5G ExPerimentation Infrastructure hosting Cloud-nativE Netapps for public proTection and disaster RElief

Innovation Action – ICT-41-2020 - 5G PPP – 5G

Innovations for verticals with third party services

D1.4: Experimentation requirements and architecture specification final version

Delivery date: January 2023

Dissemination level: Public

Project Title:	**5G-EPICENTRE - 5G ExPerimentation Infrastructure hosting Cloud-nativE Netapps for public proTection and disaster RElief**
Duration:	**1 January 2021 – 31 December 2023**
Project URL	https://www.5gepicentre.eu/

This project has received funding from the European Union's Horizon 2020 Innovation Action programme under Grant Agreement No 101016521.

Document Information

Deliverable	D1.4: Experimentation requirements and architecture specification final version
Work Package	WP1: 5G-EPICENTRE platform requirements and experimentation planning
Task(s)	T1.2: Requirements of 5G experimentation infrastructures for PPDR use case innovations T1.3 5G-EPICENTRE technical specifications and architecture
Type	Report
Dissemination Level	Public
Due Date	M24, December 31, 2022
Submission Date	M25, January 31, 2023
Document Lead	Konstantinos Apostolakis (FORTH)
Contributors	George Margetis (FORTH) Konstantinos Tsiomos (NOVA) Jorge Marquez Ortega (UMA) Almudena Díaz Zayas (UMA) Josep Mangues-Bafalluy (CTTC) Manuel Requena (CTTC) Hamzeh Khalili (CTTC) Luigi D'Addona (IST) Anna Maria Spagnolo (IST) Apostolis Siokis (IQU) Daniel del Teso (NEM) Yerasimos Yerasimou (EBOS) Nicola di Pietro (ATH) Pedro Tomás (ONE) Carlos Martins Marques (ALB) Ankur Gupta (HHI) Kirsten Krüger (HHI) Holger Gäbler (HHI)
Internal Review	Manuel Requena (CTTC) Bruno García García (UMA)

Document history

Version	Date	Changes	Contributor(s)
V2.1	02/12/2022	Initial deliverable structure after first architecture refinement workshop	Konstantinos Apostolakis (FORTH)
V2.2	10/01/2023	Revised Introduction (Section 1), Preparation of Section 3 content for gathering feedback from partners.	Konstantinos Apostolakis (FORTH) George Margetis (FORTH)
V2.3	13/01/2023	Integration of partner inputs (Section 3) after first architecture refinement workshop: ALB, HHI	Carlos Martins Marques (ALB) Ankur Gupta (HHI)
V2.4	16/01/2023	Integration of partner inputs (Section 3) after first architecture refinement workshop: NEM	Daniel del Teso (NEM)
V2.5	17/01/2023	Integration of partner inputs (Section 3) after first architecture refinement workshop: UMA	Almudena Díaz Zayas (UMA) Jorge Marquez Ortega (UMA)
V2.6	18/01/2023	Integration of partner inputs (Section 3) after first architecture refinement workshop: UMA, IST, NEM	Almudena Díaz Zayas (UMA) Luigi D'Addona (IST) Anna Maria Spagnolo (IST) Daniel del Teso (NEM)
V2.7	19/01/2023	Integration of partner inputs (Section 3) after first architecture refinement workshop: HHI, IST. Architecture information view and related diagrams and schematics.	Ankur Gupta (HHI) Kirsten Krüger (HHI) Holger Gäbler (HHI) Luigi D'Addona (IST) Anna Maria Spagnolo (IST) Konstantinos Apostolakis (FORTH)
V2.8	20/01/2023	Integration of partner inputs (Section 3) after first architecture refinement workshop: HHI, IST, UMA, CTTC, EBOS. Architecture deployment view and related diagrams and schematics.	Ankur Gupta (HHI) Kirsten Krüger (HHI) Holger Gäbler (HHI) Luigi D'Addona (IST) Anna Maria Spagnolo (IST) Jorge Marquez Ortega (UMA) Manuel Requena (CTTC) Josep Mangues-Bafalluy (CTTC) Hamzeh Khalili (CTTC) Yerasimos Yerasimou (EBOS) Konstantinos Apostolakis (FORTH)
V2.9	20/01/2023	Revised Section 2 based on the outputs of Task 1.2.	Konstantinos Tsiomos (NOVA)

V3.0	24/01/2023	Consolidation of final inputs from IQU, CTTC, UMA. Version preparation for internal review.	Konstantinos Apostolakis (FORTH) Manuel Requena (CTTC) Apostolis Siokis (IQU)
V3.1	27/01/2023	First internal review comments and feedback	Bruno García García (UMA)
V3.2	30/01/2023	Second internal review comments and feedback	Manuel Requena (CTTC)
V3.3	31/01/2023	Integration of final modifications and clarifications requested for Section 2 (NOVA) and Section 3 (FORTH, ONE, IQU, IST, CTTC, UMA), after comments from the internal reviews.	Konstantinos Apostolakis (FORTH) Pedro Tomás (ONE) Apostolis Siokis (IQU) Luigi D'Addona (IST) Almudena Díaz Zayas (UMA) Manuel Requena (CTTC)
V4.0	31/01/2023	Final version for submission, after quality review (copyediting; proofreading; formatting).	Konstantinos Apostolakis (FORTH)

Project Partners

Logo	Partner	Country	Short name
	AIRBUS DS SLC	France	**ADS**
	NOVA TELECOMMUNICATIONS SINGLE MEMBER S.A.	Greece	**NOVA**
	Altice Labs SA	Portugal	**ALB**
	Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V.	Germany	**HHI**
	Foundation for Research and Technology Hellas	Greece	**FORTH**
	Universidad de Malaga	Spain	**UMA**
	Centre Tecnològic de Telecomunicacions de Catalunya	Spain	**CTTC**
	Istella SpA	Italy	**IST**
	One Source Consultoria Informatica LDA	Portugal	**ONE**
	Iquadrat Informatica SL	Spain	**IQU**
	Nemergent Solutions S.L.	Spain	**NEM**
	EBOS Technologies Limited	Cyprus	**EBOS**
	Athonet SRL	Italy	**ATH**
	RedZinc Services Limited	Ireland	**RZ**
	OptoPrecision GmbH	Germany	**OPTO**
	Youbiquo SRL	Italy	**YBQ**
	ORamaVR SA	Switzerland	**ORAMA**

List of abbreviations

Abbreviation	Definition
5G-PPP	5G Public Private Partnership
5QI	5G Quality of Service Identifier
6G-IA	6G Smart Networks and Services Industry Association
AICO	Analytics, Intelligence, Control and Orchestration
AF	Application Function
API	Applications Programming Interface
CIR	Container Image Registry
CIS(M)	Container Infrastructure Service (Management)
CNI	Container Network Interface
CRI	Container Runtime Interface
CSI	Container Storage Interface
curl	Client URL (Uniform Resource Locator)
DoA	Description of Action
ETSI	European Telecommunications Standards Institute
EU	European Union
GA	Grant Agreement
GDPR	General Data Protection Regulation
(G)UI	(Graphical) User Interface
HTTP	HyperText Transfer Protocol
IFD	Information Flow Diagram
ISG	Industry Specification Group

IT	Information Technologies
K8s	Kubernetes
KPI	Key Performance Indicator
MANO	Management & Orchestration
ML	Machine Learning
MoSCoW	Must Have, Should Have, Could Have, Won't Have
NEF	Network Exposure Function
NIDS	Network Intrusion and Detection System
(C/V)NF(C/M)	(Containerized or Cloud-native/Virtual) Network Function (Component/Manager)
(C)NFV(I/O)	(Cloud-native) Network Functions Virtualization (Infrastructure/Orchestrator)
PaaS	Platform as a Service
PPDR	Public Protection and Disaster Relief
QoS/E	Quality of Service/Experience
R&D	Research & Development
RA	Reference Architecture
RAN	Radio Access Network
RBAC	Role-Based Access Control
REST	Representational State Transfer
SME	Small and Medium-sized Enterprise
UC	Use Case
VI(M)	Virtualized Infrastructure (Manager)
VM	Virtual Machine
WP	Work Package

Executive summary

This deliverable reports on the final list of technical requirements; component specifications; and overall platform architecture driving the design of the 5G-EPICENTRE integrated solution. It corresponds to the second and final version of deliverable D1.3 "Experimentation requirements and architecture specification preliminary version". The deliverable does not replace the precursor document, but rather complements it with information that corresponds to the progress made in Tasks T1.2 and T1.3 since the delivery of D1.3. Wherever specified however, the contents in this document shall take precedence over those in D1.3, thus reflecting the project's evolutionary approach, and capacity of the Consortium to align to evolving needs and developments outside its control.

The intention of this document is to provide a comprehensive overview of the facility technical specifications, along with a functional viewpoint of the architectural building blocks that will comprise the final platform to be delivered. Therefore, a high-level overview of the envisioned roles, responsibilities and interrelationships among platform components will be documented, to be used as guidelines for the final development of the individual technological components in the context of Work Packages 2-4.

This document provides a thorough description of the methodology and tools used to verify and validate the platform stakeholders' requirements, which have been previously analysed and classified into a comprehensive set of high-level functional and non-functional requirements in D1.3. New platform stakeholders (*i.e.*, experimenters expected to utilise the platform for the purpose of experimenting with their solutions) have been consulted to inform the 5G-EPICENTRE platform design based on their needs and expectations regarding the elicited capabilities that the system should expose for 5G experimentation. In addition, the list of technical requirements has been verified, validated and re-prioritized, based on those stakeholders' feedback. This insight has been utilised in order to re-imagine (wherever necessary) the original architectural model, to meet stakeholder intent.

The document then presents the methodology to refine this architectural model, namely the various stages of the architecture re-design approach. These constitute a technology exploration step, informed further by the joint activities of the 5G-PPP ICT-41 projects with respect to the joint definition of the Network Applications ecosystem. This step was followed by a top-down re-design of the platform architecture, identifying both new and obsolete building blocks and functionalities, as well as tools and solutions for crucial system "must-have" features, that have evolved in parallel with project activities (*e.g.*, mature Karmada management system for cross-orchestration of Kubernetes cluster resources becoming available in July 2021 – M7). The process was finally complemented by a bottom-up re-specification of both past and present functional blocks.

Finally, four architectural viewpoints are defined, namely the *functional view*; *information view*; *deployment view*; and *security view*. The functional view specifies the roles and capabilities provided by the various functional blocks identified in the re-design process. The information view defines how information flows through the integrated system during common usage scenarios of the platform component stack. The deployment view provides insight on deployment of the hardware and software artefacts across available virtualised and bare-metal infrastructure. Finally, the security view incorporates a security perspective in terms of how the larger attack surface created by embracing a Kubernetes deployment environment can be addressed by design; as well as via dedicated Network Application solutions offered by the platform itself.

The deliverable represents the final output of Task T1.2 and T1.3 of the 5G-EPICENTRE Work Package 1.

Table of Contents

List of Figures

List of Tables

1 Introduction

This deliverable reports on the second and final version of the 5G-EPICENTRE requirements specifications and platform architecture, informed by the 24 months of rigorous research and development happening across the project's technical and design Work Packages (WPs). More specifically, this document updates on the material reported in the preliminary version of this deliverable (D1.3), and thus presents the conclusive outcomes of the project Tasks T1.2 and T1.3, which have both completed their activities as of M24.

Originally, the contents of D1.3 were intended as a reference guide for the technical implementation taking place in the context of WPs 2 to 4, so to help guide developers better understand and apply the different innovative concepts and technological augmentations for the envisioned Cloud-native 5G experimentation facility and infrastructures. On the other hand, it served as a basis for the project scientific proposition on federating the facilities located in Aveiro, Berlin, Málaga and Barcelona under the contemplated Cloud-native architecture amplification approach [1].

The structure of this document aims at reflecting the work carried out in the context of the Tasks that comprised platform conceptualisation and design (Task T1.2 and T1.3):

In line with the project's evolutionary approach, the requirements of the platform (*i.e.*, the complete 5G-EPICENTRE solution and its comprising components), as well as its architecture have since been refined over the course of the project (M7-M24), to reflect both changes and evolving needs arising from the development and verification of the different functional components and subsystems (reported in deliverables D4.4, and more recently, D4.6). Further, developments external (but parallel) to the project, having significant impact on the project overall technical approach, have been taken under consideration, in order maintain the project's alignment to the "Network Application" concept being demystified in the context of 5G-EPICENTRE and its sister ICT-41 projects [2]. Special care has been taken to endorse 5G-EPICENTRE status as the reference ICT-41 project for experimenting with Network Applications, to demonstrate the transformative capabilities of 5G technology for Public Protection and Disaster Relief (PPDR) stakeholders.

Following the 5G-EPICENTRE work plan for WP1, Task T1.2 was responsible for engaging with the intended 5G-EPICENTRE platform end users (PPDR innovators, as identified in deliverables D6.1 [revision 1] and D6.2), comprised of both project beneficiaries, as well as external entities from the 5G Public Private Partnership (5G-PPP) and the 6G Smart Networks and Services Industry Association (6G-IA), of which several industry and Small and Medium-sized Enterprises (SMEs) members represent key 5G-EPICENTRE platform stakeholders (*i.e.*, with a service portfolio tailored to the communication needs of PPDR agencies and organisations). The goal of this Task has been to identify and manage desires, wants and needs of such stakeholders in using the proposed 5G-EPICENTRE solution. The final outcomes of this activity are reported in the updated Section 2.

The elicited needs and expectations of the foreseen platform stakeholders have been further processed and translated into a comprehensive set of functional (*i.e.*, **what** the platform does) and non-functional (*i.e.*, **how** the platform does) requirements. Already since D1.3, a preliminary, yet comprehensive set of these requirements has been prioritised following the MoSCoW (Must Have, Should Have, Could Have, Won't Have) analysis method, so that platform critical and immediate features were implemented with a higher sense of urgency. This prioritization has since been amended, and the technical requirements have been validated (*i.e.*, ensuring that desired platform functionality is adequately represented by its functional element), so as to ensure that they are indeed sufficient in describing the necessary technical aspects that the platform should encompass.

Task 1.3 on the other hand was responsible to transform the list of validated technical requirements into the overall platform architecture specification. As specified in D1.3, the Task has employed a three-step process involving:

(i) **technology exploration**, where the background of the project Beneficiaries, along with any third-party technologies that were deemed to satisfy technical requirement criteria were evaluated by project partners and subsequently onboarded to the 5G-EPICENTRE architectural stack;

(ii) **top-down specification**, where, the 5G-EPICENTRE abstract platform is decomposed into modules, sub-systems and functional blocks, each with distinct roles and interdependencies; and

(iii) **bottom-up refinement**, where each block defined in the top-down approach is elaborated in terms of what each functional element will be, what capabilities it shall provide (in reference to the technical requirements), and what communication aspects should be considered in its implementation.

D1.3 has been responsible for defining the first version (M6) of the system architecture, preparing the 5G-EPICENTRE development team for the prototypical implementation to be carried out in the context of the technical WPs. A recursive approach has since been followed to further refine and evolve the project's architecture diagram and components from three distinct architectural views (functional, information and deployment). The outcomes of this activity are described in the updated Section 3 of the present document. Finally, Section 4 summarises the concluding remarks of the two-year exercise that has led to the reported outcomes.

1.1 Mapping of project's outputs

The purpose of this section is to map 5G-EPICENTRE Grant Agreement (GA) commitments within the formal Task description, against the project's respective outputs and work performed.

Table 1: Adherence to 5G-EPICENTRE's GA Tasks Descriptions

5G-EPICENTRE Task	Respective Document Chapters	Justification
T1.2: Requirements of 5G experimentation infrastructures for PPDR use case innovations *"[…] Requirement extraction will be performed in order to obtain information on useful features from a sufficient sample of potential platform end-users (in this case, the SMEs and PPDR service providers who participate in the 5G-EPICENTRE consortium), collected in the form of a survey".*	Section 2.1 – Method and procedure	In Section 2.1, the survey tools and methodology are described.
	Section 2.2 – Survey participants and demographics	Participants representing potential platform end users are analysed in Section 2.2.
	Annex I: Electronic survey (Round 2) – Questionnaire	The online questionnaire is provided for reference in Annex I.
T1.2: Requirements of 5G experimentation infrastructures for PPDR use case innovations *"[…] Following the internal end-user requirements gathering, a co-design methodology will be followed for the design of the 5G infrastructure with the involvement of all relevant stakeholders".*	Section 2.3 – Survey results – stakeholder expectations	Section 2.3 presents further elaboration of platform requirements and functional concepts derived from the internal requirements gathering (D1.3), enabling stakeholders inside and outside of the Consortium to further inform the design of the 5G-EPICENTRE platform.

T1.2: Requirements of 5G experimentation infrastructures for PPDR use case innovations *"[…] requirements relating to platform capabilities will be captured (D1.4). These will include several aspects of the envisaged platform, such as the VNFs and NetApps implementations 5G-EPICENTRE should provide in order to enable a variety of PPDR experiments to be hosted on".*	Section 2.4 – Survey results – technical requirements	In the corresponding Section, the final exercise at prioritising the platform's functional and non-functional requirements is presented at a system-level.
T1.3: 5G-EPICENTRE technical specifications and architecture *"[…] This Task will address the 5G-EPICENTRE architectural design, including: i) specification of logical and deployment architectures with clear identification of the back-end components and environments, communication protocols, etc.; and ii) clear definition of interfaces, in order to ensure smooth communication and interoperability of all 5G-EPICENTRE components, as well as the adoption of a modular design and architectural style, which will allow for flexible integration of both new and existing building blocks (e.g. from partners' previous works as the core basis of the 5G-EPICENTRE platform)".*	Section 3.5 – 5G-EPICENTRE Platform functional view	In the corresponding Sections, a functional view of the architecture is provided, identifying the functional blocks required, along with well-defined roles and capabilities.
	Section 3.6 – Testbed/Infrastructure Layer functional view	
	Section 3.7 – Information view	In the corresponding Section, a view of the architecture from an information exchange perspective is provided, identifying all information flows that facilitate exchange of information among functional blocks in the architecture for different usage scenarios of the platform.
	Section 3.8 – Deployment view	In the corresponding Section, the deployment of the architectural building blocks across the available hardware and virtualized infrastructure is discussed.
T1.3: 5G-EPICENTRE technical specifications and architecture *"[...] this Task will define a general architecture and specifications for the 5G-EPICENTRE VNFs, including properties, rules and best practices targeted at VNF developers in WP5, so as to fully support the 5G-EPICENTRE capabilities and foreseen VNF lifecycle".*	Section 3.3.1 – Network Application	In the corresponding Section, implications of the common ICT-41 approach for VNF implementation on the 5G-EPICENTRE architecture are detailed.
	Annex I:	The Annex incorporates content presented in D1.3, relating to the adoption of Cloud-native in 5G-EPICENTRE. The relationship between Network Applications and VNF/CNF components is further elaborated.

T1.3: 5G-EPICENTRE technical specifications and architecture *"[…] this Task will include an overview of legal requirements that the architecture must incorporate from a privacy and data protection by design perspective, ensuring 5G security mechanisms are natively embedded within the overall platform architecture, according to security-by-design principles".*	Section 3.9 – Security view	In the corresponding Section, provisions of the 5G-EPICENTRE platform architecture with respect to its security aspects are captured.
T1.3: 5G-EPICENTRE technical specifications and architecture *"[…] This Task will produce a "live" document, which will be kept updated to reflect the functional specifications of the 5G-EPICENTRE platform, with clear references to the architectural reference model described in Section 1.3.2".*	Section 3 – 5G-EPICENTRE platform architecture Section 3.1 – Preliminary platform architecture recapitulation (D1.3) [M6] Section 3.2 – Architecture refinement approach	In the corresponding Sections, the evolutionary approach followed during the architecture specification over the course of the project is captured.

1.2 Updates since the initial deliverable version

The present document is the second (and final) version of deliverable D1.3. The following Table (Table 2) lists all updates introduced in this latest version of the deliverable, describing what material is new, or updated compared to the previous version.

Table 2: Deliverable updates since the initial deliverable version

Document Chapter	Updates since the initial deliverable version
Section 1	The Section has been updated to reflect upon the entire lifecycle of Tasks T1.2 and T1.3. A novel Section 1.2 has been added to highlight changes of the deliverable since its preliminary version (D1.3), alongside info on how the project reviewers' comments on that preliminary document have been taken into consideration in this version.
Section 2	The entire Section has been updated to reflect the work carried out in the context of Task T1.2 since D1.3. The Section contains the outcomes of a novel survey with platform stakeholders both internal and external to the Consortium, and the implications of the elicited stakeholders' expectations on the technical requirements defined in D1.3.
Section 3	The Section has been updated to reflect upon the entire lifecycle of Task T1.3.
Section 3.1	Section 3.1 now revisits the architecture as defined in D1.3, to act as a reference Section for the changes documented in the latter chapters of the deliverable. Its equivalent in D1.3 is that document's Section 3.2.

Section 3.2	Section 3.2 now summarises the approach to refining the architecture, reporting on the processes undertaken by Consortium partners during the second year of the project. Each sub-Section has hence been updated accordingly (including revised Section titles, where appropriate). Its equivalent in D1.3 is that document's Section 3.3.
Section 3.3	Section 3.3.1 now includes a summary of the Network Application (formerly referred to as a NetApp[1]) definition under the ICT-41 joint initiative, and describes implications on the 5G-EPICENTRE approach to Network Application experimentation for PPDR. Section 3.3.2 recapitulates the contents of D1.3 Section 3.1, albeit with additional terms used throughout the document.
Section 3.4	Section 3.4 now corresponds to the contents of D1.3 Sections 3.5.1 – 3.5.3, updated to reflect the final state of the 5G-EPICENTRE architectural stack.
Section 3.5	Section 3.5 now corresponds to the platform component contents of D1.3 Section 3.5.4. It presents the final version of the functional descriptions for each architectural block presented in this document's Section 3.4.1.
Section 3.6	Section 3.6 is a novel Section (D1.3 numbered a total of 5 sub-Sections). It corresponds to the testbed component contents of D1.3 Section 3.5.4, elaborating on the final version of the functional description of each block in the testbed reference framework.
Section 3.7	A novel Section 3.7 corresponds to the contents of D1.3 Section 3.5.5. It presents the final Information view of the architecture in the form of an updated Information Flow Diagram (IFD), complete with indicative sequence diagrams for common platform usage scenarios.
Section 3.8	A novel Section 3.8 corresponds to the contents of D1.3 Section 3.5.6. Whilst that Section in D1.3 briefly presented the view, this version provides a deployment diagram and identifies the hardware nodes where components are to be installed for the operational deployment of the platform.
Section 3.9	Section 3.9 is a novel Section, that presents an additional viewpoint for the architecture. Its closest counterpart in D1.3 is that document's Section 3.5.7.
Section 4	The Conclusion Section has been updated to summarise final insights on the entire Task T1.2 and T1.3 lifetime.
Annex I	The Annex now presents the second round electronic survey questionnaire, which was used to derive the updated contents of Section 2.
Annex II	The Annex now recapitulates the contents of D1.3 Section 3.4, which are a significant supplement toward realizing how the testbeds integrated a Kubernetes infrastructure.

[1] The term 'NetApp' is trademarked by the company *NetApp, Inc.*, as described in https://www.netapp.com/company/legal/trademarks/. The term 'NetApp' should therefore not be used within 5G PPP, *e.g.*, from 5G-I TB, 5G-I SB, a 5G PPP WG or a 5G PPP project perspective, unless one is referring to the NetApp company. To avoid confusion, the term 'NetApps' should also not be used. Instead, the recommendation is either to use the formulation 'Network Applications', (alternatively 'network applications'), or to use 'Network App' or 'Network Apps', as a shorter version.

Annex III	The Annex now presents the template used for bottom-up re-elaboration of the functional components depicted in the 5G-EPICENTRE architectural stack.

1.2.1 Adherence to reviewers' comments and recommendations on D1.3

In addition, efforts were spent to address comments made by the project monitors with respect to D1.3 toward the betterment of the present document. More specifically, Table 3 below, aims at explicitly clarifying how review report comments from the first project review have been addressed.

