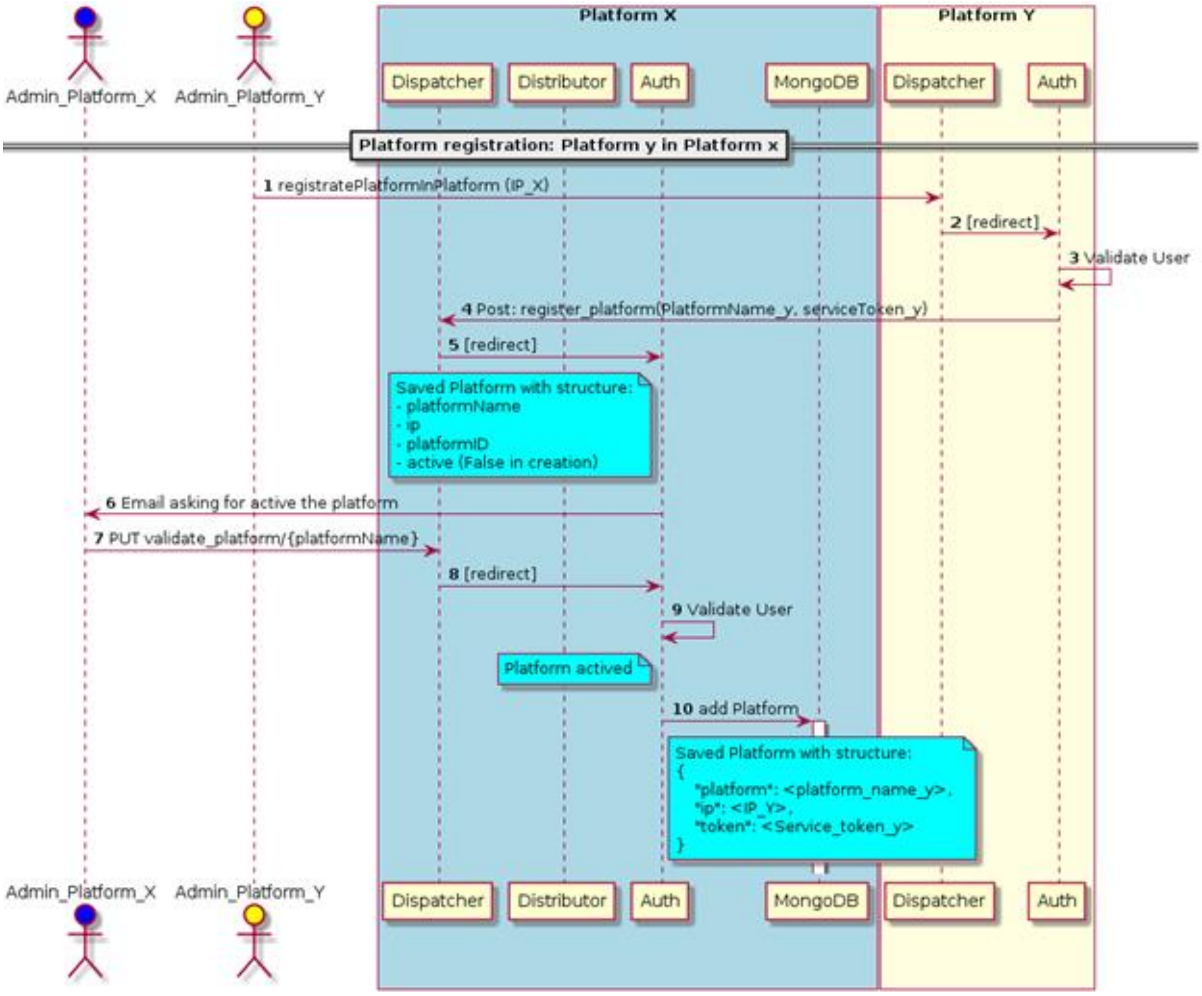

Figure 18 Remote platform registration in a distributed experiment

In order to support the execution of distributed experiments a number of new requirements have been identified and can be seen in Section 2 (EW-1 to EW-4). The following is a list of endpoints that will address these requirements.

5.2.1. Request the execution of an experiment

Endpoint	/execution
Method	POST
Payload	Experiment Descriptor (see section 4.4)
Response	{ 'execution_id': ‘xxxx’,

	'success': True/False, 'message': 'Status or error description' }

5.2.2. Retrieve the status of an experiment execution

Endpoint	/execution/{execution_id}/status
Method	GET
Payload	-

Response	{
	'success': True/False,
	'message': 'Status or error description'
	'status': [Init/PreRun/Run/PostRun/Finished/Cancelled/Errored]
	'milestones': List[str]
	}

5.2.3. Retrieve all the variables exposed by an experiment execution

Endpoint	/execution/{execution_id}/values
Method	GET
Payload	-

Response	{
	'success': True/False,
	'message': 'Status or error description'
	'values': Dict[str, str]
	}

5.2.4. Retrieve a single variable exposed by an experiment execution

Endpoint	/execution/{execution_id}/values/{value_name}
Method	GET
Payload	-
Response	{ 'success': True/False, 'message': 'Status or error description'

	'value': str }

5.2.5. Retrieve all the results of an experiment execution

Endpoint	/execution/{execution_id}/results
Method	GET
Payload	-
Response	Acceptable response formats include: - A compressed file with all results in different CSV files - A JSON object containing all the results

6. CONCLUSIONS

This deliverable describes the final 5GENESIS architecture and 5GENESIS experimentation methodology, two pillars for offering advanced experimentation features in the 5G era.