Table 3: Amendments made to address each reviewers' comments

Review comment(s) **(as provided by the reviewers)**	**5G-EPICENTRE Adherence and Document Update (short reply and reference to the chapter that details the reply)**
"The deliverable is of good quality but sometimes verbose with lengthy descriptions. It is advisable that deliverables are more focused avoiding too much high-level information, repetitive content, lengthy descriptions of processes. Please streamline the deliverables and focus on key content and contributions.".	Effort has been spent toward improving the overall focus of this deliverable in order to concentrate on project activities, particularly those that have taken place since the delivery of D1.3. Further, content that Consortium partners believe reflects important work on the Cloud-native evolution of the testbeds has been moved to an Annexed Section, thus avoiding distracting the reader with verbose descriptions with high-level information (potentially of tutorial nature to the novice reader).
"Related to the questionnaire in Annex I, it seems difficult for an external partner to be able to answer many of the questions without more information about the experimental facility and its capabilities than that provided in the questionnaire. Also, some questions are difficult to understand or quite technical for an external SME that is focused on PPDR but not familiar with 5G capabilities. This could hinder the provision of valuable information from end-users. Did you provide more information or an overview of the platform capabilities?"	A second round survey was carried out to iteratively revisit the platform expectations from the viewpoint of its intended users (and other stakeholders, see Section 2.2). Having given shape to the preliminary version of the architecture in D1.3 (a publicly available resource from the 5G-EPICENTRE website[2]), survey monitors were able to extract valuable information that further reinforced validity of the technical approach (see Section 2.3). With a clearer view of the architecture, external stakeholders were also better positioned to inform the priorities set to the original list of requirements (see Section 2.4).
"Conducting the survey and using it to identify requirement is a good activity. Do you plan to achieve all these requirements? And how do you plan to embed these requirements in the development of the platform?"	With the initial set of requirements being deemed comprehensive enough to flesh out the platform functionalities and features, requirements' tracking has been employed throughout the course of the project to ensure that technical deliverables (*e.g.*, D3.1, D3.3, D4.4) reflect that these requirements are indeed implemented as platform component capabilities. An early indication of the achievement of at least 90% of the requirements list is given in D4.4 (corresponding to project milestone MS4). The Consortium intends to deliver proof of

[2] https://www.5gepicentre.eu/public-deliverables/

	achievement of at least 99% of the requirements listed in the context of D4.5 (due in M30).
“Also, the project’s platform relies on the use of 5G testbeds provided by partners and other 5G PPP projects. Some of these requirements can affect more these underlying 5G testbeds. Does the project have the capacity to modify these 5G testbeds to comply with some of the identified requirements if the related projects finish before the end of 5GEPICENTER?”	Testbed heterogeneity has been addressed through the envisioned integration of a Kubernetes infrastructure in each facility (see Section 3.6.1, and Annex II for a reference implementation of the approach), enabling the testbed to register in the 5G-EPICENTRE federation as a member cluster, cross-orchestrated by the Karmada cross-cluster management solution (see Section 3.5.3, as well as the Section 3 of D4.4). 5G Core and 5G Radio Access Network capabilities are exposed to these clusters by means of the revamped Network Application ecosystem (See Section 3.3.1) and platform Network Applications introduced in this document (Section 3.6.3). These will be further elaborated in D4.2.
“The list of functional requirements is quite comprehensive. How many of these requirements depend on the underlying 5G testbeds brought by different partners and how many of these are already satisfied?”	Depending on the technological maturity of the underlying 5G testbeds, some requirements were deemed to be in much better progress for some of the testbeds (*e.g.*, integration of Kubernetes), while others required substantial implementation effort. The list of requirements re-prioritized in Section 2.4, hence refers to the 5G-EPICENTRE solution from the perspective of the platform, treating underlying resources as a cross-cluster deployment environment, agnostic to the testbed heterogeneous implementation (see the reference framework describing each testbed from the perspective of the platform in Section 3.4.2).
“The platform and building blocks are well explained and with quite detailed information for being a M6 activity. It is not completely clear how many changes from the architecture envisioned at project preparation result from the surveys, and how such changes can affect the project execution since they were difficult to be anticipated at project preparation.”	We have elaborated the three iterative versions of the 5G-EPICENTRE architecture (GA, D1.3 and D1.4) in the introductory paragraph in Section 3, to capture its evolutionary approach. Specifically for this document we have attempted to highlight the changes from the architecture reported in D1.3, both by recapitulating the latter in Section 3.1 (compared to the final version presented in Section 3.4), as well as by referring to changes and novelties introduced throughout the descriptions of the different architectural viewpoints (Sections 3.5 – 3.9).

2 5G-EPICENTRE experimentation requirements

This Section contains the results of the second round of research carried out to determine the requirements of the final platform. The second round of the survey was created through an online questionnaire, assisted by a selection for an additional optional oral interview.

The questionnaire is included in Annex I, and consists of 5 sections, starting with the notification of compliance and consent under the General Data Protection Regulation (GDPR). The second section intends to gather the demographic data of the survey, while the third includes questions from the 1st edition of the questionnaire, with some minor changes mainly for convenience. The fourth part has introduced new questions concerning the requirements that emerged from the first round. Finally, the fifth part collects contact information.

2.1 Method and procedure

The second round of the survey was created through an online questionnaire, posted on an appropriate platform that ensures full compliance with the GDPR. Participants were given complete freedom, with many questions being optional, providing them the option not to reply to one or more if they so wished. The questionnaire was published and distributed through various communication channels, including the project's social media accounts and 5G-PPP Technical Board mailing list. Target groups were representatives of organisations belonging to the PPDR community and also suitable entities with technical knowledge and know-how around 5G technologies. The survey lasted 5 weeks and gathered responses from 31 different participants. Upon completion, the results were collected, analysed and presented in an easy-to-understand form.

2.2 Survey participants and demographics

A total of 31 people responded to the call and filled out the online questionnaire. The responses came from Spain (20,39%), Greece (13,95%), Germany (11,63%), Italy (9,30%), Portugal (8,14%), France (6,98%) Belgium and Norway (5,81% each). Other countries achieved representation percentage lower than 5% each (Cyprus, Slovenia, Austria, Switzerland, Romania, Hungary, United Kingdom, United States of America, India, Turkey, and United Arab Emirates, see Figure 1).

Figure 1: 5G-EPICENTRE 2nd round survey response distribution.

The survey seemed to have attracted new audience, as about three quarters (76%) of the respondents had not participated in the 1st survey, although other members from their organisation may had done so (Figure 2).

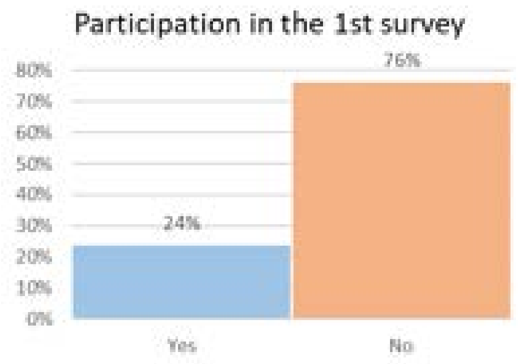


Figure 2: Percentage of participation in the 1st round survey.

The vast majority of participants came from the private sector, with a percentage of 56,25%, followed by the research sector with a percentage of 31.25%. The public sector was represented by 6,25% of the participants while the regulatory bodies and the academia were presented by 3,13% each (Figure 3).

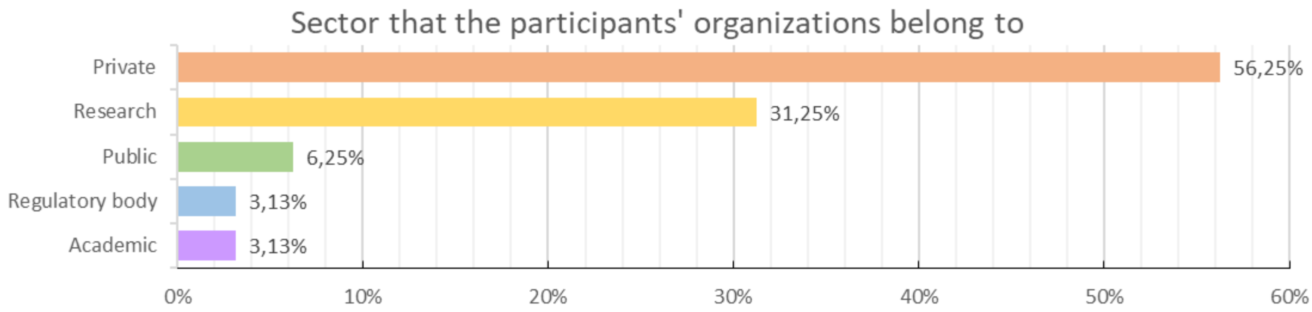


Figure 3: Distribution of 2nd round survey respondents across sectors.

The survey seems to have gathered a more smoothly distributed sample from representatives of organizations of various sizes (Figure 4), in contrast to the first round of the survey, where respondents mostly came from small businesses of up to 50 people (52,78%) or large organizations with a staff of more than 1000 people (30,56%).

Figure 4: Distribution of 2nd round survey respondents based on organization size.

A better-balanced sample seems to exist in the role played by each respondent within his/her company (Figure 5). Although the technical drivers are still the dominant group with 42% (previously 64%), followed by domain experts with 25% (previously 36%). Nineteen percent (19%) of the sample consist of strategical drivers (previously 22%), while 14% consists of business drivers (previously 33%).

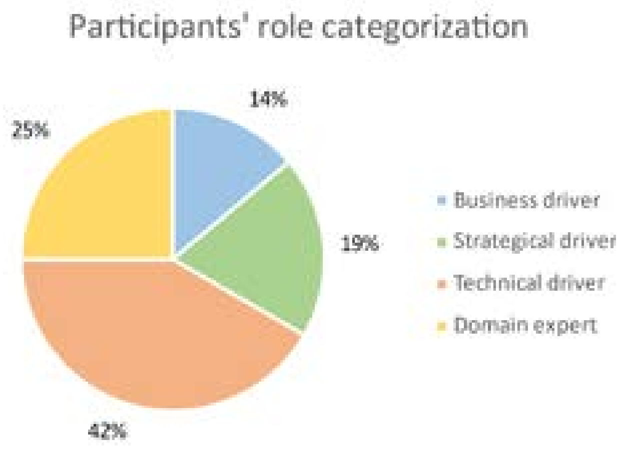


Figure 5: Distribution of 2nd round survey respondents based on role in the organization.

Regarding the management level of each of the survey participants (Figure 6), 38,71% says they are a researcher and 22,58% say they are engineers. Administrative officers (*e.g.*, office managers, supervisors, department managers, team leaders, project leaders, *etc*.) follow closely with 19,35%. Lower, with a percentage of 9,68% each are the operating officers (*e.g.*, general, plant, regional, divisional, or other managers *etc*.) and the executive officers (*i.e.*, CEO, CTO, CFO, COO *etc*.).

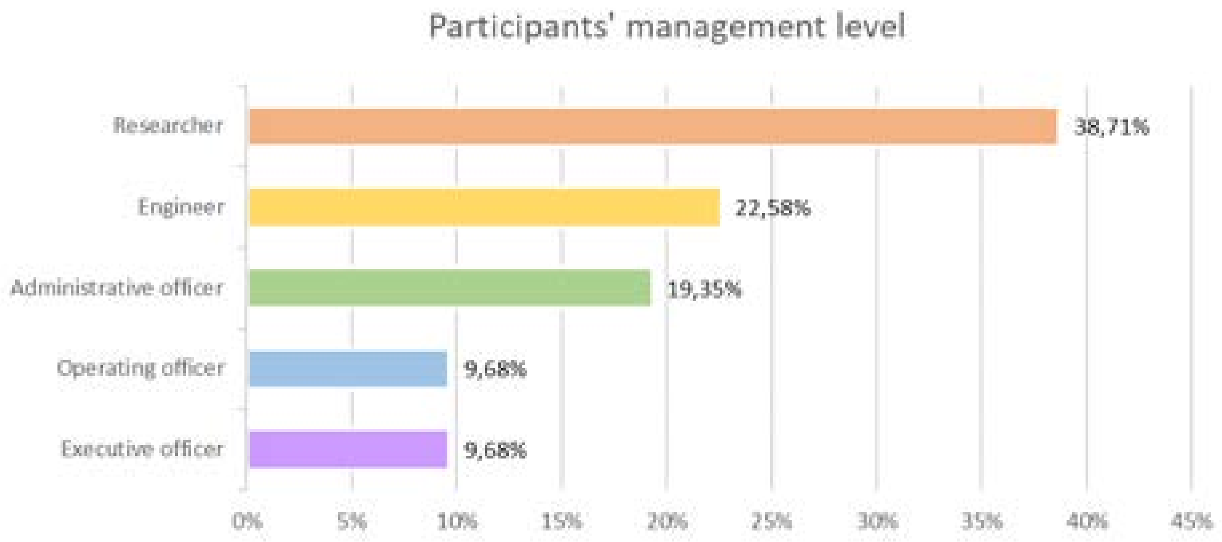


Figure 6: Distribution of 2nd round survey respondents based on management level in the organization.

Regarding their experience, more than half of the participants have 10 years or more of work experience (approximately 58%), while those with medium experience (2-10 years) achieve a combined rate of approximately 32%. Inexperienced participants (0-2 years) combine are about 10% of the total sample (Figure 7).

Finally, regarding the roles performed by the participants' organizations, 38,89% are engaged in the provision of applications and services on PPDR-related solutions, while 27,78% are operators of research or other experimental platforms. Telecommunications network operators comprise 16,67% of the sample with close behind selected “Other”, but without providing additional information about their organisation’s role (Figure 8).

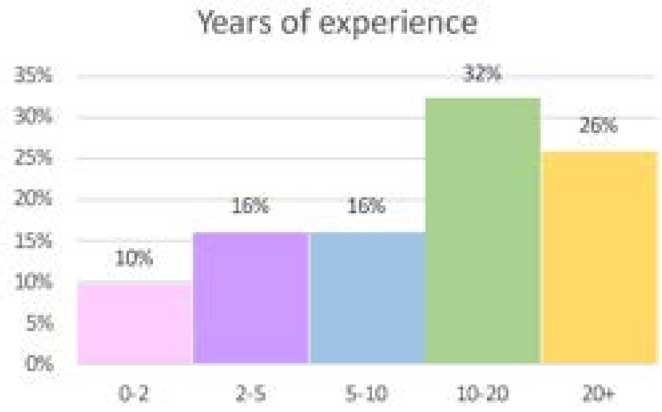


Figure 7: Distribution of 2nd round survey respondents based on work experience (in years).

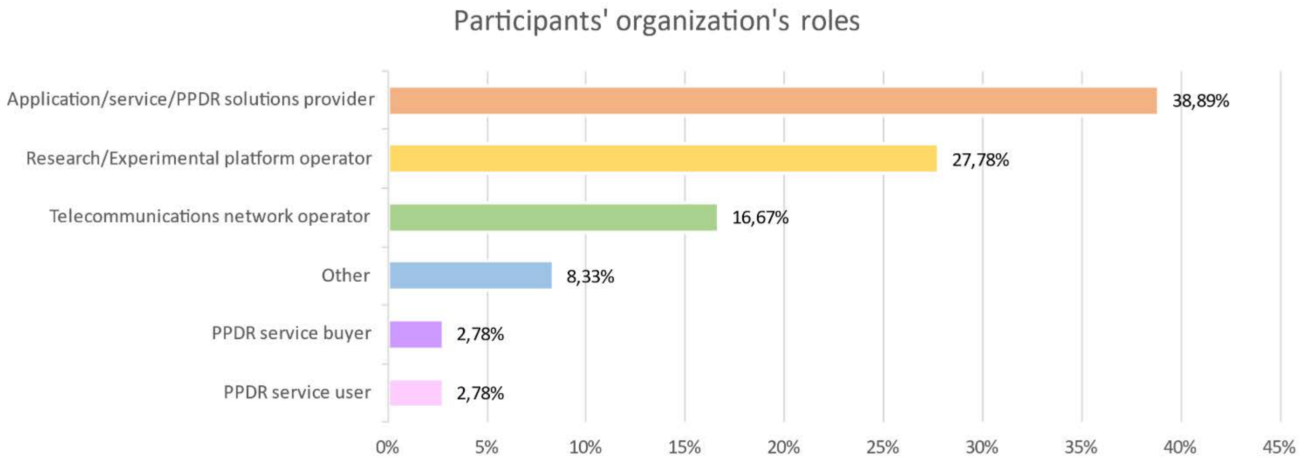


Figure 8: Distribution of 2nd round survey respondents based on organisational roles.

One in 10 is an administrator of experimental or research platforms. Little participation is observed by organizations that are PPDR service buyers as well as buyers, with 2,78% each.

2.3 Survey results – stakeholder expectations

Participants were called to rate, on a scale from 1 (totally disagree) to 5 (totally agree), if they consider that PPDR services are limited by current networks and/or existing solutions. It is important to note that no-one disagreed completely on the matter, and only 9% disagreed partially. The majority of participants (50%) were neutral on the matter, and a total percentage of 41% were on the "agree" side, with 27% partially agreeing and 14% agreeing completely (Figure 9).

Asked to name which Key Performance Indicators (KPIs) or PPDR service aspects they believe are mostly limited; the participants provided the following answers (replies listed in alphabetical order):

- Adaptability.
- Applications have not matured enough to reach sufficient edge/cloud adaptation via containerisation.
- Automated deployment.
- Availability.

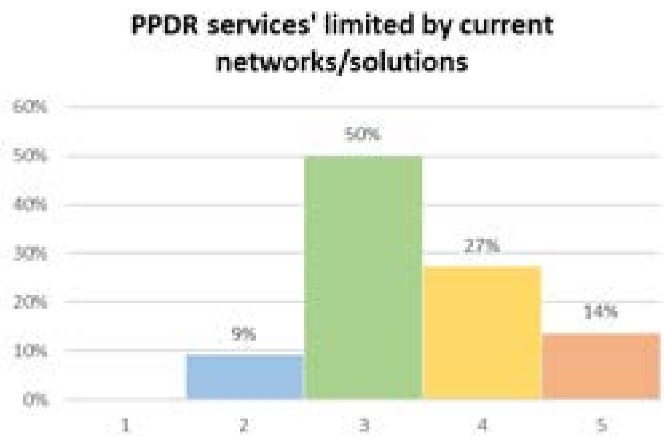


Figure 9: Distribution of responses to the question “are PPDR services limited by current networks and/or existing solutions?” [1: totally disagree; 2: partially disagree; 3: neutral; 4: partially agree; 5: totally agree].

- Broadcast services.
- Coverage in rural areas.
- Cross-border operations.
- Different technologies (or its implementations) in different countries/regions.
- End-to-end Quality of Service (QoS), together with slicing support (including Radio Access Network slicing and cloud-transport integration).
- Flexible traffic prioritization and access to network resources.
- Fluctuation of network characteristics can prove fatal for some applications.
- High-quality real-time video streaming.
- Implementation of QoS.
- Lack of edge node servers (also cause increases in latency).
- Lack of real-time forecast, support, monitoring and/or supervision of an incident.
- Large scale private networks based on public networks.
- Latency x3.
- Low bandwidth.
- Need to upgrade to broadband solutions that enable new advanced functionalities.
- Network does not provide sustainable connections in several areas that can maintain an acceptable Quality of Experience (QoE).
- Network resource deployment.
- No data/digital applications.
- No dynamic regarding UE’s integration.
- Overall efficiency, performance, and usability.
- PPDR users are sometimes too dependent on single suppliers, which makes the reform process required by the sector expensive.
- Proper network slicing for PPDR end-users.
- Reliability, especially under stress conditions.
- Services availability, prioritisation, broadband direct mode.
- Throughput.

On the same scale, survey participants were also asked if they currently test the PPDR services they either develop or procure, in either operational environments or experimentation facilities (before moving them to production). Although during the previous survey almost half (44,45%) of the participants were on the disagreement side, none were observed this time (Figure 10).

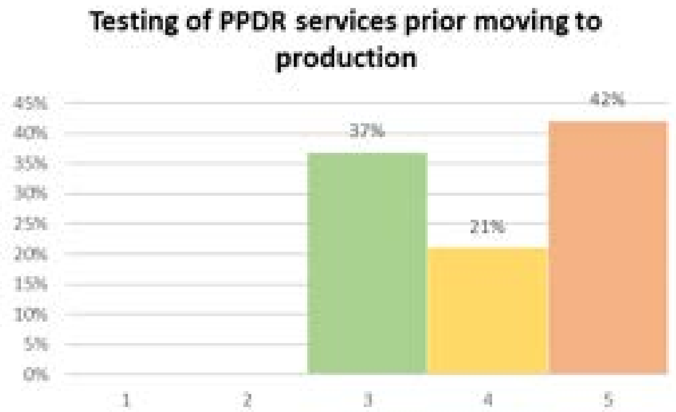


Figure 10: Distribution of responses to the question "are you testing PPDR services prior to moving them to production?" [1: totally disagree; 2: partially disagree; 3: neutral; 4: partially agree; 5: totally agree].

Concerning the focus of their PPDR services, participants we asked to choose from a multiple-choice list, while also having the capability of providing additional ones. The results showed that most participants' have their services oriented towards mission critical communications (42,11%). Services testing came second with 21,05%, followed by large-scale-events management and law enforcement with 10,53% each. Command & Control, emergency medical services were less represented in the sample with 5,26% each. Participants that chose "Other" were also 5,26% of the sample, but did not provide additional information on their PPDR services focus (Figure 11).

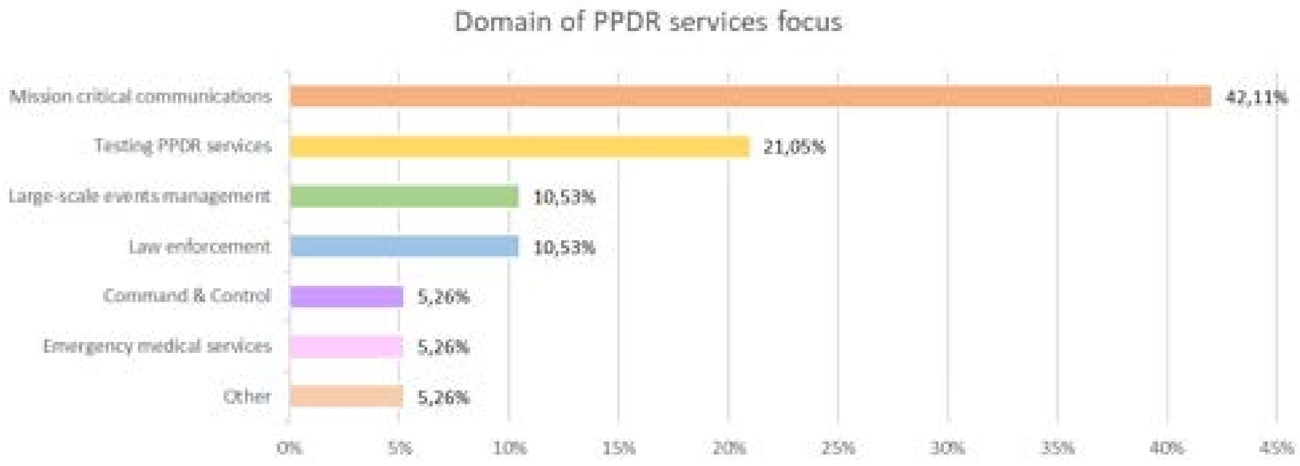


Figure 11: Distribution of responses to the question "are you testing PPDR services prior to moving them to production?" [1: totally disagree; 2: partially disagree; 3: neutral; 4: partially agree; 5: totally agree].

For the benefits that the survey's participants anticipate reaping by utilizing the 5G-EPICENTRE experimentation platform, the "Test performance in a near operational environment" was the dominant choice, with a percentage of 23,53% (Figure 12). More specifically:

- Top tier choices
    1. Test performance in a near operational environment (23,53%).
    2. Test KPIs (19,12%).
    3. Collaborate with other providers to create a value chain (17,65%).
- Medium tier choices
    1. Provide the ability to allow end-user to run their tests (11,76%).
    2. Test performance in simulated extreme conditions, (10,29%).
- Lower tier choices (less than 10%)
    1. Provide the ability to define the business collaboration model between involved parties (5,88%).
    2. Provide the ability to the operators to test onboarding/operating of the service" (4,41%).
    3. Test compliance to standards (4,41%).
    4. "Other" (2,94%) was accompanied with the suggestion of showing through empirical evidence how communication solutions over 5G networks can offer performance guarantees to the PPDR sector.

On the multiple-choice question concerning the functions that such an experimentation facility should provide, the results (Figure 13) were the following:

- In the 8-10% margin (most popular)
    1. Visualisation of KPIs measure and automated analysis vs required performance/advanced reporting (9,22%)
    2. Friendly user interface guiding to perform test cycle processes (9,22%)
    3. Traffic generator (8,51%)
    4. Ability to define test cases and KPIs (8,51%)
    5. VNF/NETApps repository and ability to use (8,51%)
- In the 6-8% margin
    1. Slices parameters definition (7,80%).
    2. Service onboarding/parametrisation functionality (7,80%).
    3. Service setup latency (7,09%).
    4. Simulate extreme operation conditions (physical and digital) (7,09%).
    5. Network resources repository and ability to use (6,38%).
- In the 4-6% margin
    1. Fault tolerance of a service and self-healing (5,67%).
    2. Scaling ability inspection (4,96%).
    3. Ability to test in various 5G releases (4,96%).
- In the 0-4% margin
    1. Security inspection and security breach test (2,84%).
    2. Interference inspection (0,71%).
    3. Standards compliance inspection (0,71%).

On the open question about what kind of services the experimentation platform should provide, the participants provided the following answers:

- Documentation and FAQ (including a minimum requirements list for various features).
- Service exploration.
- Training and/or tutorial.

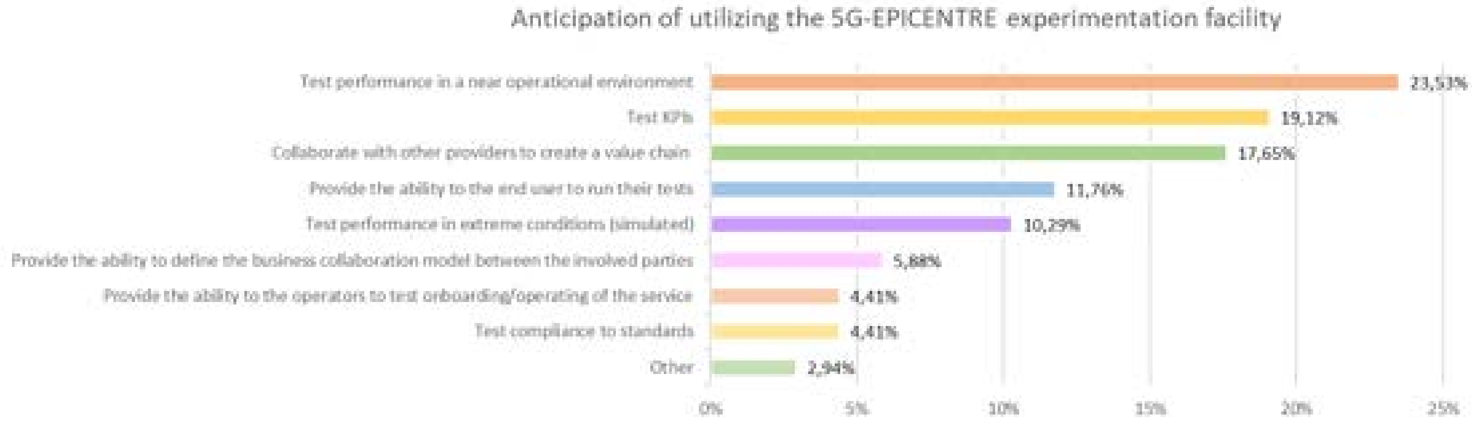


Figure 12: Distribution of responses regarding anticipation of utilising 5G-EPICENTRE experimentation platform

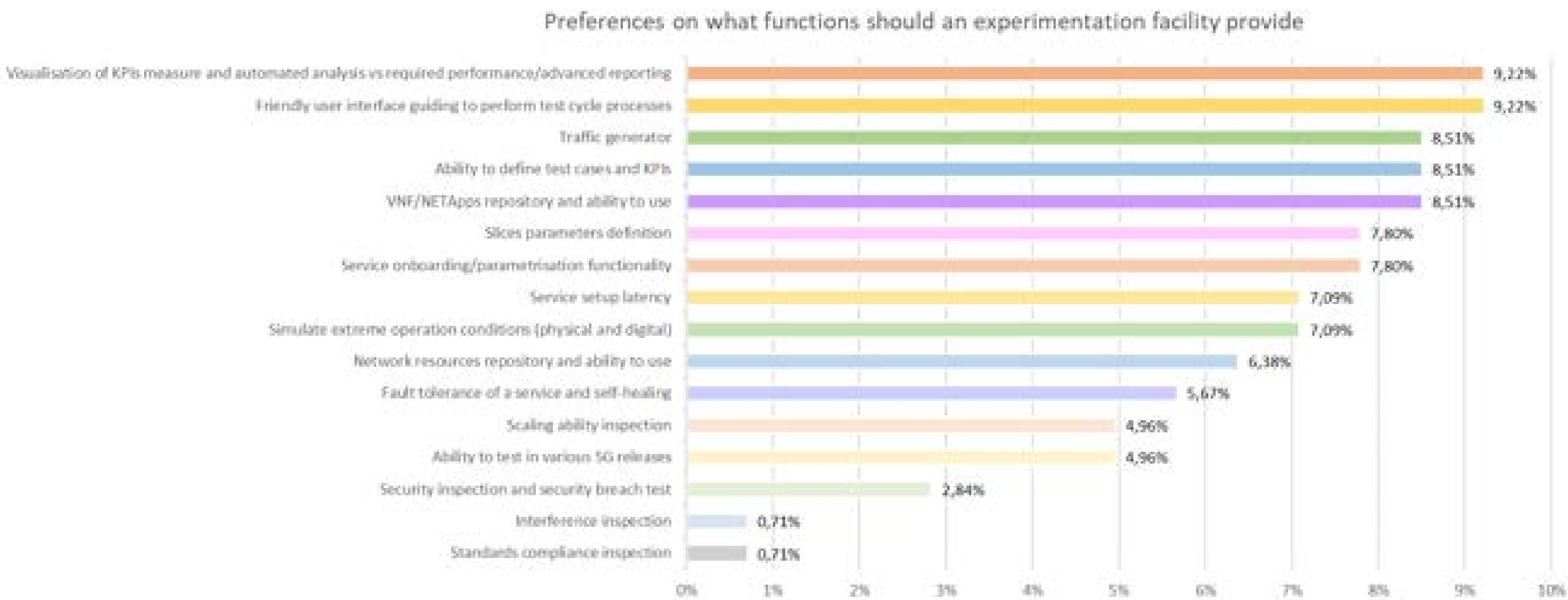


Figure 13: Distribution of responses regarding preferences on what capabilities the 5G-EPICENTRE experimentation platform should expose.

- Scheduling, experiment planning and resource booking.
- Ability to build testing scenarios and define testing KPIs.
- Support for 3rd party application deployment (supported by well-defined APIs).
- Monitoring:
  - Network monitoring (GUI-visualization, real-time, history, *etc*.).
  - Resource monitoring.
  - Radio monitoring.
- Results reporting:
  - Measurements (including KPIs).
  - Statistics.
  - Historical data.
- Billing.
- Customer support (including ticketing).
- Community.
- Consultancy to define testing scenarios and test cases tuned with the requirements of the use case under test.
- Physical and virtual facilities (with remote access) with Visual Control Interface.
- Service creation and deployment
- A list of potentially connected edge nodes should include geolocation/approx. latency & bandwidth / CPU / GPU / RAM specifications, along with availability of the edge node.
- Support integration of experimenter's equipment (UEs).
- Automatic migration of Network Applications between PDU session endpoints as a result of mobility.
- Traffic load simulator.

On the opposite side, about what things should be avoided in the implementation of the facility, the participants reported the following answers:

- Downtime.
- Taking a lot of time to run a test.
- Too much complexity (*i.e.*, user non-friendliness) or assumptions that it is easy to use.
- Complex interface and background in mobile communications.
- Necessity for every experimenter to configure the network "from scratch" (some configs should be already prepared).
- Being ambiguous, or being rigid in what can and cannot be tested (*i.e.*, if certain issues or incompatibilities are known, they should not be hidden. If clarity of several description is insufficient, this creates confusion for the end-users).
- Unclear installation/deployment conditions.
- Unclear testing capabilities.
- Not being transparent.
- Unpredictability about what will happen next.
- Failures without explanation.
- Very fixed experiment setups.
- Avoid narrow processes.
- Offer limited variability and possibilities to the experimenter (the more the options, the more attractive the solution will be).
- Limitation customisation degree or limited adaptation to the new 5G environment.

Cross-domain experiment support is a feature that the vast majority of participants voted for (74%), with another 21% not being sure about it (5% voted “No”, see Figure 14).

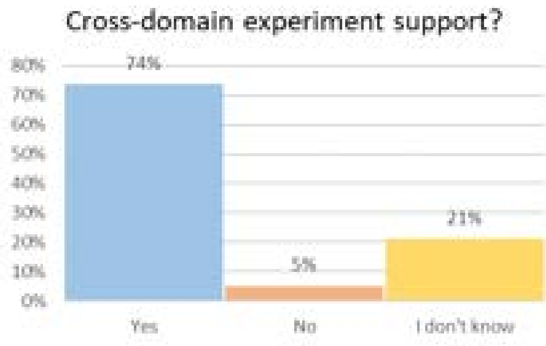


Figure 14: Distribution of responses regarding whether the platform should support cross-cluster experimentation.

As for the expectations of the respondents towards network slice configuration during setup phase, almost half of the participants (47,37%) prefer to be able to fine tune parts of the network slice by invoking appropriate Network Applications. About one third of the survey’s participants (31,58%) would better prefer for 5G-EPICENTRE experts to perform network slice configuration. 10,53% would like to be able to manage the lifecycle of the involved Virtual Network Functions (VNFs) and fine tune their operation while 5,26% want to be able to manage the Virtual Machine (VM) hosting the VNFs, as well as manage and fine tune these VNFs (Figure 15).

Regarding the way that the facilities resources reservation will be performed, in order to conduct a number of experiment iterations over time, almost all participants (94,73%) chose an online option (Figure 16): 73,68% opt in for booking provided by 5G-EPICENTRE portal as a “one-stop-shop”, while 21,05% prefer that resource booking should be provided by each experimentation facility infrastructure separately. Although none of the respondents chose to have this procedure offline, a 5,26% did not declare any preference.

The reporting of the experiments’ results is mostly demanded to be delivered by standard reports, generated by the platform (63,16%), with a little more than a quarter (26,32%) of the respondents prefer that the platform should support customised reports. A 15,79% would like to be able to customise security schemes. The remaining 5,26% provided no opinion on the matter (Figure 17).

For their expectations towards the security schemes during runtime, more than half of the respondents (52,63%) would prefer for 5G-EPICENTRE experts to handle them. A combined 42,11% would go for more flexibility: 26,32% want to be able to define standard security schemes and 15,79% to be able to customise them. The remaining 5,26% provided no opinion on the matter (Figure 18).

Concerning their expectations towards the cross-layer privacy network, the majority (45%) prefer once again to have the 5G-EPICENTRE experts to handle it, and 25% would like to have the ability to proceed with configurations. The remaining 30% provided no opinion on the matter (Figure 19).

2.4 Survey results – technical requirements

For the next questions of the survey, that concern the functional and non-functional requirements of the platform, the MoSCoW prioritisation method (Must-have, Should-have, Could-have, Will-not-have) was partially utilised. More specifically, the “Will-not-have” part was purposefully omitted, as it was not in par with this Section’s nature. The requirements listed in each case, derived from the first round of the survey, and listed in paragraph 2.2 “5G-EPICENTRE platform requirements”, of D1.3. The survey’s participants were asked to choose from multiple-choice lists, the necessity of each requirement in all cases. The platform functional and non-functional requirements listed in D1.3 were re-prioritized according to the results shown in Figure 20 – Figure 25.

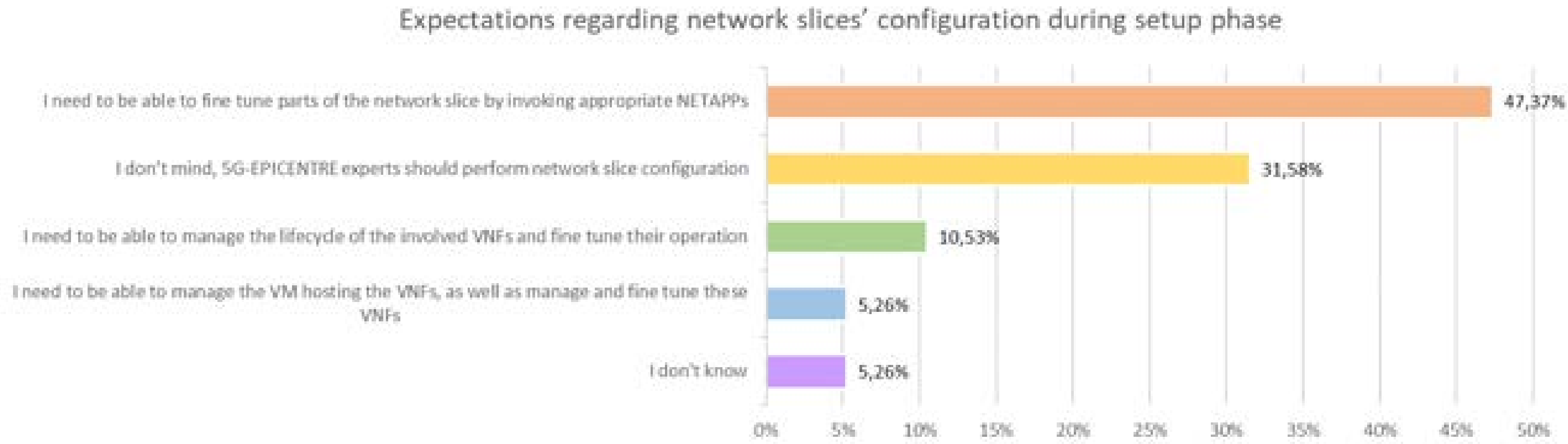


Figure 15: Distribution of responses regarding expectations on network slices' configuration capabilities exposed during the experiment setup.

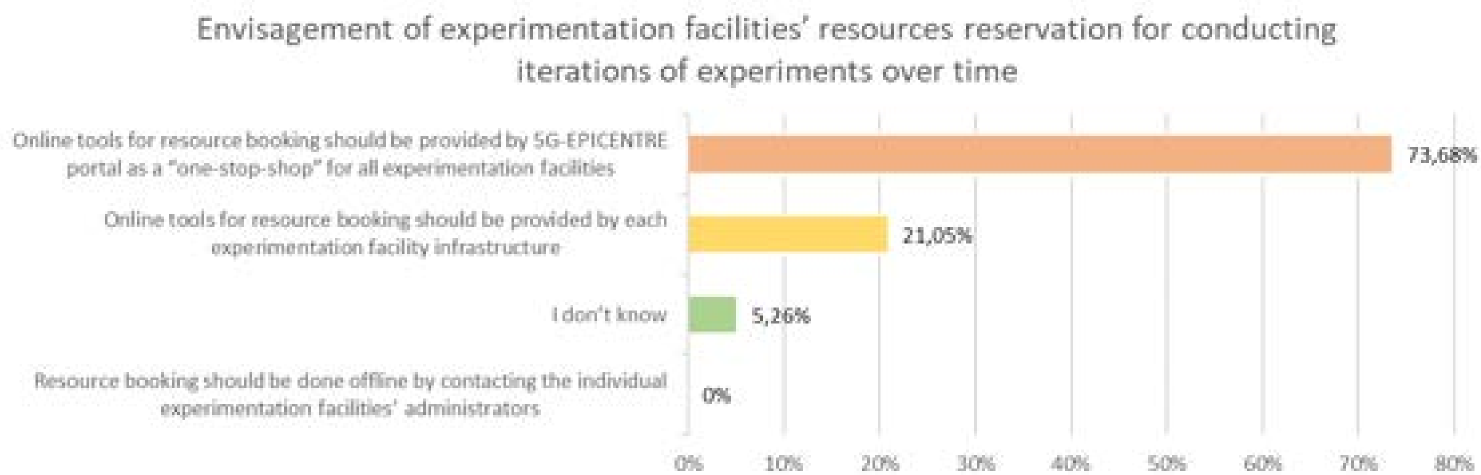


Figure 16: Distribution of responses regarding resources' reservation for carrying out several experiment iterations over time.

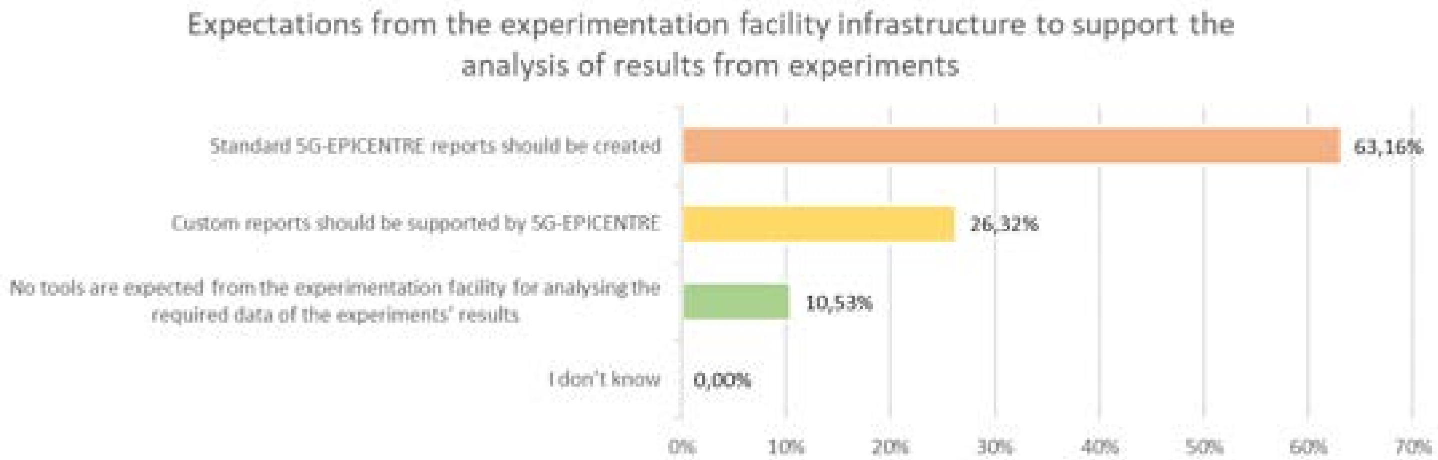


Figure 17: Distribution of responses regarding expectations for the platform to support analysis of results from experiments under test.

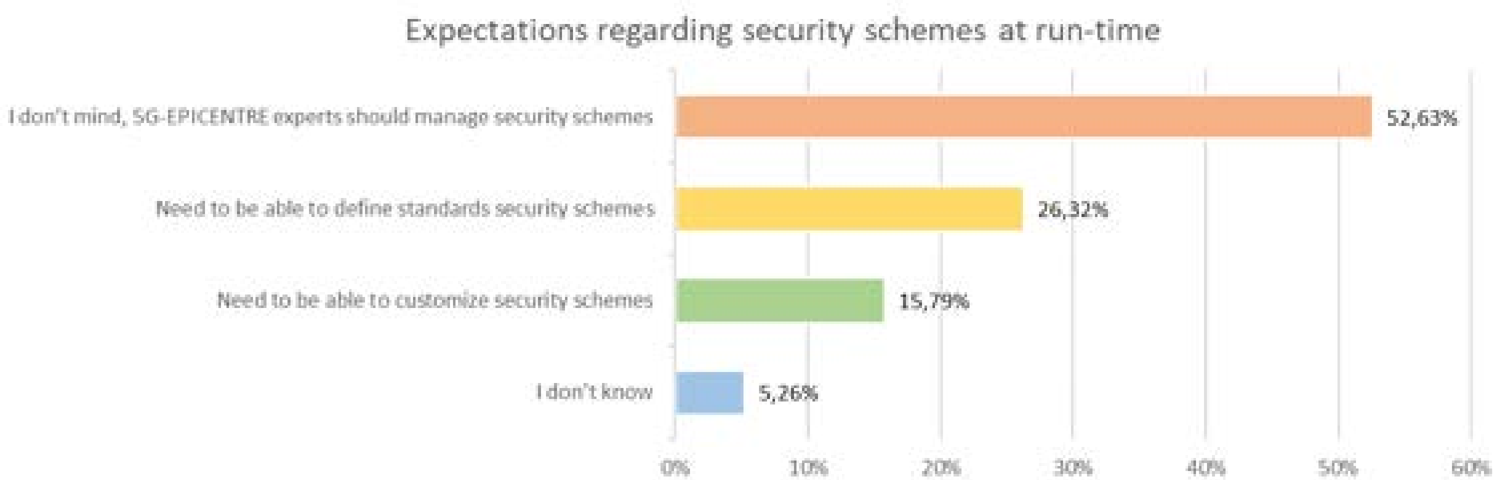


Figure 18: Distribution of responses regarding expectations for the platform to provide security schemes at runtime.

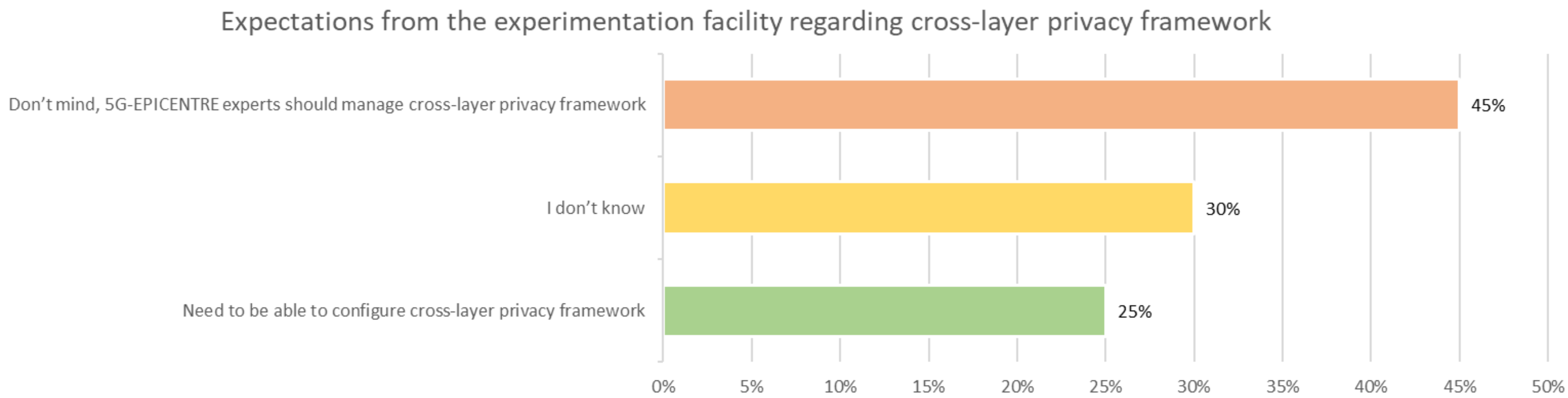


Figure 19: Distribution of responses regarding expectations for the platform's cross-layer privacy network.

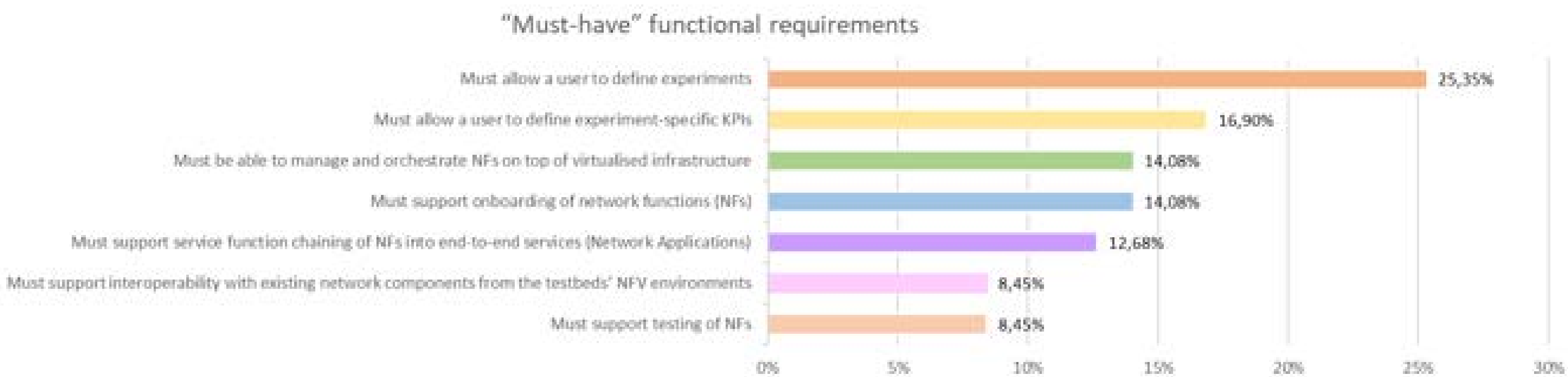


Figure 20: "Must-have" Functional Requirements of the D1.3 platform requirements specification re-prioritization.

Figure 21: "Should-have" Functional Requirements of the D1.3 platform requirements specification re-prioritization.

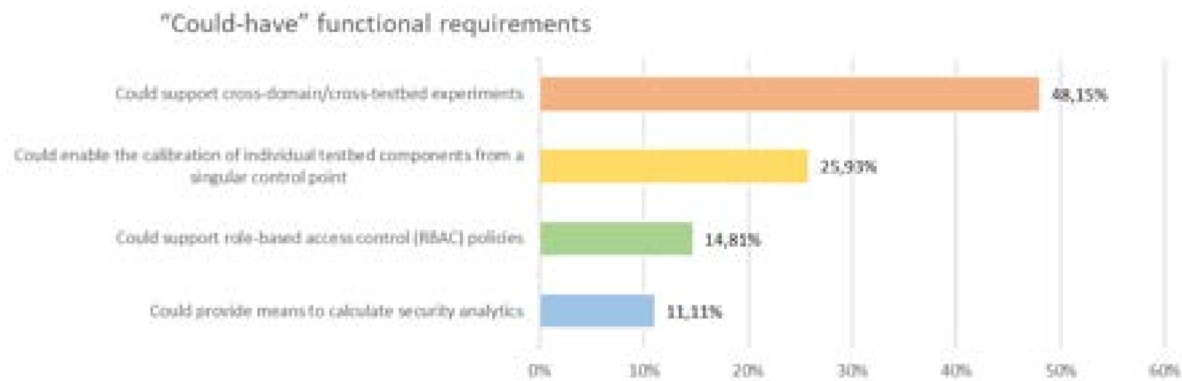


Figure 22: "Could-have" Functional Requirements of the D1.3 platform requirements specification re-prioritization.