The final 5GENESIS architecture has been built based on a set of detailed requirements presented in this document. These requirements are the results of a large series of internal meetings.

The explosion of uses cases covered by 5G networks makes necessary to adopt a systematic approach for testing the solutions coming from all the vertical industries. Due to this, 5GENESIS project has centered its efforts in providing an open and flexible 5GENESIS experimentation framework which implements the 5GENESIS experimentation methodology and delivers an open and common approach based on test cases for the benchmarking and trialing of 5G solutions and services.

Finally, a distributed experimentation methodology is also presented in this deliverable. Due to the variety of 5G use cases, testbeds specialize in some of them integrating solutions not available in order platforms. This distributed methodology enables the execution of experiments between 5GENESIS platforms.

REFERENCES

[1] 3GPP TS 23.501 System architecture for 5G Systems, Stage 2, Release 16, http://www.3gpp.org/ftp/Specs/archive/23_series/23.501/
[2] H2020 5GENESIS Deliverable D2.2 "5GENESIS Overall Facility Design and Specifications", https://5genesis.eu/wp-content/uploads/2019/12/5GENESIS_D2.2_v1.0.pdf
[3] H2020 5GENESIS Deliverable D2.3 "Initial planning of tests and experimentation", https://5genesis.eu/wp-content/uploads/2019/12/5GENESIS_D2.3_v1.0.pdf
[4] H2020 5GENESIS Deliverable D2.1 "Requirements of the Facility", https://5genesis.eu/wp-content/uploads/2019/12/5GENESIS_D2.1_v1.0.pdf
[5] H2020 5GENESIS Deliverable D3.7 "Open APIs, service level functions and interfaces for verticals (Release A), https://5genesis.eu/wp-content/uploads/2019/10/5GENESIS_D3.7_v1.0.pdf
[6] H2020 5GENESIS D3.5 "Monitoring and Analytics (Release A)", https://5genesis.eu/wp-content/uploads/2019/10/5GENESIS_D3.5_v1.0.pdf
[7] H2020 5GENESIS D3.15 "Experiment and Lifecycle Manager (Release A)", https://5genesis.eu/wp-content/uploads/2019/10/5GENESIS_D3.15_v1.0.pdf
[8] H2020 5GENESIS D6.1 "Trials and experimentation - cycle 1" , https://5genesis.eu/wp-content/uploads/2019/12/5GENESIS_D6.1_v2.00.pdf
[9] H2020 5GENESIS D3.13 "5G Security Framework (Release A)", https://5genesis.eu/wp-content/uploads/2019/10/5GENESIS_D3.13_v1.0.pdf
[10] H2O2O 5GENESIS D3.3 "Slice Management (Release A)", https://5genesis.eu/wp-content/uploads/2019/10/5GENESIS_D3.3_v1.0.pdf
[11] H2020 SaT5G deliverable D3.2 "Detailed Network Architecture", https://sat5g-project.eu/wp-content/uploads/2019/04/SaT5G_D3.2_ADS_v2.0_D.pdf
[12] 5G PPP Platforms Cartography, "5G Pan-EU Trials Roadmap Version 4.0", https://5g-ppp.eu/wp-content/uploads/2018/11/5GInfraPPP_TrialsWG_Roadmap_Version4.0.pdf
[13] H2020 5GENESIS deliverable D4.11 "The Surrey Platform (Release B)", https://5genesis.eu/wp-content/uploads/2020/02/5GENESIS_D4.11_v1.0.pdf
[14] https://www.opennetworking.org/cord/
[15] https://www.opennetworking.org/onos/
[16] https://docs.opennebula.io/5.10/advanced_components/application_flow_and_auto-scaling/appflow_use_cli.html#oneflow-services-management
[17] Monroe experiments, https://github.com/MONROE-PROJECT/Experiments
[18] V. Mancuso et all. "Results from running an experiment as a service platform for mobile broadband networks in Europe", Computer Communications, Volume 133, 2019, Pages 89-101
[19] atana Slice Manager, https://github.com/skolome/katana-slice_manager
[20] ETSI NFV Industry Specification Group (ISG) Publications", https://docbox.etsi.org/ISG/NFV/Open/Publications_pdf/Specs-Reports
[21]3GPP TS 22.280 Mission Critical Services Common Requirements, Release 16, http://www.3gpp.org/ftp/Specs/archive/22_series/22.280/22280-g30.zip

[22] "What is Virtualized Infrastructure Manager (VIM)? Definition," [Online]. Available: https://www.sdxcentral.com/nfv/definitions/virtualized-infrastructure-manager-vim-definition/
[23] H2020 5GENESIS deliverable, "Documentation and supporting material for 5G stakeholders (Release A)", https://5genesis.eu/wp-content/uploads/2020/07/5GENESIS-D5.3_v1.0.pdf
[24] 5G open core, http://open5gcore.org