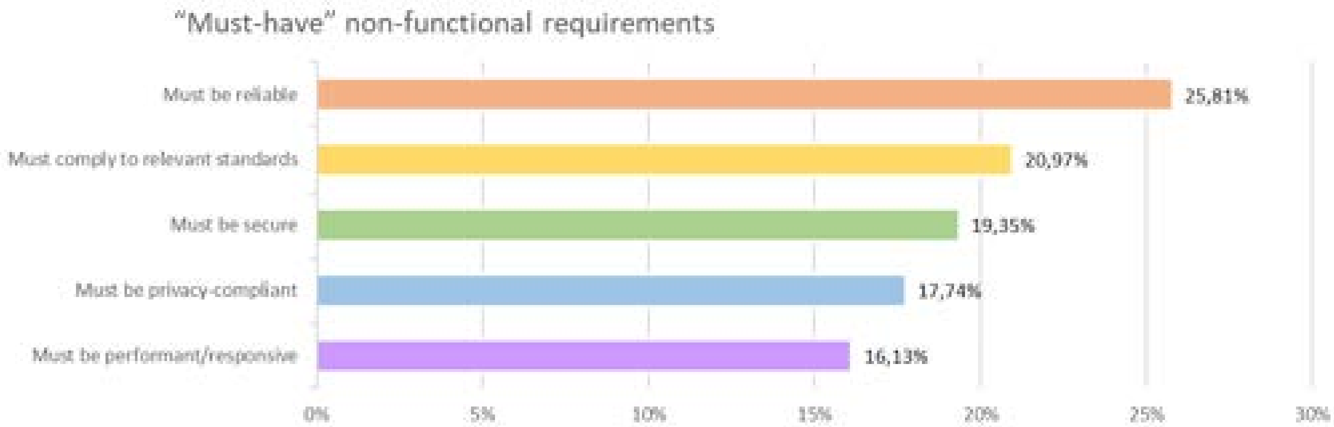


Figure 23: "Must-have" Non-Functional Requirements of the D1.3 platform requirements specification re-prioritization.

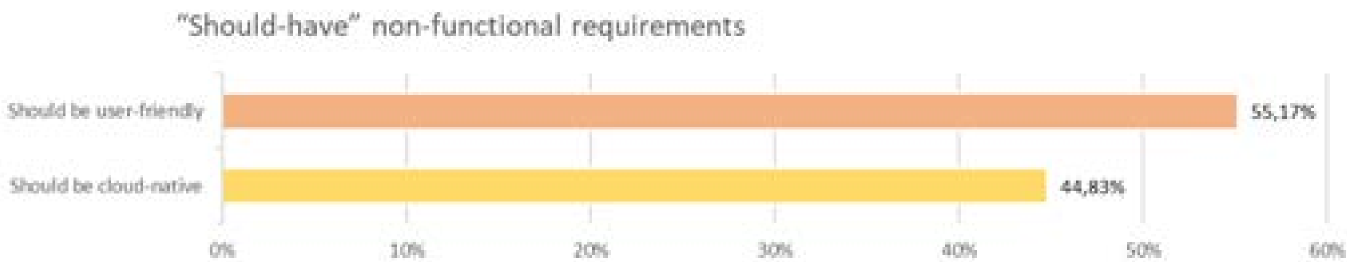


Figure 24: "Should-have" Non-Functional Requirements of the D1.3 platform requirements specification re-prioritization.

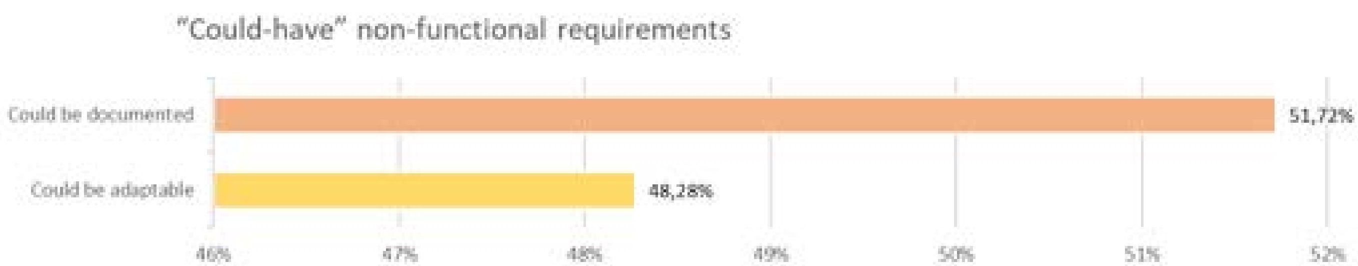


Figure 25: "Could-have" Non-Functional Requirements of the D1.3 platform requirements specification re-prioritization.

3 5G-EPICENTRE platform architecture

In this Section, the conclusive 5G-EPICENTRE platform architecture is elaborated. Thereby, this Section contains a final overview of all the functional components, along with their individual internal architectures, interfaces and information flow diagrams (where applicable). The deliverable introduces revisions to the content reported in D1.3, which had served as a guide for the development activities taking place in Work Packages (WPs) 2 to 4, toward the implementation of the architecture building blocks and their eventual integration into the preliminary (D4.4) and final (D4.5) experimentation platform versions.

In general, the 5G-EPICENTRE platform architecture is the result of a recursive approach (described in Section 3.2), that has resulted in three distinct versions:

(i) The first iteration of the platform architecture, which was drafted during project preparation activities (at the proposal level), and which foresaw the development of the 5G-EPICENTRE experimentation platform. It was described in detail in the Annex 1 (part A) of the 5G-EPICENTRE GA (Description of Action – DoA) and its purpose was to be consulted as a reference model for the ensuing preliminary version of the concrete 5G-EPICENTRE platform architecture.

(ii) The second iteration corresponds to the contents reported in deliverable D1.3 (M6). It envelops the precursory organisational structure of the 5G-EPICENTRE concrete modules and subsystems toward giving concrete shape and substance to the different functional elements being developed across the technical WPs (WP2, WP3 and WP4). Thereby, the aforementioned WPs were able to draft implementation and integration roadmaps for the preliminary versions of each of their respective outputs.

(iii) The third (and final) iteration corresponds to the contents reported in the present document. It represents the penultimate concrete architecture that project partners are implementing toward delivering on the objectives and scope of the 5G-EPICENTRE project.

The structure of this Section will be as follows: Section 3.1 briefly outlines the preliminary version of the 5G-EPICENTRE platform architecture, so as to better illustrate the changes introduced in this final version. Section 3.2 elaborates on the overall approach taken for the specification of the 5G-EPICENTRE platform architecture, including the description of the architecture definition phased processes and the activities of the dedicated online architecture design workshops, along with their outcomes. Section 3.3 then elaborates on general concepts used throughout this Section to help the reader better comprehend the architecture. Finally, Section 3.3 addresses the final fundamental structure of the 5G-EPICENTRE architecture, presenting the different architectural views (functional, information, deployment), which focus on: i) defined system elements as building blocks, their relationships, interfaces and internal processes; ii) the way information is stored, managed and distributed within the specified architecture; and iii) the physical characteristics and constraints of the architecture.

3.1 Preliminary platform architecture recapitulation (D1.3) [M6]

The preliminary 5G-EPICENTRE platform architecture was delivered with D1.3 in the context of Task 1.3 (Figure 26). Its main purpose was to serve as guidance for the development of the functional components in the technical WPs by outlining the roles and responsibilities of each identified component, without however implying specific implementation, as it was provided at a functional level only. The architecture followed the reference model's layered design and segmentation on a vertical axis, thereby identifying functional elements across four (4) architectural layers:

- The **Front-end Layer**, which includes subsystems and processes related to the interaction of the end users with the platform front-end components.
- The **Back-end Layer**, which includes the core experimentation components dealing with the coordination, onboarding and lifecycle management of the experiments.

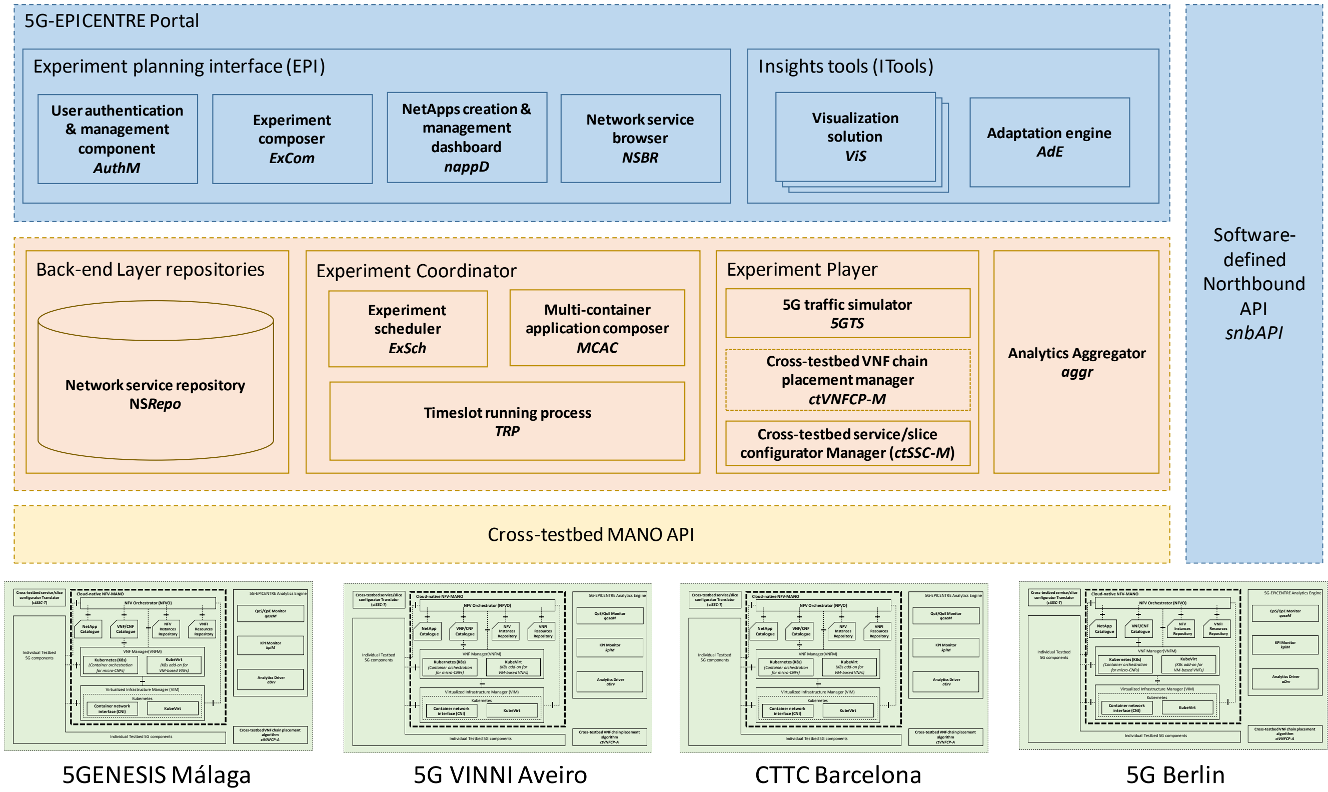


Figure 26: 5G-EPICENTRE overall functional architecture component diagram, as defined in D1.3.

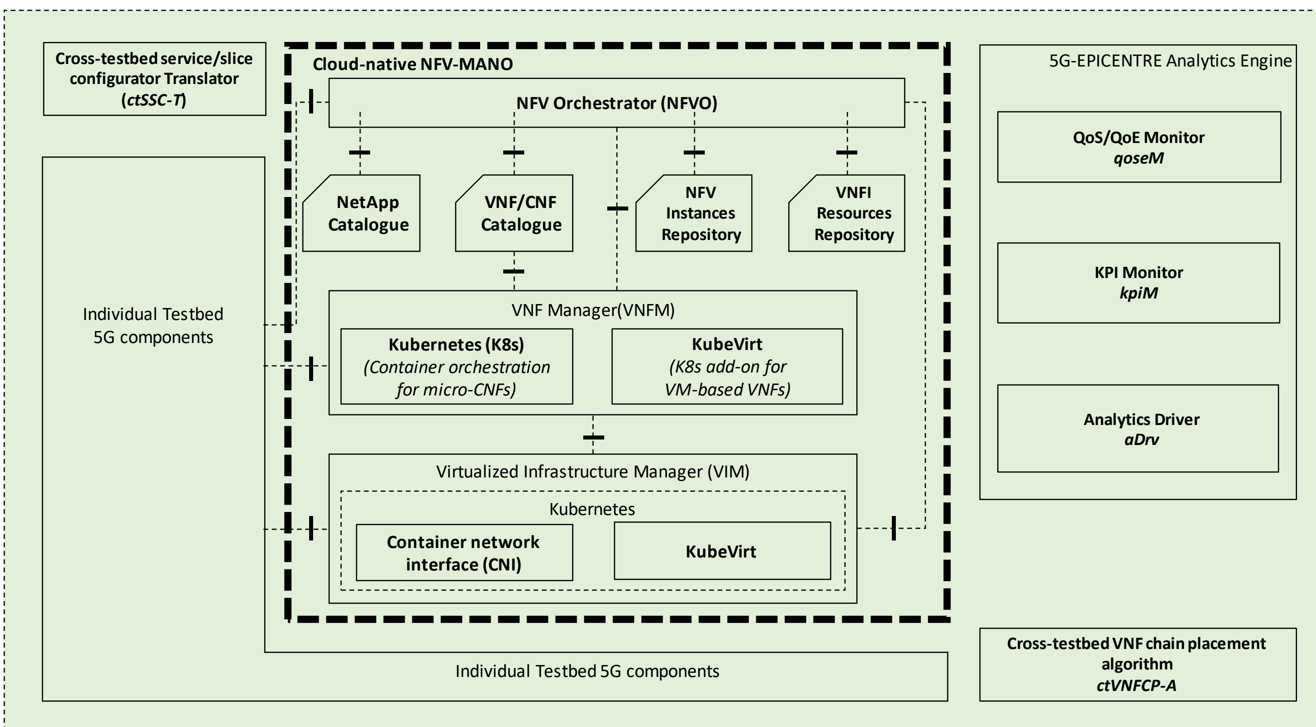


Figure 27: 5G-EPICENTRE Infrastructure Layer reference frame, as defined in D1.3.

- The **Federation Layer**, which comprises the cross-testbed combining of underlying virtualized resources in order to create new ones with greater capabilities.
- The **Infrastructure Layer**, which includes the modules that need to be replicated across the project's testbeds joined in federation to enable the 5G-EPICENTRE project to function. It also encompasses the cloud-native augmentation and implementation toward achieving the desired cross-orchestration of resources using a uniform management system (Figure 27).

More information on the different components is available in D1.3 and [1].

3.2 Architecture refinement approach

This Section outlines the approach followed in the context of Task 1.3 *'5G-EPICENTRE technical specifications and architecture'* toward refining the preliminary version of the 5G-EPICENTRE platform architecture reported in D1.3. In particular, the contents in this Section will describe the activities undertaken in the past twelve months (M13-M24 – architecture refinement), following the specification of the different interfaces in the architectural update provided in D4.1. As was the case in the previous architecture specification iteration, refinement of the architecture along with the clear definition of components, communication interfaces and deployment considerations in the context of the 5G-EPICENTRE project is undertaken as a three-step process, as shown in Figure 28. The three steps include technology exploration; decomposition via a top-down approach; and elaboration via *bottom-up* processes. Actual activities undertaken in the context of these phases will be thoroughly described in the following Sections.

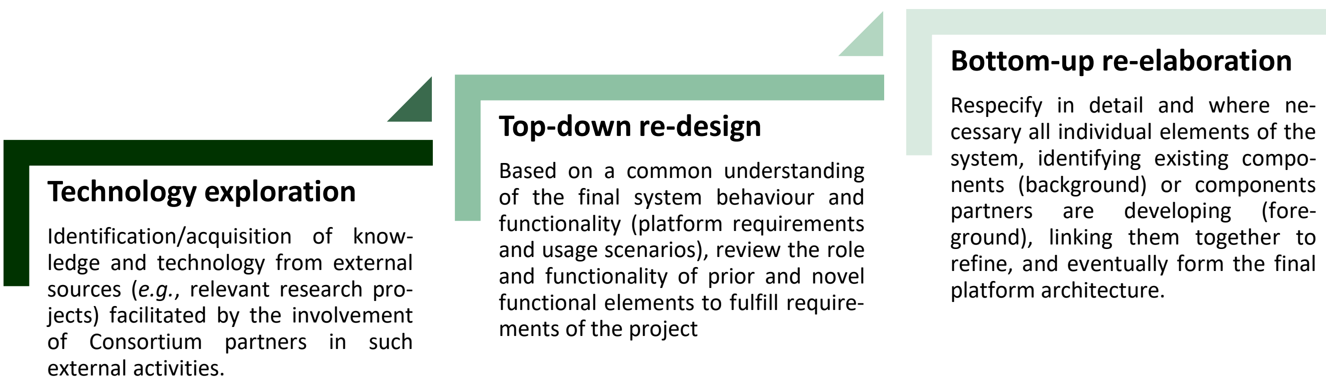


Figure 28: Architectural design approach within 5G-EPICENTRE during the architecture refinement step (M12-M24).

3.2.1 Technology exploration

Technology exploration constitutes the first step toward the architecture definition and re-specification processes, where technologies relevant to 5G-EPICENTRE are surveyed; and highly relevant results from other national and international (EU-funded) Research & Development (R&D) activities are considered as inputs into the architecture design. The role of this process in general is to: i) identify fundamental elements that existing technologies can deliver to 5G-EPICENTRE; and ii) provide evidence for compliance of aforementioned technologies with legal and ethical guidelines and concerns attached to the 5G-EPICENTRE technology and overall vision.

Technology exploration in the architecture refinement step has undertaken the further elaboration of identified technologies that are meant to be brought into the architecture (*i.e.*, partners' background, or third-party technologies with which partners are solving project-related requirements), as well as those implemented during its course (partners' foreground). For the latter point, project activities (and particularly, deliverables) throughout the technical WPs, and decisions taken therein, have been monitored to record their implications on the platform architecture. As such, the 5G-EPICENTRE system architects were able to address module characteristics with respect to information exchange and deployment across the available physical or virtualized resources.

Furthermore, technology exploration has continued the monitoring of relevant R&D activities, particularly those of other ICT-41 projects and 5G-PPP working group activities (including scientific outputs in the form of published papers in international conferences and journals), so as to remain vigilant of any group specifications with direct implications on the 5G-EPICENTRE architectural model or approach to Network Application function exposure via the contemplated platform.

3.2.2 Top-down re-design

By definition, top-down refinement of an architecture design entails breakdown of the overall system into its compositional elements thus identifying (but not detailing) any-level subsystems and base components. The goal is to specify the role in terms of services and functions that each component provides, as well as a high-level overview of the interactions between them.

The top-down re-design of the architecture was carried out in parallel to the technology exploration, so as to mark down changes to the original architecture design as soon as those were discovered and reported by project Beneficiaries. The main goal of the activity throughout M12-M24 has been to identify and include in the original design any changes to the functional elements and overall structures depicted in Figure 26 and Figure 27. This has referred to:

- the identification of previously unidentified components, modules and/or subsystems;
- merging of prior components into novel functional blocks, to be seen and treated by other components in the architecture as "black boxes" (simplification of complex structures);
- refinement of previous components and subsystems, as it pertains to their roles, functions and capabilities' exposure;
- identification of third-party tools (from technology exploration) as surrogate technologies to fulfil requirements and foreseen component functions (thus limiting the amount of work Consortium members spent on building platform elements, and increasing their capacity to focus on experimentation, as indicated by the most recent project monitoring and evaluation feedback – RV2).
- break-down of previous components and subsystems into more elaborate structures, so as to highlight independent functions and particular contributions introduced to proprietary, or third-party components;
- removal of obsolete functional elements, including also the absorption of prior components' functionalities by newly introduced functional blocks with extended capabilities.

Throughout this exercise, effort was spent to maintain the alignment to the original architecture layered design by placing novel components and subsystems in the architectural layer where they were deemed to fit more appropriately, based on their communications' needs and functional descriptions. Afterwards, communication flows into and out of each novel functional block were drafted to better illustrate the block's workings inside the overall architecture diagram. Consortium partners were then asked to elaborate on the roles and functions of components and subsystems under their care, to ensure concrete descriptions were attached to each architectural block; and hence no ambiguity, or misconceptions were raised among the other partners.

3.2.3 Bottom-up re-elaboration

In software development, bottom-up design entails the collection and interconnection of modules and components, which serve as building blocks toward designing larger, more complex systems. Within the context of the 5G-EPICENTRE architectural specification, the bottom-up approach followed the activities of the top-down architecture decomposition, involving the identification of technologies and software components that the individual technical partners envisioned to contribute to the project.

The architecture refinement step was concluded with the bottom-up re-elaboration of the functional elements, corresponding to the maturing state of development and integration activities taking place throughout the duration of the project. The purpose was to gather concrete info on all functional elements, and validate that they fulfil the technical requirements of the project (Section 2.4).

Bottom-up activities were hence grouped carried out in two distinct stages:

- **Stage 1** was aimed at elaborating and giving substance to identified components using a final, elaborate component refinement template (see Annex III). The goal was to gain insight into the architectural components required for the technical realisation of the information flows specified in D4.1 (M12-M17) and D4.4 (M18-M24). Concretely, during this stage, critical information was gathered, especially about the novel functional blocks listed in Section 3.5.
- **Stage 2** was carried out in the form of two architectural workshops, aimed at both presenting the refinements based on the outputs of technology exploration and top-down specification, as well as gathering the final needed information, insight and requirements of the final compiled list of components.

More information about the conclusive activities of Stage 2 are presented in the following subsection (3.2.3.1). The final specification and description for each component and its interfaces is presented in Section 3.3.

3.2.3.1 Architecture workshops

The 5G-EPICENTRE partners gathered together twice in online architecture design workshops organized under the responsibility of Task T1.3 Leader FORTH (hence concluding the activity with a total of five architectural workshops held over the course of T1.3 lifetime). The goals and conclusions of each workshop during architecture refinement are indicated in the summary list in Table 4). Both technical and UC-leading partners were invited to partake in workshop activities in order to acquire a complete operational picture of the 5G-EPICENTRE architecture and its different views to be defined.

Table 4: Architecture workshop meetings held throughout the lifetime of Task T1.3 since D1.3 (M6).

Date	Date (in Project Months)	Workshop goals	Conclusions and lessons learned
02 Dec 2022	M24	Align the 5G-EPICENTRE architecture to the Network Application ecosystem model, realizing the potential of defining Network Application chains within components of the architecture itself.	Drafted a revision of the original schematics to accommodate both the actual component implementation; and the content from the Network Application white paper by the 5G-PPP SB WG [2].
12 Jan 2023	M25	Presentation of the pre-final platform architecture stack, and discussion among partners on final details.	Elaborate on changes brought up by technical partners, the foreseen information flows and the deployment considerations (basically, where the components in the front-end and back-end layers will be hosted).

3.3 General concepts

Prior to describing the architecture definition processes in detail, it will be important to establish several key concepts that will be used throughout this Section.

3.3.1 Network Application

The contents of this Section aim at summarising key insights and developments in the 2022 5G-PPP Software Network Working Group White paper on *"Network Applications: Opening up 5G and beyond networks"* [2], and discuss their implications on the 5G-EPICENTRE architectural refinement. It discusses the co-development of the Network Application concept with other ICT-41 projects in the same programme, aiming at defining the common aspects of the Network Application delivery model (in general), as well as the benefits it could unlock for different vertical sectors (in the 5G-EPICENTRE case, the PPDR vertical).

Network Applications are defined as an interaction layer between vertical applications and the network control plane. To facilitate such an interaction, Network Applications are defined as pieces of software that can consume northbound Application Programming Interfaces (APIs), most notably of the 5G Core (5GC), thus providing an extent of programmability over the 5G network itself.

3.3.1.1 Network Applications in 5G-EPICENTRE

Specifically, for 5G-EPICENTRE, Network Applications can act on the control plane functions of the 5GC as (or via) Application Functions [3], hence providing services to the network subscriber. If an Application Function (AF) is trusted, it can interact directly with 5GC network functions. If it is developed by a 3rd party (to be considered untrusted), exposure of network capabilities is handled by consuming the services exposed by the 5GC Network Exposure Function (NEF). In this way, Network Applications can exploit a number of network features, *e.g.*:

- influence traffic policies and routing decisions (*e.g.*, to support edge computing);
- trigger specific UEs' actions;
- request the execution of services provided by the 5GC Location Management Function;
- gather from, and share with the network any kind of data and analytics via the 5GC Network Data Analytics Function;
- exploit the N5 interface exposed by the 5GC Policy Control Function, to exercise QoS management over the network.

The Network Applications model described above fits well with the responsibility of the 5G-EPICENTRE project related to demonstrating the transformative benefits of 5G technology for PPDR agencies and end-users. Embedding support for the Network Application delivery model into the 5G-EPICENTRE platform architecture affords vertical application developers the capacity to experiment and demonstrate enhanced capabilities of the 5G network, such as:

- Guaranteed privileged QoS to PPDR vertical services.
- Assured prioritization of specific emergency flows of data via an integration of 5G QoS Identifier (5QI) management capabilities.
- Ensuring that every data stream in PPDR mobile services can be transmitted over a 5G network, within a very short time frame and with guaranteed quality.

3.3.1.2 Implications on the 5G-EPICENTRE platform architecture

According to [2], a Network Application can either be integrated (chained) within the vertical application, or it can act as a middleware between the vertical application and the 5G system. Referring to the deployment of Network Applications across the different ICT-41 projects, three options of interaction are proposed between vertical application developers and 5G network system (*e.g.*, testbed) owners (Figure 29):

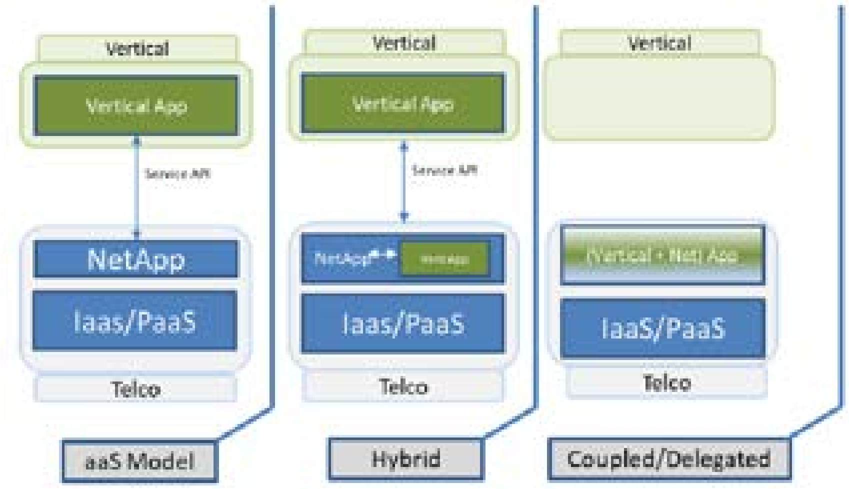


Figure 29: Options of interaction between vertical applications and Network Applications (Image retrieved from [2], licenced under CC BY 4.0).

- **As-a-Service:** the vertical application consumes API(s) exposed by the Network Application(s). The vertical application is considered to be fully deployed in the vertical service provider domain.
- **Hybrid:** A part of the vertical application is delegated to (and subsequently managed by) the testbed owner (for instance, parts of the application needing to be deployed on the edge). Part of the vertical application resides in the vertical service provider domain, and interacts with the Network Application(s) in the testbed by means of their exposed APIs/
- **Coupled/Delegated:** the vertical application developer delegates the entire application to the testbed owner. The Network Application is hence embedded (*e.g.*, chained) into the vertical application itself, and is managed by the operator.

Key implications of the above classification for the 5G-EPICENTRE architecture are as follows:

1. Network Applications are always part of the testbed architecture.
2. Already developed (or in development) software in 5G-EPICENTRE may fit the description of the Network Application approach, warranting transferral to the Network Application model alongside the use case vertical application components.

3.3.2 Nomenclature

Table 5 specifically highlights these general concepts. Key common definitions used in the context of the project and established in D1.6 (M18), is reiterated in this space for the sake of clarity and completeness.

Table 5: Architecture general concepts.

Concept	Definition
Architecture	An architecture comprises the organizational structure of an IT system or component, along with guidelines, principles and relationships that dictate its design, and its development over time [4] [5]). From a system standpoint, an architecture should specify its rudimentary structure from the perspective of its elements; goals; relationships; interfaces; functions; limitations; behaviours; principles/rules; characteristics; and physical and logical properties.

Architecture reference model	A reference model in general refers to a solid model used as a frame of reference to describe the design of specific models in a given use case. It comprises a conceptual framework that contains the minimum number of necessary unifying concepts and connections that can summarize all interactions between entities inside a given environment. An architecture reference model is used to describe the structure of a system on a high level and provide guidance toward developing specific and concrete architectures through the relationships in its model.
Container	Containers are executable software offering computer resources virtualization, provisioning those resources into instance that can run software services (microservices) and applications. Container images contain an entire runtime environment (*i.e.*, the full application code along with all its dependencies, libraries and binaries), enabling an application to be deployed efficiently in multiple computing environments. Container images become containers at runtime.
Facility	In the context of 5G-EPICENTRE, the term “facility” is related to the testbeds of the project, meaning that it refers to the 5G & Kubernetes implementations done in the scope of the project (see *Testbed*).
Platform	In the context of 5G-EPICENTRE, the term "platform" is related to the full 5G-EPICENTRE solution, from the user interface, which will allow the user to define the experiments to be performed, through the Experiment Coordinator and the Cross-testbed MANO, to the analytics engine that will prepare the results for visualisation.
Testbed	In the context of 5G-EPICENTRE, the term "testbed" is related to facilities located in Aveiro, Berlin, Málaga and Barcelona, where a 5G Standalone deployment has been made, on which the different Use Cases are going to deploy their services.

3.4 5G-EPICENTRE platform architecture re-specification

5G-EPICENTRE envisions a federation of four (4) independent *testbeds/facilities* (definition, Section 3.3), each characterized by different 5G standalone implementation and technologies, yet all capable to support deployment of (containerized) network functions and applications, that are managed by a Kubernetes (K8s) cluster architecture. On top of this federation, a centralized experimentation *platform* (definition, Section 3.3) is proposed, which enables access to each of the underlying testbeds’ resources, either deploying the experimenter’s K8s applications to any of the facilities independently, or across multiple Kubernetes clusters distributed across the facilities. Therefore, and as a result of deliberation among consortium partners under different roles (*e.g.,* testbed owners, technology developers and experimenters) during the architecture workshops, it was decided for some elements of the 5G-EPICENTRE architecture to be replicated across all four testbeds toward ensuring synchronisation and harmonisation of the different platforms partaking in the federation, most prominently the cloud native NFV components. Key to this federation is to ensure interoperability with the other platforms, particularly in the Management and Orchestration (MANO) capabilities for deploying and running experiments’ workflows in K8s clusters. Hence, each platform accommodates a K8s-based infrastructure, to orchestrate and manage deployment and operation of vertical containerized applications on top of their 5G infrastructures.

The final platform architecture diagram, resulting from the processes described in Section 3.2, is shown in Figure 30. As previously mentioned, this final version retains the architectural layers specified both in the conceptual architecture reference model described in the GA, as well as the preliminary version reported in D1.3, namely the ‘**Front-end layer**’, ‘**Back-end layer**’, ‘**Federation layer**’ and ‘**Infrastructure layer**’. The first three layers constitute the entities comprising the *5G-EPICENTE platform*. The 'Infrastructure layer', on the other hand, corresponds to the framework describing the augmentations necessary for each individual testbed in the federation, so as to support the 5G-EPICENTRE technical requirements.

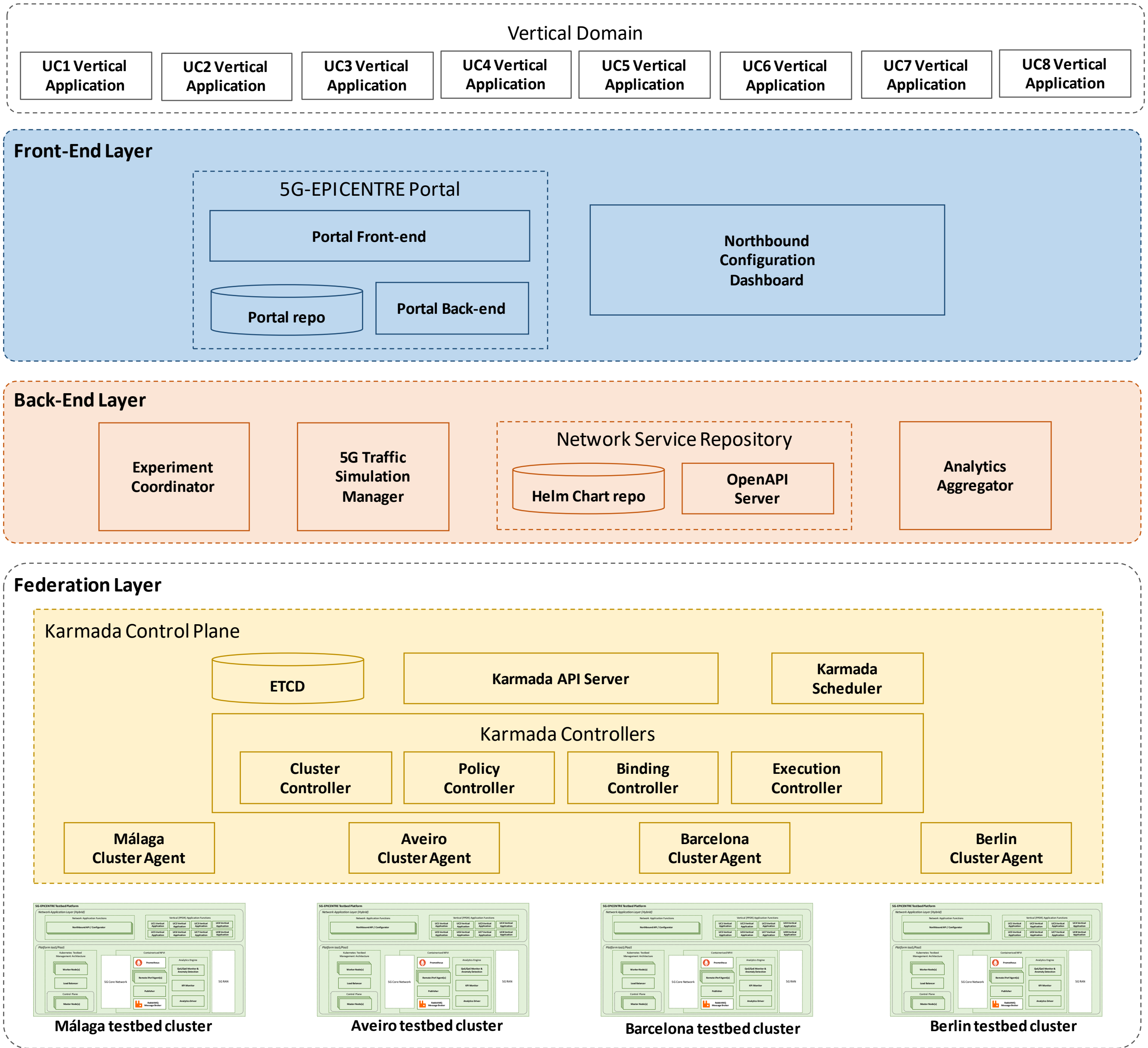


Figure 30: 5G-EPICENTRE overall functional architecture component diagram.

3.4.1 5G-EPICENTRE platform

On the ***'Front-end'*** Layer, 5G-EPICENTRE is now composed of functional components related to facilitating the interaction between the platform and targeted end-users, *i.e.*, vertical[3] application developers. It aims to effectively address how these actors can utilize the front-end platform components to define, execute, and configure 5G experiments' workflows. Hence, this layer houses the **5G-EPICENTRE Portal**, a web-based user interface where actors can define their experiment as a workflow consisting of steps; and delegate (parts of) their vertical application(s) to be composed and managed by the testbed owner. It is further used to depict the (part[s] of) the

[3] 'Vertical', for the purposes of the deliverable, refers to the public protection and disaster relief (PPDR) 5G vertical sector.

vertical application deployed in the vertical service provider domain (the ones that consume the 5G-EPICENTRE Network Applications as-a-Service). To unlock such features, a **Northbound Configuration Dashboard** is provided to experimenters, for managing interconnection between the vertical applications' components, with the services and network applications provided for configuring the network infrastructure within predefined sets of values. Functional elements under this layer are described in Section 3.5.1.

The ***'Back-end'*** Layer incorporates key functional components of the centralized platform, essentially: (i) translating and delegating workflows originating at the front-end layer to the underlying testbed(s) assigning K8s clusters to such workflows; as well as (ii) aggregating metrics and Key Performance Indicators (KPIs)' information from the different testbeds, for visualization at the Portal user-facing components. This layer hence defines the synergies that front-end components should implement with the testbeds *e.g.*, all necessary data the system has to provide to the different facilities, as well as the means by which this data is accessed from the outside. Functional elements under this layer are described in Section 3.5.2.

The ***'Federation'*** Layer finally handles cross-testbed cluster management services, with each testbed being treated as a point-of-presence. The layer embeds **Karmada** as its multi-cluster management solution, allowing clusters and cluster lifecycles to be managed in one place using unified, standard APIs [6]. Functional elements under this layer are described in Section 3.5.3.

3.4.2 Testbed reference framework

As can be seen in Figure 30, the ***'Infrastructure'*** Layer (a sub-Layer to the "Federation" Layer, as shown in Figure 30) is comprised of the four augmented testbeds, each representing its own individual 5G standalone system configuration, with integrated support for a K8s management environment (thus favouring a platform-as-a-service [PaaS] model [7]). Each testbed is comprised of its own NFV ecosystem, including the specific implementations of the 5G architecture (5GC, 5G Radio Access Network [RAN]), alongside an augmented Network Functions Virtualization Infrastructure (NFVI), that integrates all virtualized components needed for a testbed virtualized infrastructure to deliver on the 5G-EPICENTRE technical requirements. Each platform should integrate a Kubernetes-based architecture for the coordination and management of both the virtual resources in the NFVI domain; and Virtual Network Functions (VNFs)' lifecycle, in accordance with [8] (see also, Annex II). Each testbed becomes hence able to deploy a K8s cluster able to deploy Cloud-native VNFs (CNFs). Using the Karmada multi-cluster management system (see previous Section), the project's vision on cross-testbed federation (as illustrated in Figure 31 and presented in D4.4) can be realized.

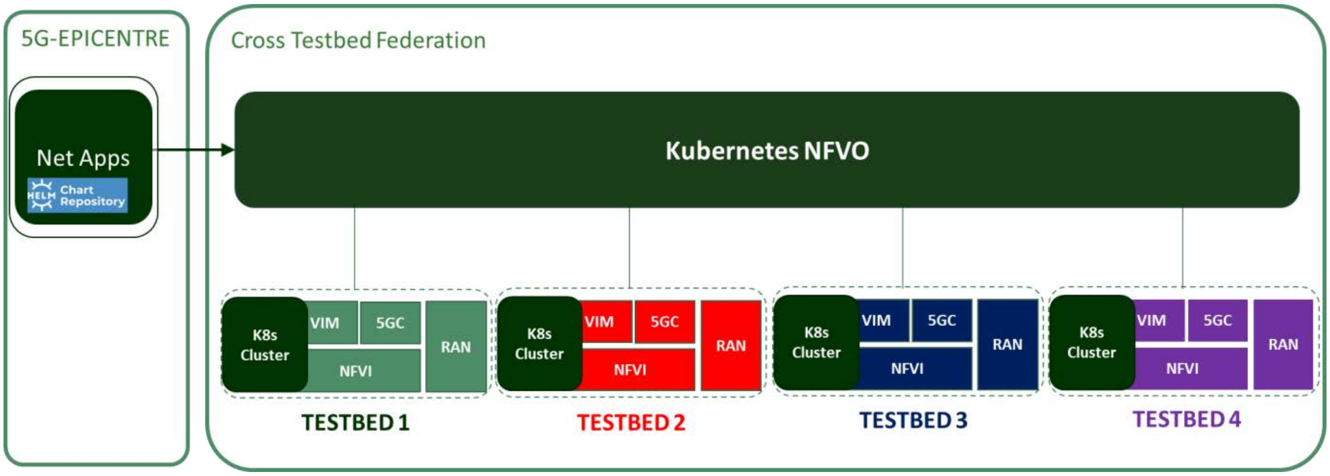


Figure 31: Cross-testbed federation concept for 5G EPICENTRE, retrieved from D4.4.

From the Network Applications' ecosystem perspective (recall from Section 3.3.1), the abovementioned developments correspond to the PaaS layer as shown in Figure 29. In line with the definition of Network Applications as essentially a middleware layer on top of the PaaS, the testbed reference framework is further elaborated to depict the various Network Applications contemplated in the context of the project (representing different levels of trust and openness to third parties for the purposes of experimentation).

Figure 32 zooms in on this common frame of reference for the augmentation of the individual testbeds into the 5G-EPICENTRE federation. Functional elements in this layer are described in Section 3.6.

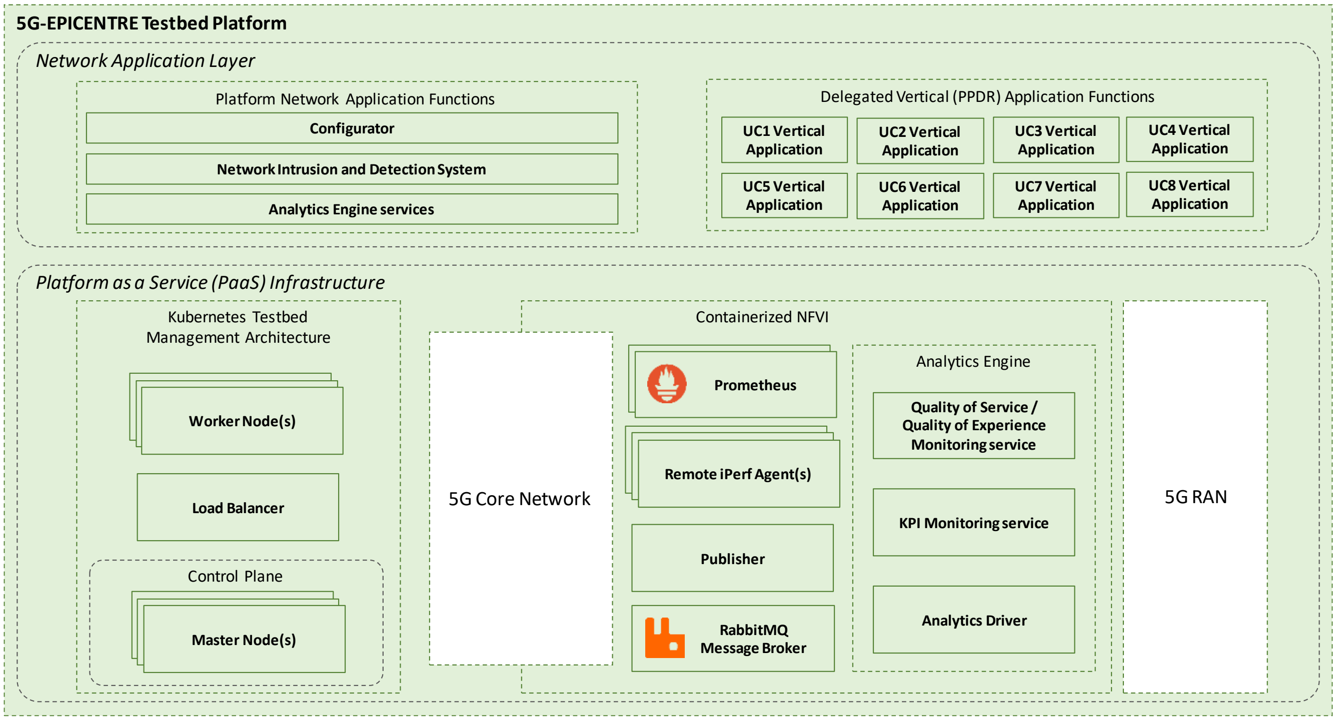


Figure 32: 5G-EPICENTRE Infrastructure Layer reference frame.

3.5 5G-EPICENTRE Platform functional view

In this Section, the functional entities of the 5G-EPICENTRE architectural stack (as depicted in Figure 30), their responsibilities, and interactions with other components will be elaborated. This Section aims to represent the functional blocks specified in accordance with the design methodology elaborated in Section 3.2 provided at a *functional* level, implying no specific implementation (unless otherwise specified). Hence, the following Sections will focus on detailing the main components of the 5G-EPICENTRE platform, presenting their main functionalities and well-represented responsibilities, thus providing a comprehensive overview of the overall architecture functional view. All component descriptions present in this Section are derived from the templates used for bottom-up design re-specification, as presented in Section 3.2.3 and in Annex III.

3.5.1 Front-end Layer functional entities

3.5.1.1 5G-EPICENTRE Portal

The 5G-EPICENTRE Portal (henceforth referred to simply as "the Portal") is a user-facing, web-based application, responsible for the following operations:

- It enables users authenticated on the platform as experimenters, to book/reserve one or more testbeds for a newly defined, or reproduced experiment (*i.e.*, workflow composed of steps under a specific environment) on one or more testbeds' K8s clusters.

- It enables users authenticated on the platform as experimenters or function developers, to delegate (*i.e.*, by uploading and sharing, either completely, or partially) their vertical application components / functions, which should be instantiated and managed by the 5G testbed owner (under either the 'Delegated' or 'Hybrid' option for interaction between the vertical and operator, as depicted in Figure 29).
- It enables users authenticated on the platform as testbed owners to obtain vertical application components intended for deployment on the operator's domain, and confirm the scheduling of an experiment by "programming" the platform to anticipate the experiment execution date and automatically start the experiment on the designated date and time.
- It enables users authenticated on the platform as experimenters, to keep track of scheduled experiments, and execute the experiment on demand.
- It enables users authenticated on the platform as experimenters, to plot and visualize metrics arriving from the testbed(s) in a comprehensive dashboard to be able to interpret and exploit the value of experiment KPIs.

The internal architecture of the Portal is presented at an abstract level, and comprises both frontend and backend components, following the implementation guidelines and functionality described in D3.3 (concerning experiment planning user interfaces) and D4.4 (visualization of metrics). An elaborate description of the various elements comprising this functional element as a whole is planned for the conclusive output of WP3 (deliverable D3.2, due in M30).

The Portal consumes an API exposed to it by the **Experiment Coordinator** (see Section 3.5.2.1), and does not expose any APIs to other functional entities in the architecture.

3.5.1.2 Northbound Configuration Dashboard

The Northbound Configuration Dashboard (henceforth referred to simply as "the Dashboard") is a user-facing, Single Page Application (SPA), which delivers a UI experience for experimenters to configure network properties within predefined sets of values. It is an optional component in the architecture, provided for experimenters to access the services exposed through the Configurator Network Application (see Section 3.6.3.1), serving as an enabler for configuration of the network (*e.g.*, policies on traffic prioritization).

It corresponds to functionality previously associated with the *Software-defined Northbound API* block (D1.3 – for an intermediate "snapshot" of the state of the *Software-defined Northbound API* block, which first identified the Dashboard as a component for the functionalities envisioned by the prior component, consult D3.1). A final, elaborate description of the various elements comprising this functional element as a whole is planned for the conclusive output of WP3 (deliverable D3.2, due in M30).

The Dashboard is accessible through a URL that the testbed owner associates to the experiment entry at the Portal when the experiment components are onboarded. It does further not expose any APIs to other functional entities in the architecture.

3.5.2 Back-end Layer functional entities

3.5.2.1 Experiment Coordinator

The Experiment Coordinator (henceforth referred to simply as "the Coordinator") encompasses 5G-EPICENTRE platform functionality corresponding to automated experiment scheduling; and production of the necessary configuration files during experiment deployment and execution. A central block to the 5G-EPICENTRE experimentation workflow, the Coordinator undertakes the following responsibilities:

- It is responsible for receiving experiment execution requests from the Portal (which contain a description of the experiment's requirements and general configuration, *e.g.*, the Helm chart(s) to unpack), and deliver commands for deploying the experiment vertical application components through the Karmada K8s management stack (see Section 3.5.3).

- It automatically generates the configuration of the different platform/testbed components that are involved in the experiment execution, *e.g.*,
    - it prepares experiment metadata to be added to an experiment list maintained by the testbed's Publisher component;
    - it triggers the traffic simulation processes at the 5G Traffic Simulation Manager, effectively imparting the information with respect to the traffic characteristics to be simulated (and info on the pre-deployed iPerf Agents to be configured and started for its generation, see Section 3.6.2.5), as well as the topic exchanges where results must be published.
- It oversees the experiment execution based on the experiment representation as a workflow of steps, and initiates the different processes, or experiment stages whenever required.

The Coordinator exposes an API towards the **Portal** (see Section 3.5.1.1), which supports the following capabilities/operations (Table 6):

Table 6: Experiment Coordinator API endpoints exposed to the 5G-EPICENTRE Portal

Method	Endpoint	Description
POST	/experiment	Endpoint for creating and queueing a new experiment execution, based on the contents of the Experiment Descriptor (JSON).
GET	/execution/{id}/logs	Endpoint for retrieving all log messages generated by an experiment execution (in JSON format), separated by experiment stage.
GET	/execution/{id}/results	Endpoint for retrieving logs and files (in a compressed file) generated by the experiment execution.

The Coordinator consumes the *Karmada API*, exposed to it by the **Karmada API Server** (see Section 3.5.3.1), based on the prescribed functionalities. It also consumes the APIs exposed to it by the **5G Traffic Simulation Manager** (see Section 3.5.2.2) and testbed **Publisher** component (see Section 3.6.2.4), to support the methods described in their respective Sections. It further integrates a Helm client for the communication with the JFrog Helm repo component of the **Network Service Repository** (see Section 3.5.2.3), toward downloading the Helm chart and applying its contents to the **Karmada API Server**.

3.5.2.2 5G Traffic Simulation Manager

The 5G Traffic Simulation Manager (henceforth referred to simply as "the Traffic Simulator") is responsible for handling the creation of artificial network traffic, in order to simulate different 5G network conditions. This traffic is generated by traffic agents (based on iPerf tool[4], see Section 3.6.2.5) that are deployed at each individual testbed. Traffic parameters are specified by the experimenter user for an experiment at its definition stage (occurs in the Portal) and are obtained by the Coordinator when the experiment is scheduled, containing the identifiers of the agents involved, as well as the necessary parameters to generate traffic between their internal components. The Traffic Simulator then effectively sends the required parameters to the designated traffic agents in order to simulate the requested network conditions during the experiment. The concept is illustrated in Figure 33.

[4] https://iperf.fr/

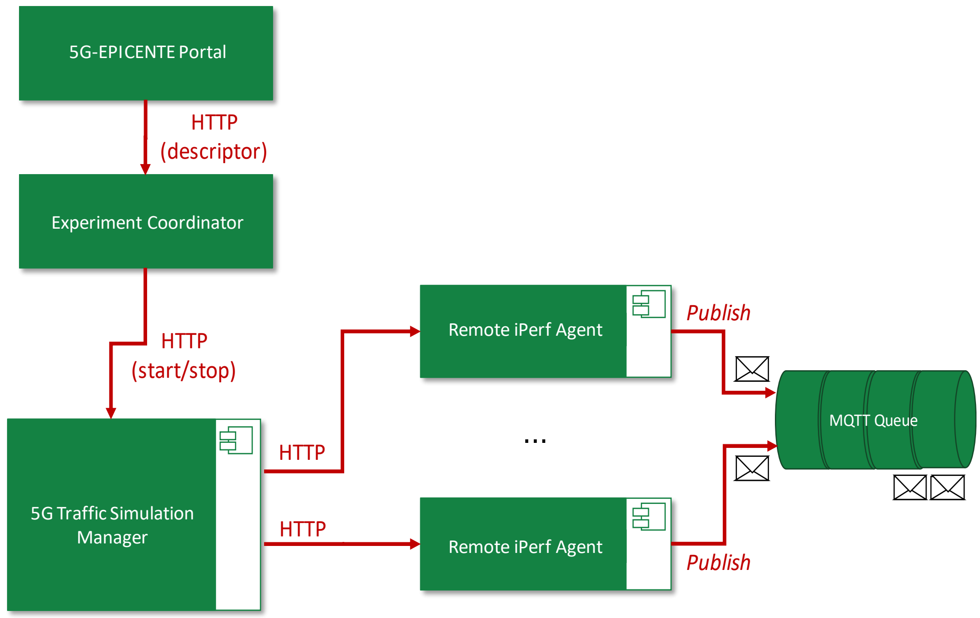


Figure 33: 5G traffic simulation engine functional elements and interface concept

The Traffic Simulator exposes an API towards the **Coordinator** (see Section 3.5.2.1), which supports the following capabilities/operations (Table 7):

Table 7: 5G Traffic Simulation Manager API endpoints exposed to the Experiment Coordinator

Method	Endpoint	Description
POST	/start	Endpoint for executing an iPerf command (described using iPerf parameters) on the designated Remote iPerf Agent in the request JSON payload, effectively telling that agent to start a server/client pair for generating the specified traffic between them.
POST	/stop	Endpoint for terminating an iPerf server running on an agent, identified by a dictionary with keyword "agent_id" and value "ID".
GET	/retrieve?agent_id={id}	Endpoint for returning the final result produced by the specified agent hence reporting traffic status information for experiment lifecycle management. If the designated agent does not exist, the endpoint returns a JSON object with the corresponding error.
POST	/add_iperf_agent	Endpoint for adding an iPerf agent to the Traffic Simulator.
POST	/remove_iperf_agent	Endpoint for removing an iPerf agent from the Traffic Simulator.

The Traffic Simulator consumes an API exposed to it by the **Remote iPerf Agent** (see Section 3.6.2.5).

3.5.2.3 Network Service Repository

The Network Service Repository (henceforth referred to simply as “the Repository”) provides the 5G-EPICENTRE platform architecture with centralized support for storage and management of Network Service (NS) and application artefacts. These will be packaged and stored in the form of Helm[5] Charts. These Helm Charts aim at defining templates for properly installing the vertical application on a K8s cluster in an automated manner. They will be hence utilised in the southbound exchange between the Repository and testbed K8s orchestrator (managed over the **Karmada control plane block**, see Section 3.5.3) for setting up and instantiating clusters executing CNFs over the testbeds.

The internal architecture of the Repository is shown in Figure 34. A JFrog[6] private Helm repository is used to store the chart packages, providing methods for uploading, retrieving, updating and deleting Chart packages. The JFrog Helm Repo exposes an extensive REST API[7] that supports a fully automated provisioning of Helm charts to a Kubernetes cluster, that can be used for communication with the **Coordinator** (see Section 3.5.2.1) to obtain chart packages and apply them over the **Karmada API Server**. An OpenAPI[8] server is used as a proxy, for facilitating the interaction of the platform front-end components (Portal) with the private Helm repository for supporting these basic functions.

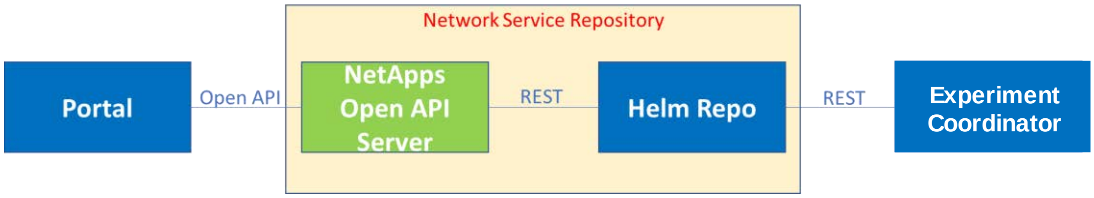


Figure 34: Network Service Repository functional elements and interfaces

As a result, the Repository exposes an API, which supports the following capabilities/operations (Table 8):

Table 8: Network Service Repository API endpoints exposed to the 5G-EPICENTRE Portal

Method	Endpoint	Description
GET	/	Returns the list of filenames in the repository.
GET	/{filename}	Returns the specified file information and content. If the file is in an archive file format (implying Helm chart), the contents of "Chart.yaml" are returned.
DELETE	/{filename}	Deletes a file by filename.
PUT	/{filename}	Uploads a file to the repository.

The Repository does not consume any APIs exposed by other functional entities in the architecture.

3.5.2.4 Analytics Aggregator

The Analytics Aggregator (henceforth referred to simply as "the Aggregator") summarises the analytics and performance data that are generated from the different testbeds in the Infrastructure Layer (*i.e.*, KPIs, and potential anomalies discovered), and forwards them in a unified output format to the Portal for visualization.

As with all analytics components in the 5G-EPICENTRE architecture, the Aggregator communicates with northbound (*i.e.*, the Portal), and southbound (*i.e.*, components of the Analytics Engine) functional elements by means

[5] https://helm.sh/
[6] https://jfrog.com/
[7] https://www.jfrog.com/confluence/display/JFROG/Kubernetes+Helm+Chart+Repositories
[8] https://swagger.io/specification/

of asynchronous message queues. Hence, the Aggregator does not consume APIs exposed by other functional entities in the architecture, nor does it expose any APIs of its own to other modules. Through these asynchronous interfaces, the Aggregator supports the following capabilities/operations:

- **Southbound communication:** The Aggregator will subscribe to the message queue published to by the internal Analytics Engine modules deployed at each individual testbed (see Sections 3.6.2.2 and 3.6.2.3), aggregating their information to produce reports on network performance.
- **Northbound communication:** The Aggregator will construct the aggregated measurements message, notifying of values corresponding to anomalous conditions or trends, routing it to the appropriate queue to be consumed by the subscribing Portal service. The Aggregator will not wait for a response.

3.5.3 Federation Layer functional entities

Distributed K8s cluster federation is achieved in 5G-EPICENTRE using the Karmada[9] Kubernetes management system. Based on an elaborate mapping of the preliminary 5G-EPICENTRE platform architecture functional view (D1.3) and the Karmada features and components, the foreseen requirements assigned to the prior abstract 'cross-testbed MANO API' concept for federation of the testbeds, are fulfilled by the architectural blocks of the Karmada control plane (Figure 35). The following Sections aim at demystifying the roles and responsibilities of the aforementioned components, with respect to the functionalities they support for 5G-EPICENTRE to achieve cross-testbed resource management and orchestration capabilities in accordance to the project's GA objectives.

To enable the access to the federated resources, the Karmada control plane (via the **Karmada API Server**, see Section 3.5.3.1) exposes several HTTP API endpoints towards the **Coordinator** (see Section 3.5.2.1), supporting capabilities and operations for initiating the different experiment stages, upon request by platform users (authenticated as experimenters) during experiment executions. At the deployment stage, the API is also exposed to request a slice configuration and its eventual deployment over a specific testbed, hence affording the Karmada control plane to carry out service and slice management. Both operations are carried out in two stages, one reaching the Karmada controllers and triggering the necessary exchanges between them; and the other related to forwarding the configuration of the specific K8s cluster to the underlying K8s cluster block in each individual testbed. As a result, the functionality previously assigned to the *cross-testbed Service and Slice configurator Manager* and *Translator* components, as described in D1.3 (previously distributed over the Back-end and Infrastructure Layers, see also Section 3.1) are to be considered deprecated, and part of the Karmada control plane functionality.

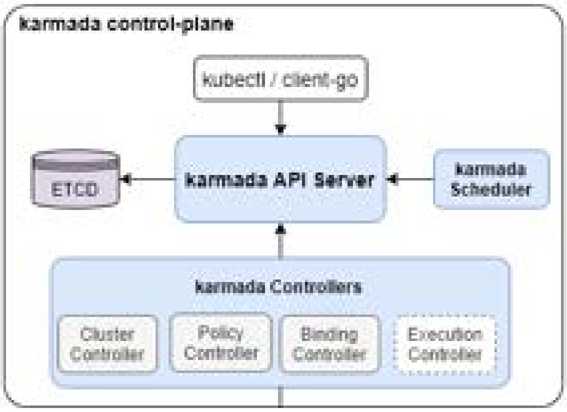


Figure 35: Karmada control plane architecture (adapted from [6])

[9] https://karmada.io/

3.5.3.1 Karmada API server

The Karmada API Server is an extended K8s API Server, attached to a separate key-value store (**etcd**, see Section 3.5.3.3). Because the Karmada control plane architecture mirrors that of a single K8s cluster [9], the Karmada API server extends the K8s native API and Policy APIs to support federation of K8s clusters running across the different testbeds, essentially treating them as a single K8s cluster. Therefore, the Karmada API server is core component in the Karmada control plane architecture, responsible for communication of all other Karmada control plane entities, as well as communication of external components (to the Karmada control plane) to communicate with the multi-cluster environment, as if it were a single K8s cluster. Therefore, it can be considered the access point (frontend) to the entire federation.

As previously mentioned, the Karmada API Server is responsible within the Karmada control plane block, to expose an API towards the **Coordinator**, supporting capabilities and operations described in the *Karmada OpenAPI specification*[10].

3.5.3.2 Karmada scheduler

The Karmada Scheduler (henceforth referred to simply as "the Scheduler") provides advanced management and scheduling of workloads across the federated testbed clusters. This involves advanced multi-cluster scheduling policies (including potential application splitting across clusters), to best match the resource request against resource object availability in the K8s cluster block of 5G-EPICENTRE testbed platforms. Like all components in the Karmada control plane, the Scheduler communicates with all other internal and external (to the Karmada control plane) blocks by means of the Karmada API Server.

The aforementioned communication and extended scheduling capabilities and policies offered by the Scheduler, can serve as a means for further determining optimal placement of VNFs. The approach is divided in two stages. First, the Scheduler arrives at a decision on binding the deployment of the Helm Chart package contents to a member K8s cluster, at which point data can be exchanged between the Karmada control plane Policy controller (see Section 3.5.3.4) and the cluster-specific API server/Agent (see Section 3.5.3.5) to coordinate with that cluster's K8s API Server to carry out deployment in its cluster. After the specific cluster is decided, placement is determined on the most suitable node of that cluster, by means of that cluster's K8s API server and K8s Scheduler interactions. As a result, the functionality previously assigned to the *cross-testbed VNF Chain Placement Manager* and *Algorithm* components, as described in D1.3 (previously distributed over the Back-end and Infrastructure Layers, see also Section 3.1) are to be considered deprecated, and part of the Karmada, and individual testbed K8s control plane functionalities (extended via project developments in the context of Tasks T2.3 and T4.3).

The Scheduler, as already established, does not consume, neither exposes any APIs from/to other functional entities in the 5G-EPICENTRE architecture.

3.5.3.3 etcd

Etcd within the Karmada control plane architecture is a separate key-value store for the storage of all data related to the operation of the multi-cluster environment (*e.g.*, state of the system, desired state of the system, system configuration files). Essentially, it stores the K8s/Karmada API objects that are packaged in the Helm Charts downloaded from the **Network Service Repository** (see Section 3.5.2.3)

Since the Karmada API Server is solely responsible for connecting to the etcd, the latter does not consume, neither exposes any APIs from/to other functional entities in the 5G-EPICENTRE architecture.

3.5.3.4 Karmada Controller Manager

The Karmada Controller Manager consists of several controllers charged with different responsibilities within the Karmada control plane:

[10] https://github.com/karmada-io/karmada/blob/master/api/openapi-spec/swagger.json

- The **Cluster controller** manages the lifecycle of the clusters registered in the federation and maintains the status of each cluster as reported by its respective cluster agent (see Section 3.5.3.5). It will undertake the necessary operations for the cluster to reach the desired state from the current state.
- The **Policy controller** defines how the clusters of the federation are to be used, and how the vertical applications are to be distributed among the different clusters. Its combined usage with the Scheduler (see Section 3.5.3.2) facilitates the foreseen functionality for optimal VNF placement to the most appropriate, or across different clusters in the federation.
- The **Binding controller** is responsible for binding a resource template and a list of targeted clusters, enabling the Karmada control plane to treat the various clusters across the testbeds as a single K8s cluster.
- The **Execution controller** finally manages the workloads of the clusters.

Each controller exposes its own API that is exclusively consumed by the Karmada API Server. Therefore, the elements of the Karmada Controller Manager do not consume, neither expose any APIs from/to other functional entities in the 5G-EPICENTRE architecture that are described in this document.

3.5.3.5 Cluster Agent

Each testbed cluster is represented (*i.e.*, registers, and exposes access to the cluster) by the Karmada Cluster Agent (one per cluster). It is responsible for creating the K8s objects that should be associated with the cluster, and maintaining (and reporting) the cluster's status as part of the interaction with the Cluster controller (via the Karmada API Server). This allows the testbeds to retain their independence, while still following the federation guidelines.

A Cluster Agent does not consume; neither exposes any APIs from/to other functional entities in the 5G-EPICENTRE architecture. From the point of view of the user of the federation, the agent is transparent – it only affects the way the member cluster and manifests are synchronised in the Karmada control plane.

Note: At the time of writing, 5G-EPICENTRE experiments with both PUSH and PULL modes the management of member clusters. In the former, the Cluster and Execution controllers forward the tasks to the K8s API, while in PULL mode, the Cluster Agent is responsible for forwarding the tasks.

3.6 Testbed/Infrastructure Layer functional view

In this Section, the functional entities depicted in the testbed reference framework (Section 3.4.2), corresponding to the Infrastructure Layer architectural stack (Figure 32) will be presented. Thereby, individual functional entity responsibilities, and interactions with other 5G-EPICENTRE components will be elaborated. Again, the majority of sub-Sections shall imply no specific implementation, unless otherwise specified. Project-specific component descriptions are once more derived from the templates used for bottom-up design re-specification, as presented in Section 3.2.3 and in Annex III.

It is worth noting that 5GC and 5G RAN blocks are depicted to help readers better visualize the 5G-EPICENTRE end-to-end approach, particularly with respect to how the project's Network Applications ecosystem delivers interactions with the 5G network control plane. As these are independently developed in each testbed, they will not be covered in this deliverable (treated as "black boxes" in the context of the 5G-EPICENTRE architecture).

3.6.1 Kubernetes Testbed Management architecture

Each of the four testbeds in the 5G-EPICENTRE federation needs to support hosting of network and vertical applications as containerized workloads that are orchestrated and managed by a K8s architecture. Therefore, each testbed should be capable of deploying K8s, resulting in each testbed represented in the federation by a K8s cluster. Each cluster is represented in a typical K8s architecture[11], not to be dictated by this document. A testbed

[11] https://kubernetes.io/docs/concepts/overview/components/

may opt to support high-availability K8s clusters (*i.e.*, multi-master setup), so as to increase robustness of the architecture in case a master node should fail [10] (but this is not a requirement).

Similar to the Karmada control plane (as Karmada was built specifically to mirror the K8s cluster architecture, mentioned in Section 3.5.3), the K8s control plane (master node) similarly consists of the **API server**, **etcd**, **scheduler** and **controller manager**, exposing the K8s API through the server component for both internal and external (to the cluster) component communication. For the purposes of the 5G-EPICENTRE architecture, the K8s API is exposed towards its corresponding **Karmada Cluster Agent** (see Section 3.5.3.5), allowing it to run commands against the cluster (*e.g.*, deploy applications, manage resources and retrieve logs).

3.6.2 Testbed Augmented NFVI architecture

The NFVI block in each testbed corresponds to the virtualized infrastructure components (containerized, or VM-based, managed and orchestrated by K8s with the potential use of add-ons, see also Annex I) needed for the 5G-EPICENTRE Infrastructure to work in accordance to the foreseen information flows (see Section 3.7).

It is worth noting that 5G-EPICENTRE architecture remains interoperable with both virtualized and bare-metal implementations of the 5GC[12], since K8s can be deployed both over virtualized and bare-metal infrastructures [11]. Hence, the NFVI block may integrate a 5GC VM as part of the infrastructure, or be a separate entity. The specifics of each 5GC implementation are not to be dictated in this document, as each testbed integrates its own implementation.

3.6.2.1 Analytics Engine: Analytics Driver

The Analytics Driver (henceforth referred to as "the Driver") is a core component of the 5G-EPICENTRE Analytics Engine and the frontend of the analytics pipeline. It is responsible for collecting metrics information generated at the testbed level by the infrastructure and the vertical applications being experimented with. Its purpose is to pre-process the data in order to forward them to the other Analytics Engine components for KPI calculation and Machine Learning (ML)-based analysis. The metrics collected could be related to:

- Network performance (*e.g.*, latency, usage, bandwidth, resource consumption), performance of container management actions (*e.g.*, service creation time). Both originate at the same, or separate Prometheus instances (see Section 3.6.2.6).
- Traffic conditions, originating at the Remote iPerf Agents (see Section 3.6.2.5).
- Specific KPIs of interest to the experimenters, originating at vertical application components. On these metrics the vertical can request Analytics Engine services under the Network Application model (see Section 3.6.3.3).

The metrics are obtained via asynchronous communication established with the Publisher (see Section 3.6.2.4), which is responsible for associating the different kinds of metrics to the proper experiment metadata for identification purposes. Apart from subscribing to the Publisher's message queue, the Driver further publishes metrics to both the KPI (see Section 3.6.2.2) and QoS/QoE Monitoring (see Section 3.6.2.3) services.

3.6.2.2 Analytics Engine: KPI Monitoring service

The KPI Monitoring service (henceforth referred to simply as "the KPI Monitor") is tasked with carrying out analysis of metrics input data from the testbed exposed by the Driver, in order to calculate KPIs (when needed, for instance, for KPIs calculated over a predefined number of iterations, as described in D1.6), monitor metrics for prescribed experiment KPI validation and calculating statistics on specific time windows. It further stores computed values in order to track KPIs' evolution over time.

[12] The 5GC present in the HHI testbed is not a virtualised core, but as a hardware implemented solution.

The KPI Monitor then provides computed KPI values to the **Analytics Aggregator** at the Back End Layer, using the latter's Southbound communication mechanism (see Section 3.5.2.4).

3.6.2.3 Analytics Engine: Quality of Service / Quality of Experience service

The Quality of Service / Quality of Experience (QoS/QoE) Monitoring service (henceforth referred to simply as "the QoS/QoE Monitor") is responsible for continuously analysing the pre-processed data received from the Driver, with the primary aim being to perform anomaly detection. To do this, it is comprised of ML components that are trained on what constitutes 'normal' operation with one, or multiple experiments running concurrently on the 5G-EPICENTRE platform. This enables it to provide KPI prediction and to flag close to real-time conditions in which anomalies are detected.

It then exposes the results corresponding to anomalous network conditions or trends (*e.g.*, via a notification template) to the **Analytics Aggregator**, using the latter's Southbound communication mechanism (see Section 3.5.2.4), hence notifying it to any detected anomalies. It finally also generates a report, after the experiment, to provide a complete overview of the performed analysis. This report is also sent to the **Aggregator**.

3.6.2.4 Publisher

When vertical applications are experimented with, the measurement messages generated for the analytics components are not aware of experiment parameters, such as the experiment ID, or the scenario configured. The data is however necessary to identify the measurements and the configuration applied in the testbed cluster for a particular experiment. Therefore, the Publisher is a component charged with "adding" this missing metadata to the measurements corresponding to the experiment applications under test. It stores an experiment list with metadata to associate agnostic metrics with the proper experiment ID. If an experiment is not recognized, as a result of it not having been added to the list, the Publisher will fill metadata with predefined default values instead.

To obtain metrics, the Publisher consumes the endpoints exposed by one or more **Prometheus instances** (see Section 3.6.2.6) through REST API calls (Prometheus API, described in the Prometheus HTTP API documentation[13]). The Publisher is further configured to optionally integrate with asynchronous messaging (via the Message Broker described in Section 3.6.2.7) regarding both:

- Artificial traffic parameters related to the experiment, and generated by (at least one) pair(s) of Remote iPerf Agents (see Section 3.6.2.5).
- Vertical application-specific metrics calculation implementations (such that are not retrievable through the Prometheus instances deployed). In this respect, vertical application components delegated to the testbed owners and orchestrated on the testbed K8s cluster, are configured (externally, by the vertical application developers) to publish MQTT messages to a queue with the routing key "application" (see also Section 3.6.2.1).

As with the metrics obtained through the Prometheus API, the Publisher fills in any missing experiment metadata, and publishes new messages using a different routing key. The messages in this queue are then consumed by the Driver (see Section 3.6.2.1). It is worth noting that, for the purposes of fault tolerance, if a message consumed by the Publisher has an invalid format, or is missing any mandatory field, it is not published to the Driver.

To obtain metadata values, the Publisher exposes an API towards the **Experiment Coordinator** (see Section 3.5.2.1), which supports the following capabilities/operations (Table 9):

[13] https://prometheus.io/docs/prometheus/latest/querying/api/

Table 9: Publisher API endpoints exposed to the Experiment Coordinator

Method	Endpoint	Description
POST	/publish	Auxiliary endpoint for publishing a message using an HTTP request, for entities that cannot connect to the message queues directly.
POST	/fetch_metrics	Starts fetching new metrics from the Prometheus client if they are available, until the time specified in the "end_time" field is reached, or it is stopped by the Coordinator.
POST	/add_experiment	Endpoint for adding metadata for a particular experiment to the experiment list.
POST	/remove_experiment	Endpoint for deleting metadata for a particular experiment from the experiment list.

Since the Publisher utilizes an asynchronous communication scheme for forwarding metrics to the Analytics Driver, it does not consume any APIs exposed by other functional entities in the architecture.

3.6.2.5 Remote iPerf Agent(s)

A Remote iPerf Agent (henceforth referred to as “the Traffic Agent”) is a functional entity implemented using the iPerf tool, tasked with translating experiment traffic parameters supplied by the **Traffic Simulator** (see Section 3.5.2.2) into a workflow for generating artificial network traffic at an IP level. This traffic is generated between two Traffic Agents, one of which is instantiated as a “server” and the other as a “client” (the role and traffic properties are supplied to each agent by the Traffic Simulator, each with its own REST request). Each Traffic Agent then asynchronously dispatches messages with measures about the traffic generated or received, which are consumed by the Publisher (see Section 3.6.2.4).

As mentioned previously in Section 3.5.2.2, the Traffic Agent exposes an API towards the **Traffic Simulator**, which supports the following capabilities/operations (Table 10):

Table 10: Remote iPerf Agent API endpoints exposed to the 5G Traffic Simulation Manager

Method	Endpoint	Description
POST	/iperf	Executes iPerf process with the specific parameters in the request body in JSON format. Returns a JSON reporting the success of the execution and a message.
GET	/Close	Closes iPerf process. Returns a JSON reporting the success of the process closure and a message.
GET	/LastRawResult	Retrieve the results of the previous execution. Returns a JSON reporting the success of the retrieval, a message and a list of Results (full line).
GET	/LastJsonResult	Retrieve the results of the previous execution. Returns a JSON reporting the success of the retrieval, a message and a list of Results (dictionary with parsed results).

GET	/LastError	Retrieve the errors of the last execution. Returns a JSON reporting the success of the retrieval and a message informing of the error.
GET	/StartDateTime	Retrieve the date and time of the last execution. Returns a JSON reporting the success of the retrieval, a message and a list of Errors.
GET	/IsRunning	Check if there is another execution running. Returns a JSON reporting the success of the check and a message informing if is running or not.

3.6.2.6 Prometheus instance(s)

Prometheus[14] instances are deployed to collect, store and monitor operational metrics related to both the 5G and K8s deployments in the testbed cluster. Each testbed may implement one or more instances (*e.g.*, one for retrieving data from the network cores; another for the Kubernetes infrastructure, *etc.*), in accordance to their preferences. Each Prometheus instance exposes HTTP API endpoints to be consumed by the **Publisher** (see Section 3.6.2.4), for either configuring metrics to be stored; or querying accumulated metrics, respectively.

3.6.2.7 Message Broker

A Message Broker (henceforth referred to as "the Broker") should be deployed in each testbed to facilitate the asynchronous message-based communication between the analytics pipeline components, hence collecting, processing and transmitting data as soon as they become available. The Broker provides all the services needed for message publishing components in the architecture to route messages to a specific temporary storage address (an exchange, or queue), to be consumed by the message subscribing services at their own pace.

While any message-oriented middleware solution should work, the specific implementation opted for by all testbeds is RabbitMQ[15], extended with the MQTT plugin to support the OASIS MQTT protocol.

3.6.3 Network Application Layer

In line with the Network Applications ecosystem initiative, 5G-EPICENTRE incorporates the Network Applications perspective from the combined ICT-41 projects [2] into the architecture as a separate layer from the PaaS components. This is not to state that a Network Application differs much from the PaaS components described in the previous Section in terms of deployment, since 5G-EPICENTRE treats, orchestrates and manages Network Applications as part of the testbed cluster. However, for the purposes of demystifying the concept of Network Applications with some concrete examples, entities identified under the concept are considered as parts of this architectural Layer.

Network Applications in 5G-EPICENTRE, as specified in Section 3.3.1, interact with the control plane functions of the 5GC network and expose features that are relevant for vertical applications, such as location or QoS. Since they interact as (or via) AFs, and according to the level of trust, 5G-EPICENTRE Network Applications can interact either directly with 5GC network functions (if the Network Application is considered to be trusted), or through the services exposed by the 5GC NEF (in an untrusted situation, *e.g.*, a Network Application, onboarded to one of the 5G-EPICENTRE testbeds by a third party). To illustrate these varying levels of trust, we make a distinction between *platform Network Applications* and *vertical Network Applications*.

Platform Network Applications are all the 5G-EPICENTRE functional components that are defined as Network Applications following the definition given in [2]. They are considered agnostic to the vertical application and (being internal project implementations) can be considered trusted elements (directly communicating with the 5GC functions, bypassing the NEF). Within 5G-EPICENTRE, several such entities have been identified, re-purposed

[14] https://prometheus.io/
[15] https://www.rabbitmq.com/

from their original roles as platform components, frameworks, or APIs (in accordance to the material reported in D1.3). These are the **Configurator** (see Section 3.6.3.1); the **Network Intrusion and Detection System**; and (as previously mentioned) the **Analytics Engine services**, comprised of the components reported in Sections 3.6.2.2 and 3.6.2.3.

Vertical Network Applications correspond to the parts of the vertical application which is delegated to the telco/testbed operator under the 'Delegated', or 'Hybrid' options of interaction (recall Figure 29 from Section 3.3.1.2). For illustrative purposes, the block incorporates (hypothetical) vertical application components delegated by the UC owners for each of the eight foreseen UC vertical applications to be experimented with in the context of the project's WP5, however, it encapsulates all vertical application parts that are delegated by third parties using the processes available to experimenters and function developers at the 5G-EPICENTRE Portal. Therefore, these components operate on a mutual agreement of trust, set when the experiment is requested using the Portal experiment definition interface and ensuing experimenter-testbed owner communication. It is important to note here that all elements in this block are not to be considered part of the 5G-EPICENTRE architecture, but rather as extensions that might be offered to a third-party experimenter based on decisions made at the UC level regarding openness of the vertical Network Application component. Therefore, a description of these elements is beyond the scope of this deliverable (Network Application implementation within the project, either Platform or Vertical, will be described in detail in deliverable D4.2).

The following sub-Sections briefly elaborate on the platform Network Applications, since those are considered parts of the platform offered for experimentation. As mentioned, more details regarding implementation and foreseen properties of these blocks as Network Applications will be reported in D4.2.

3.6.3.1 Configurator Network Application

This Network Application aims at simplifying 5G service exposure, allowing verticals (*e.g.*, UCs) to request specific conditions from the 5G Core network. It corresponds to functionality previously associated with the *Software-defined Northbound API* block (D1.3 and D3.1, see also Section 3.1). Its purpose is to wrap different network features that PPDR UCs are more likely to want to access and configure (based on UC owners' requirements), and make them available as configurable (to some predefined extent) parameters mapped to the **Northbound Configuration Dashboard** UI environment (see Section 3.5.1.2). For instance, the Network Application can communicate directly with the 5GC Policy Control Function to request a 5QI so that verticals can request (and the platform can demonstrate) prioritization of their traffic in severely congested network conditions (simulated using the traffic components available to the 5G-EPICENTRE platform and testbeds, see Sections 3.5.2.2 and 3.6.2.5).

The Configurator Network Application exposes a simple REST API to vertical applications in either the vertical service provider domain, or the testbed domain, enabling the experimenter to request (via the Dashboard in the front-end Layer, or via another vertical application component consuming this API) exercising QoS management over the network.

3.6.3.2 Network Intrusion Detection Network Application

This Network Application comprises the Network Intrusion and Detection System (NIDS) described in D2.1, charged with detecting anomalous traffic that can be traced back to a set of security attacks. It comprises a set of collection agents, to collect traffic data; and an **Analytics, Intelligence, Control and Orchestration (AICO)** component, which detects and classifies anomalous flows as potential threats and can trigger security policies for blocking the origin of the anomalous traffic.

The NIDS Network Application requires and assumes an Istio service mesh implementation[16] for the vertical application under test. It can then expose a REST API to the vertical application components in either the vertical

[16] https://istio.io/latest/

service provider domain, or the testbed domain, enabling the experimenter to act upon any irregularities (*e.g.*, identify source of anomalous traffic, configure/override security policies, *etc*.).

3.6.3.3 Analytics Services Network Application

The services described in Sections 3.6.2.1 to 3.6.2.3 comprising the Analytics Engine block, are an integral part of the 5G-EPICENTRE platform architecture, since they are eventually consumed by the platform front-end components to deliver the platform's experiment monitoring capabilities. The capabilities of the Analytics Engine are further exposed as a Network Application, so that verticals can take advantage of the capabilities provided (*e.g.*, perform statistical analysis on predefined time intervals) to handle vertical application-specific measurements (*i.e.*, metrics that are calculated directly by the vertical). The Analytics Engine can perform statistical analysis on the indicated measurements, and send the results to the Portal through the analytics pipeline path described in the aforementioned Sections.

The Analytics Engine hence exposes a simple REST API that can be consumed by vertical application components, in order to request the Analytics Engine to provide the available analytics services for the measurements, subscribing the services for their specific vertical application ID (the Analytics Driver will otherwise "ignore" messages sent from verticals that are not subscribed).

3.7 Information view

The architectural information view of 5G-EPICENTRE is derived from defining the necessary interconnections (*i.e.*, interfaces) between the specified functional architecture components, elaborated in the previous Section. For each functional block, the list of API endpoints exposed towards other elements, or consumed by other entities in the architecture, has been thoroughly listed. For each API, the updated bottom-up template (Annex III) was used to capture information elements exchanged over each endpoint, toward generating API documentation following the OpenAPI Specification[17]. Both synchronous (OpenAPI) and message-driven (AsyncAPI[18]) APIs were hence specified by project partners, responsible for their development, in a similar manner. The resulting API documentation for each individual functional element will be described in its corresponding deliverable (not within the scope of this architectural update).

Figure 36 depicts the 5G-EPICENTRE Information Flow Diagram (IFD), layered on top of the functional architecture, in order to illustrate the manner in which information flows through the platform. Through the IFD, the way that information is exchanged between functional elements in the architecture during different usage scenarios is highlighted.

A more elaborate description of these information flows during selected, four (4) indicative functional scenarios of the 5G-EPICENTRE platform will be presented in the following sub-Sections with the use of sequence diagrams. These will aim at depicting the information exchanges triggered during individual processes and foreseen platform use cases (not to be confused with the experimentation UCs). For each of the diagrams presented, the roles and responsibilities of each components partaking in the information exchange will be elaborated.

3.7.1 Delegating vertical application components to the testbed operator

The diagram depicted in Figure 37 illustrates how the platform can be used by a vertical end user (authenticated on the platform as either an experimenter, or function developer, in accordance to the roles specified in D3.3), so that vertical network application components can be instantiated in the testbed operator domain. The action will allow these components to be composed and managed by the service provider (*i.e.*, testbed owner), following the 'Hybrid', or 'Delegated' option for interaction between the vertical and the telco [2].

[17] https://swagger.io/resources/open-api/
[18] https://www.asyncapi.com/

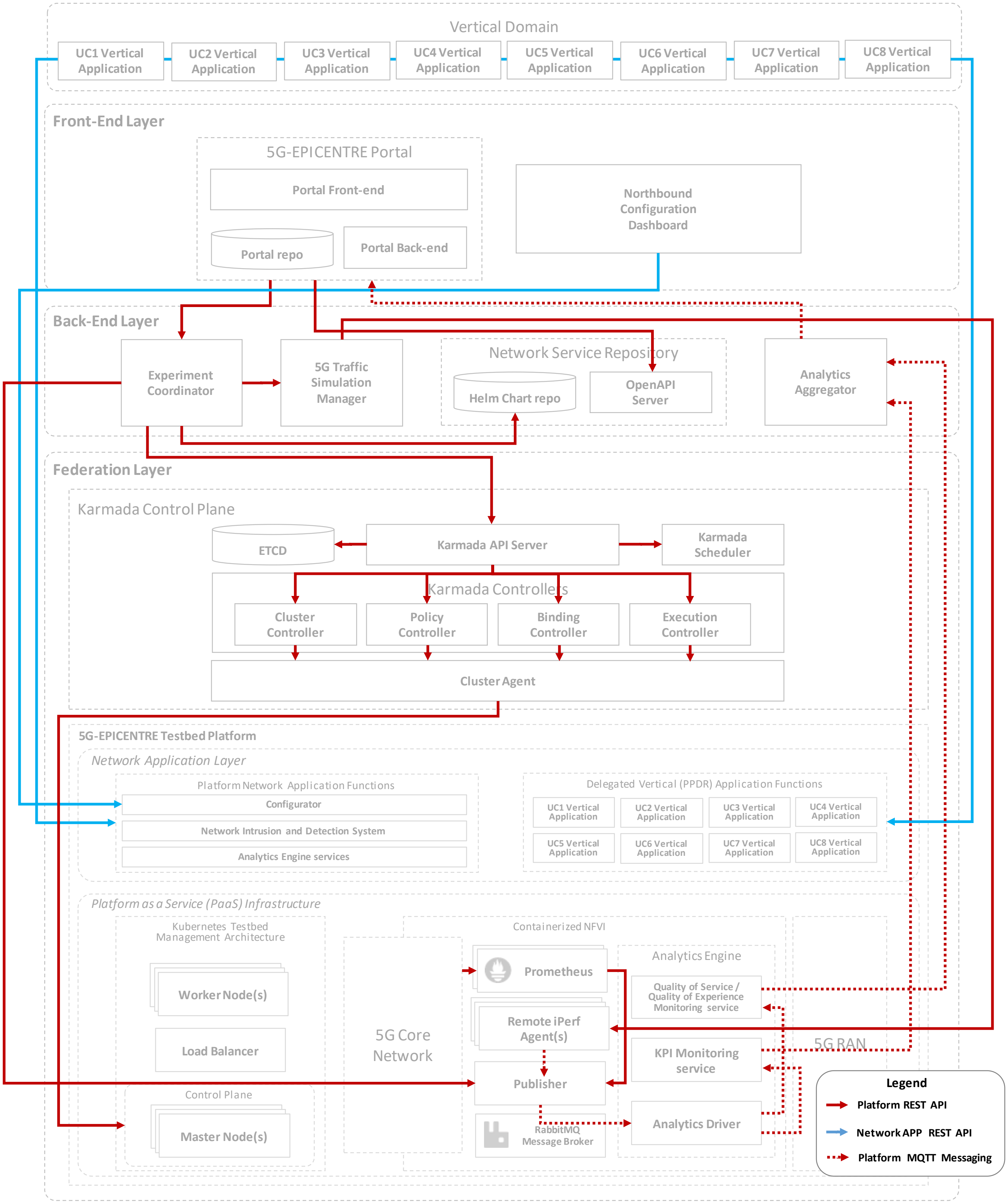


Figure 36: 5G-EPICENTRE high-level Information Flow Diagram

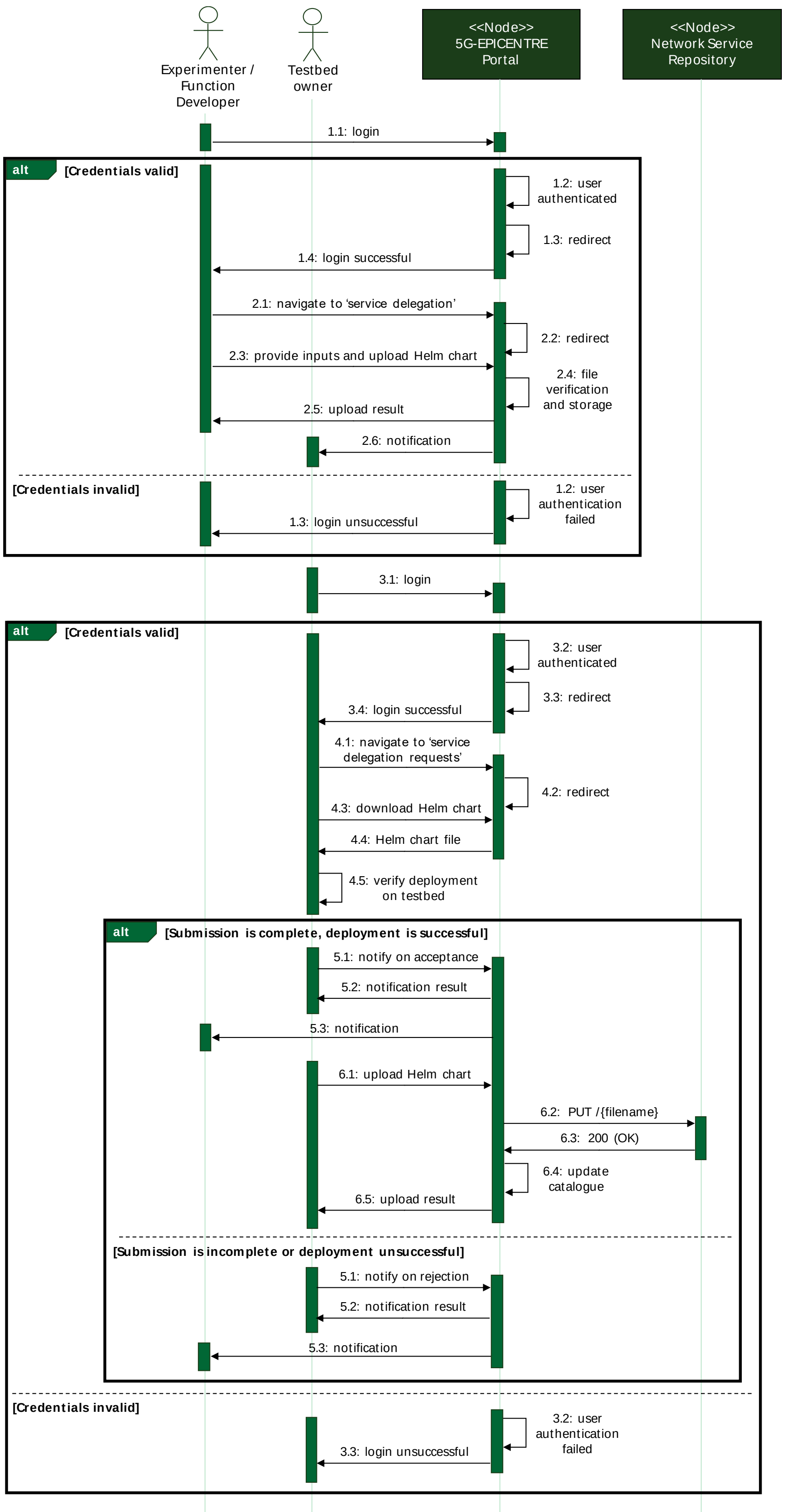


Figure 37: Sequence diagram for delegating (part of) the vertical application to the testbed using the 5G-EPICENTRE platform

At first, the end user authenticates on the **Portal** and consumes the user interfaces provisioned to create a new 'service delegation' request; fill in the required input fields (*e.g.*, service name, organization, desired testbed cluster for the deployment, openness to third parties, *etc*.); and upload (and thus share) a Helm chart package (*e.g.*, a .tgz, or .zip file), making it available to a user with the 'testbed owner' role (note that the responsibility of creating the chart, and packaging it remains with the vertical application developer – remember that a Helm chart is an archive that contains all resource definitions for streamlining the deployment of an application to a K8s cluster). Once the request is submitted, the chart should become available to the testbed owner, who can download the packaged Helm chart file and install the chart archive to deploy the vertical's software onto the testbed's K8s cluster for verification of its functionality (based on the information/instructions provided in the mandatory README Markdown file and/or NOTES plain text file included in the chart package).

According to the verification of the service, the testbed owner might either request modifications, reject altogether, or accept the delegation request (both actions using the **Portal** reviewing mechanism). In the latter case, the testbed owner invokes the **Repository** Helm chart upload mechanism (via the Portal consuming the Repository's northbound API), to make it available to the *Experiment Coordinator* upon experiment deployment (see Section 3.7.3).

3.7.2 Scheduling an experiment

The diagram depicted in Figure 38 illustrates how the platform can be used by a vertical end user (authenticated on the platform as an experimenter), to request execution of an experiment with a vertical application. For the purposes of this particular scenario, it is assumed that both experimenter and testbed owner(s) have authenticated in the **Portal**, (since the login functionality remains constant across all provided Portal functions, and is already depicted in the diagram shown in Figure 37).

When requesting a new experiment, the experimenter uses the **Portal** to essentially instruct the testbed owner to setup the deployment of all prior delegated vertical (or platform) Network Applications, which shall expose APIs for the vertical application components deployed in the vertical service provider domain. Selection is made from the list of Network Application Helm charts made visible to the user (based on the "openness" property defined during the delegation of such Network Application components – see Section 3.7.1). The **Portal** receives the list of available Helm charts by making the proper API calls to the **Repository**.

The process further involves optional specification of network traffic simulation parameters, along with platform metrics and KPIs to be monitored, selected from a list of predefined values. In case of custom metrics pre-processing (*e.g.*, via proprietary vertical application components delegated to the testbed owner prior to this activity), information on the routing keys where the components publish to is entered as well (for the *Coordinator* to configure the *Publisher* at the experiment deployment stage, see Section 3.7.3).

The experimenter finally requests the reservation of one or more testbed clusters at a specific date and time, as well as the need for the platform to expose a *Northbound Configurator Dashboard* instance for network control plane configurations.

Once all required input fields are filled in, the request is forwarded to the testbed owner(s) responsible for the testbed cluster deployment.

The testbed owner(s) receive the experiment scheduling request, and according to the experiment description, might either request modifications, reject altogether, or accept the delegation request (both actions using the **Portal** reviewing mechanism). In the latter case, the testbed owner invokes the **Coordinator** scheduling mechanism, to "program" the platform to automatically deploy the vertical application cluster upon arriving at the predesignated date and time. If a *Northbound Configurator Dashboard* is requested, a (generic) instance is configured and the URL to it is added to the experiment entry in the Portal.

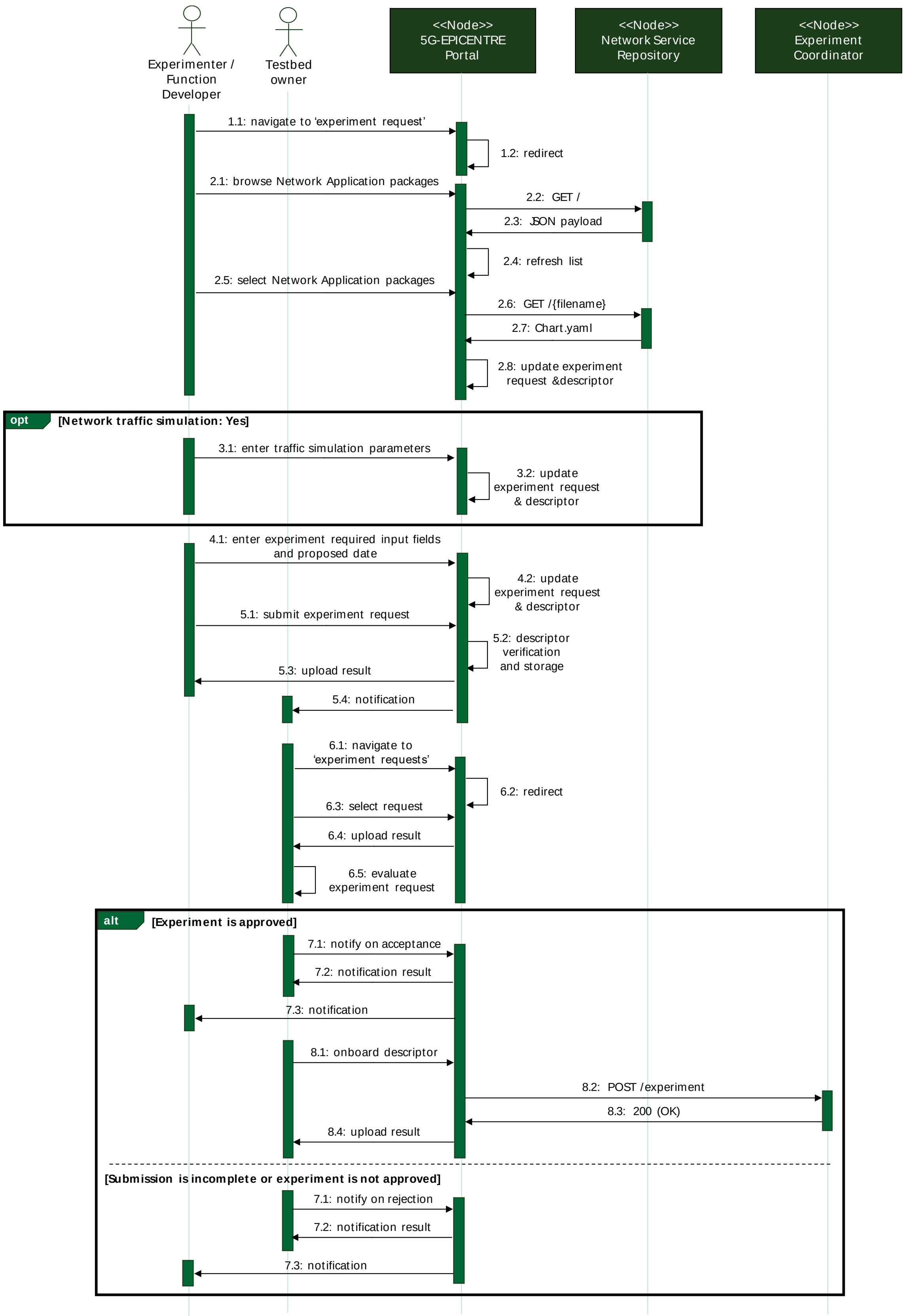


Figure 38: Sequence diagram for scheduling an experiment using the 5G-EPICENTRE platform

3.7.3 Experiment deployment

The diagram depicted in Figure 39 illustrates how the platform deploys the experiment vertical and platform application components through the Karmada cross-cluster management stack.

Upon arrival at the designated date and time to deploy and start the experiment execution, the **Coordinator** will call the Karmada API so that the **Karmada API Server** can install the application components packaged in the designated Helm charts from the *Repository*, indicated in the experiment deployment YAML file. Through the **Karmada Scheduler**, the application components packaged in the charts are distributed across the optimal K8s member clusters. Utilizing the **Karmada API Server** to establish communication between the **Policy controller** and the cluster's **Karmada Agent**, the latter coordinates with the cluster's K8s master node to deploy the Helm chart application contents to the testbed's K8s infrastructure. The most suitable node of that cluster is elected, through a similar scheduling procedure involving the cluster's K8s API server and Scheduler. When the deployment is finished, the **Karmada API Server** stores all K8s/Karmada API objects to the **etcd**, and notifies the **Coordinator**.

Once all vertical application and platform components and Network Applications are deployed, and in accordance with the experiment descriptor, the **Coordinator** consumes the **Publisher's** API, to: (i) add the experiment's metadata to the latter's experiment list; and (ii) to configure, through the **Publisher**, the metrics of interest to be monitored and collected by the *Prometheus instance(s)*.

Finally, provided specific traffic simulation conditions are requested, the **Coordinator** makes API calls to the **Traffic Simulator**, in order to supply the latter with the network traffic to be simulated (all flows), information on which pre-deployed *Traffic Agents* to Configure and start, and the topic exchanges where the Traffic Agents must dispatch messages with the traffic data generated, or received. The **Traffic Simulator** then makes API calls to each **Traffic Agent** involved in the traffic simulation request, sending it its role (client or server), traffic characteristics and publishing information; and proceeds to start it.

3.7.4 Experiment execution

The diagram depicted in Figure 40 illustrates how the platform components behave (normally) during experiment execution. For the purposes of this particular scenario (since other experiment execution options are also supported), it is assumed that an experimenter has authenticated in the **Portal**, and redirected themselves to the visualization dashboard environment.

During execution of the experiment, the **Prometheus Instance**(s) gather(s) the pre-designated metrics, making them available to the **Publisher** upon request at regular intervals. The Publisher further optionally "listens" for messages in the specified queue configured for traffic simulation (published to by the **Traffic Agents**). In each case, it fills in the experiment metadata, and re-routes the newly formatted messages to a different queue to be consumed by the subscribing **Driver**. The latter then pre-processes this data, and forwards it to the other analytics engine components (**KPI Monitor** and **QoS/QoE Monitor**). Each function performs its routines, and in turn publishes its output messages to the **Aggregator**. Provided a multi-cluster deployment, the **Aggregator** accumulates all messages received in a pre-specified time window, and routes a new, aggregated message to the **Portal** subscribed to its queue. The **Portal** finally unpacks the message, and updates the visualization environment to depict experiment KPIs' evolution over time.

The entire process is repeated until the experiment is terminated.

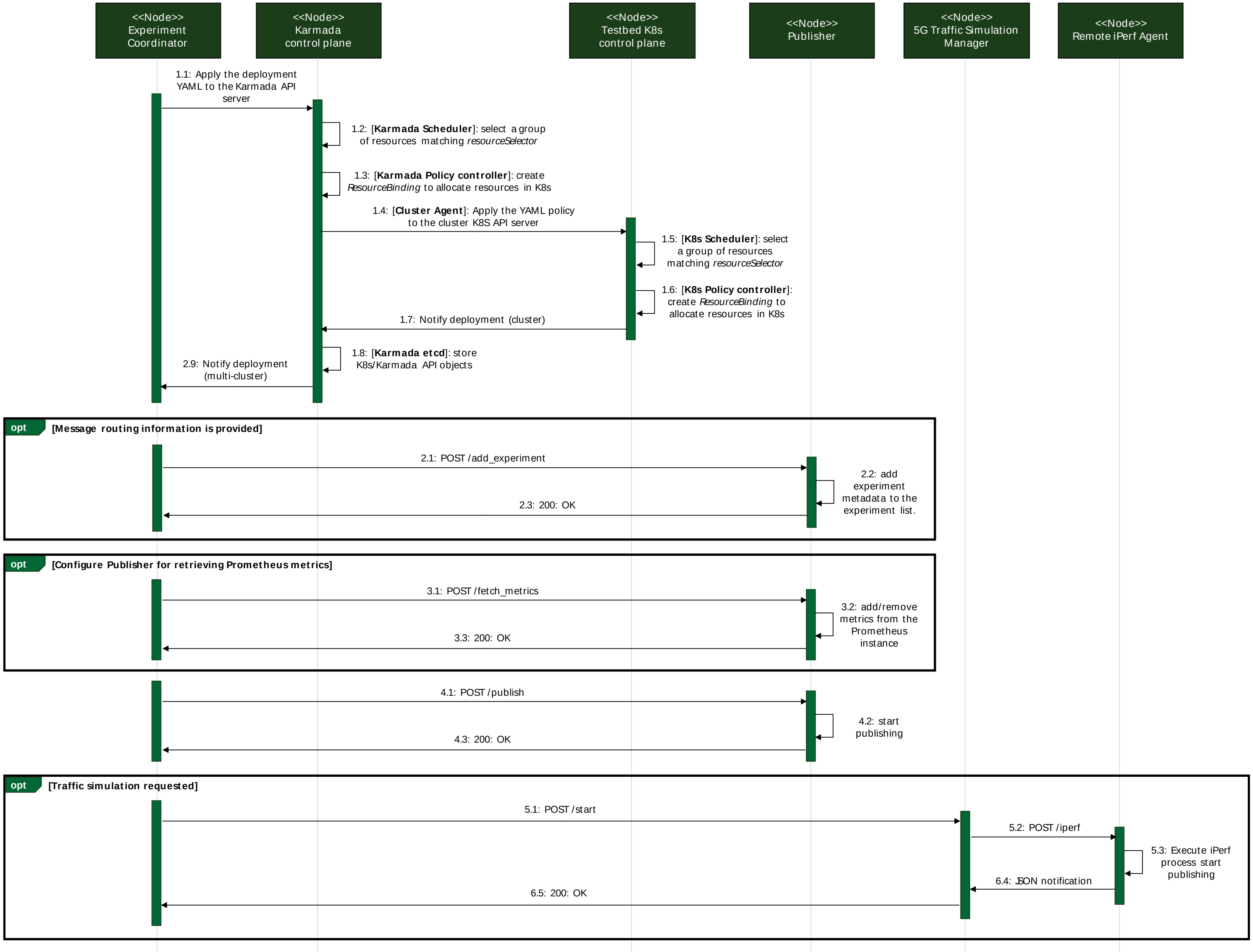


Figure 39: Sequence diagram for (automated) experiment deployment within the 5G-EPICENTRE platform

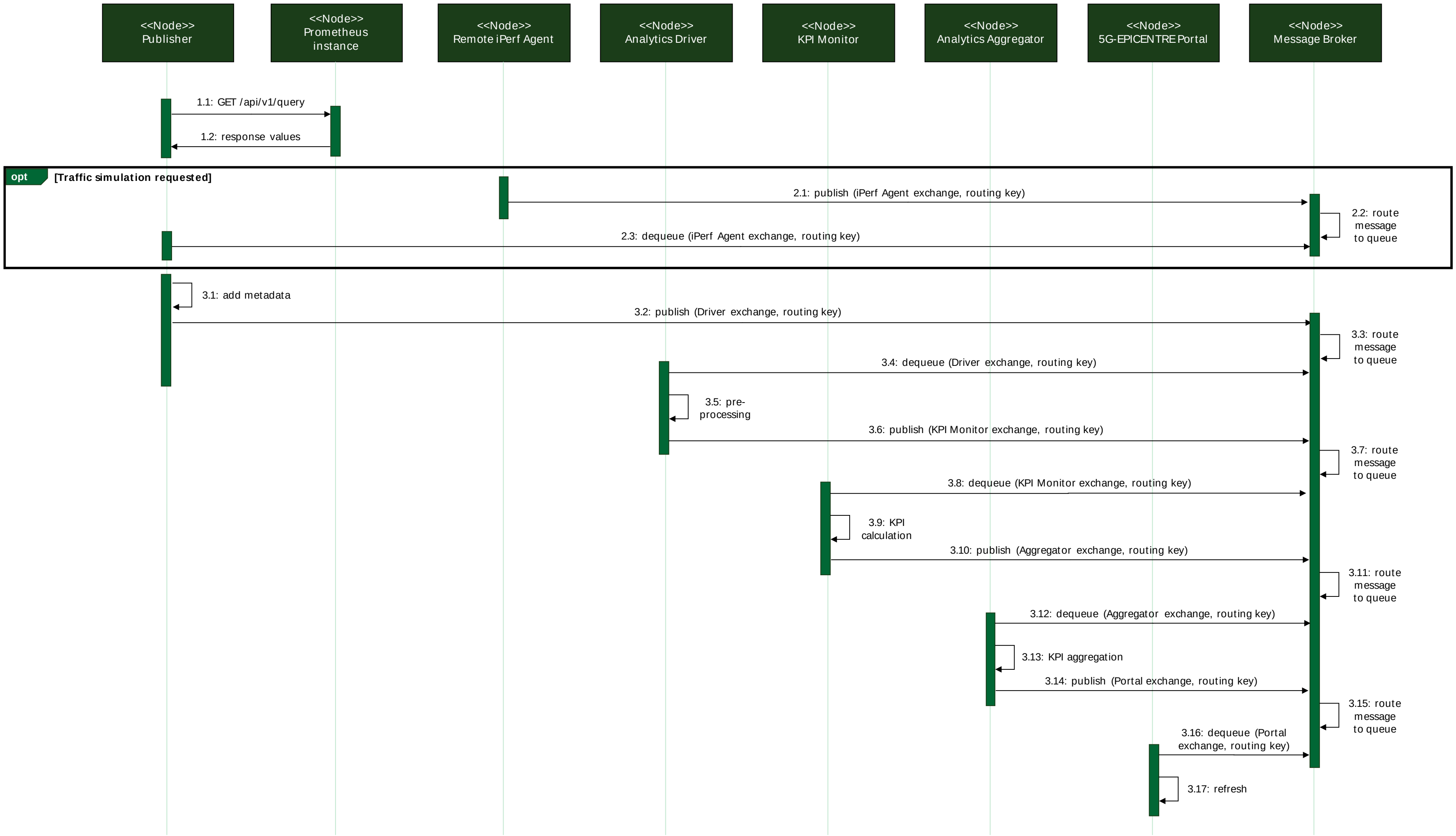


Figure 40: Sequence diagram for (monitored) experiment execution within the 5G-EPICENTRE platform

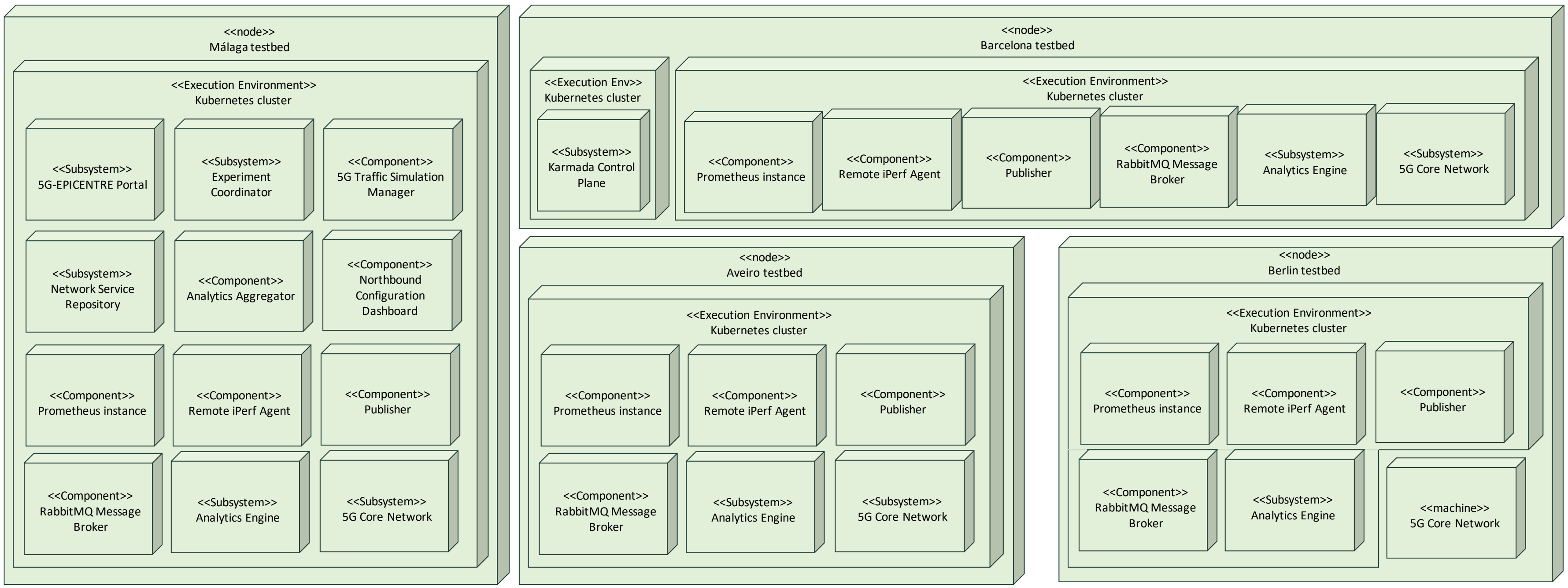


Figure 41: 5G-EPICENTRE Deployment Diagram

3.8 Deployment view

The 5G-EPICENTRE architecture deployment view is used to define the physical environment for the deployment of the system across virtual and real machines on the available infrastructure across the four facilities. The deployment view is meant to describe how the 5G-EPICENTRE software will be deployed into the available hardware, and how it will interact with that hardware to deliver the platform's functionality. For illustrative purposes, it will be elaborated by means of the deployment diagram shown in Figure 41.

As can be seen, 'Infrastructure Layer' components are replicated across the facilities. Centralized Back-end and Front-end functional elements are hosted on the Málaga premises. The Karmada control plane is deployed in the CTTC testbed

3.9 Security view

The security view of the architecture places emphasis on how engineers are to approach security concerns related to the architectural structure, in order to ensure its integrity against outside tampering. Security is a major topic addressed in the context of WP2, and more specifically, Task T2.6. Primary security considerations for 5G-EPICENTRE are reported in deliverable D2.1. The key implications of that report for the architecture security perspective are summarised in the following points:

- The architecture shall support integration with a **service mesh** implementation, to manage API traffic among services and support security and observability. The service mesh is introduced into the vertical application under test as a set of network proxies (separate containers called *sidecars*, running in parallel to the application/service container). These proxies do not introduce functionalities to execution environment, and are therefore not to be considered as functional entities in the architecture. A service mesh can easily and transparently be integrated onto a K8s deployment using the Istio service mesh, without modification to the building blocks described in this Section.
- A Network Intrusion and Detection Network Application can be exposed to experimenters to detect traffic anomalies related to security incidents, and effectuate a response on a policy level (see Section 3.6.3.2).

4 Conclusions

This deliverable corresponds to the conclusive insights from 5G-EPICENTRE WP1 Tasks T1.2 and T1.3. On the one hand, the document recounts the activities toward technical requirements' validation from an informed perspective of the 5G-EPICENTRE platform stakeholders, whilst on the other, it summarises the penultimate version of the 5G-EPICENTRE platform architecture, designed to meet those requirements.

In terms of requirements' validation, effort has been spent toward managing 5G-EPICENTRE platform stakeholders' expectations with respect to the final system to be delivered. To elicit this information, a wide-spread electronic survey was prepared, attracting interest from entities making their first contact with the 5G-EPICENTRE goals and visions. This has allowed the Consortium to further refine what the 5G-EPICENTRE solution should be from those stakeholders' perspective (*e.g.*, identifying users and operational scenarios with insight coming outside the Consortium's own interpretation of needed system functions). Given this information, the original comprehensive set of platform's technical (functional and non-functional) requirements was verified and validated by the same stakeholders, to re-evaluate need, correctness and completeness of the specifications they dictate regarding desired properties that the system should possess.

In order to refine the final version of the architecture specified in this document, the phased approach elaborated in the precursor deliverable D1.3 was refined, in order to account for changes discovered during the project execution, as well as the project's contributions to the definition of the Network Application concept with other ICT41 projects. This latter goal was pursued to implement support for common aspects of the envisioned Network Applications' delivery model into the 5G-EPICENTRE envisioned experimentation platform, and align the project with potential joint standardisation initiatives. This second version of the architecture was defined following activities carried out in the period M12-M24. In particular, the preliminary architecture reported in D1.3 was carefully reviewed by taking into account inputs from the technical work packages (*i.e.*, WP2, WP3 and WP4); the verified and validated list of requirements from T1.2 (available in Section 2 of this document); and the project Use Cases' vertical application transferal effort to the ICT-41 common Network Application model, which is to be demystified further in D4.2. Following this approach, the 5G-EPICENTRE architecture was refined and re-documented. Four different views were specified in full, namely, functional, information, deployment and security perspectives, each imparting critical information regarding the final system implementation activity. This will culminate in the next six months, as a final version of the platform is scheduled for M30 (although earlier iterations will be pursued, given the advanced maturity of the implemented components).

All functional elements described in this document constitute final versions with respect to their exposed capabilities and interdependencies. They will hence be embedded into the final platform "as described", deployed for the purposes of experimentation with Consortium platform stakeholders, for demonstrating 5G capabilities to PPDR end users and other innovators. The platform will further be employed to support the foreseen third-party experimentation activities to take place along the lines of WP5. From the point of view of the implementation, the Consortium shall strive to deliver on the technical requirements as presented, and deliver each functional block in accordance to the specifications listed in this report, with no major deviations.

The deliverable marks the conclusion of both Tasks 1.2: "Requirements of 5G experimentation infrastructures for PPDR use case innovations" and T1.3 "5G-EPICENTRE technical specifications and architecture".

References

[1] Arampatzis, D., Apostolakis, K. C., Margetis, G., Stephanidis, C., Atxutegi, E., Amor, M., et al. (2021). Unification architecture of cross-site 5G testbed resources for PPDR verticals. In *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)* (pp. 13-19). IEEE. https://doi.org/10.1109/MeditCom49071.2021.9647591

[2] Sayadi, B., Chang, C.-Y., Tranoris, C., Iordache, M., Katsaros, K., Vilalta, R., et al. (2022). *Network Applications: Opening up 5G and beyond networks* [White paper]. Zenodo. https://doi.org/10.5281/zenodo.7123919

[3] 3rd Generation Partnership Project. (2021, December) 5G; System architecture for the 5G System (5GS) (3GPP TS 23.501 version 16.11.0 Release 16) [Technical Specification]. https://portal.etsi.org/webapp/workprogram/Report_WorkItem.asp?WKI_ID=64055

[4] IEEE Computer Society. (1990). IEEE standard glossary of software engineering terminology. *IEEE Std 610.12-1990*, 1-84. IEEE. https://doi.org/10.1109/IEEESTD.1990.101064

[5] International Standards Organisation. ISO/IEC JTC 1/SC 7 Software and systems engineering Technical Committee. (2011). 42010-2011 - ISO/IEC/IEEE Systems and software engineering -- Architecture description. *ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000)*, 1-46. IEEE. https://doi.org/10.1109/IEEESTD.2011.6129467

[6] Wang, K. & Yifan, S. (2022, March 22). Karmada: Multi-cluster Management with an Ocean of Nodes [Blog post]. *Cloud Native Computing Foundation*. https://www.cncf.io/blog/2022/03/22/karmada-multi-cluster-management-with-an-ocean-of-nodes/

[7] Brown, G. (2021, April 20). Cloud infrastructure for the 5G core [Blog post]. *LightReading*. https://www.lightreading.com/5g/cloud-infrastructure-for-5g-core/a/d-id/768873

[8] Margetis, G., Valera-Muros, B., Apostolakis, K. C., Díaz Zayas, A., Panizo, L., Tomás, P., et al. (2022). Validation of NFV management and orchestration on Kubernetes-based 5G testbed environment. In *2022 IEEE Globecom Workshops (GC Wkshps)* (pp. 844-849). IEEE.

[9] Wang, K. (2020). *Karmada: Open, Multi-Cloud, Multi-Cluster Kubernetes Orchestration* [GitHub repository]. https://github.com/karmada-io/karmada

[10] Hakvoort, S. (2021, August 13). Setting up a high-availability Kubernetes cluster with multiple masters [Blog post]. *Medium*. https://medium.com/@sven_50828/setting-up-a-high-availability-kubernetes-cluster-with-multiple-masters-31eec45701a2

[11] Bäckström, H. & Bohra, R. (2022, May 3). Why Kubernetes over bare metal infrastructure is optimal for cloud native applications [Blog post]. *Ericsson Blog*. https://www.ericsson.com/en/blog/2022/5/kubernetes-over-bare-metal-cloud-infrastructure-why-its-important-and-what-you-need-to-know

Annex I: Electronic survey (Round 2) – Questionnaire

Informed consent form to participate in research activities of the Project

5G ExPerimentation Infrastructure hosting Cloud-nativE Netapps for public proTection and disaster RElief

Please read carefully this document before your participation

Key contact details

NOVA TELECOMMUNICATIONS SINGLE MEMBER SA (NOVA)

Survey coordinator / contact details

Mr. Konstantinos Tsiomos

email: [redacted]

Key information about the Project

Project Coordinator: AIRBUS DS SLC
Funding Program: Horizon 2020
Website: https://www.5gepicentre.eu
Contact: info@5gepicentre.eu

Description of the research project

LTE-Advanced systems and 5G are key technologies for future mission-critical public protection and disaster relief (PPDR) services. Contributing to this field, as well as lowering barriers to 5G adoption, the EU-funded **5G-EPICENTRE** project will develop based on a Service oriented Architecture, following the current best DevOps practices (containerization of micro-services), an open, end-to-end, experimentation 5G platform that focuses on software solutions that serve the needs of PPDR. It aims at facilitating adoption of current services to capitalise on 5G networks advanced capabilities as well as to provide a solid ground for the proliferation of such services.

The project will pave the way providing concrete experiments and libraries of PPDR oriented VNF chains and Netapps. Through this platform, SMEs and developers will be able to learn about the latest 5G applications and approaches for first responders and crisis management and build up and experiment with their solutions. **5G-EPICENTRE** aims at establishing an environment that bridges the development and operation worlds for PPDR services capitalising on current and upcoming 5G network opportunities.

The project lasts three years (running from January 2021 to December 2023) and is funded by the European Commission (contract no. 101016521), within the context of the Horizon 2020 programme. The project partners are:

- AIRBUS DS SLC
- NOVA TELECOMMUNICATIONS SINGLE MEMBER AE
- ALTICE LABS SA
- FRAUNHOFER GESELLSCHAFT ZUR FORDERUNG DER ANGEWANDTEN FORSCHUNG EV
- IDRYMA TECHNOLOGIAS KAI EREVNAS
- UNIVERSIDAD DE MALAGA

- CENTRE TECNOLOGIC DE TELECOMUNICACIONS DE CATALUNYA
- ISTELLA SPA
- ONE SOURCE CONSULTORIA INFORMATICA LDA
- IQUADRAT INFORMATICA SL
- NEMERGENT SOLUTIONS S.L.
- EBOS TECHNOLOGIES LIMITED
- ATHONET SRL
- REDZINC SERVICES LIMITED
- OPTOPRECISION GMBH
- YOUBIQUO SRL
- ORAMAVR SA

Purpose of the study and of data collection

This study aims at collecting the views, requirements, comments, and proposals of participants with regard to 5G-EPICENTRE platform in order to obtain information on useful features from a sufficient sample of potential platform end-users (in this case, the SMEs and PPDR service providers.

Personal data

Your identity data (full name, contact details) will not be collected for the purposes of the study.

Data processing / confidentiality

In the context of the study, only the personal data that is absolutely necessary for conducting the relevant research will be collected and processed through the "pseudonymization" process. Your participation in this study will remain confidential and your identity will not be stored by any means along with your replies or screenshot data. Your personal data will receive a code number and the digital list linking your name to that number will be stored in a secure, locked digital file. When the data is used, your name will not be displayed under any circumstances. The participants' data are protected and kept safe throughout the Project. After the completion of the Project, the list linking your name to the code number of your data will be deleted. The data processing and analysis will be carried out by the processing coordinator of this study as well as by processing coordinators of the partners. The results of the study may be published in scientific journals and conferences in an anonymous form.

Participation (benefits / motives)

Your participation in the present study is on voluntary basis. No financial reimbursement is provided for the participation in the present study.

Right to refuse or withdraw

As participants volunteering in the study, you are free at any time and until the end of the study to refuse to participate or withdraw your participation / consent for the data collected in the study. There are no risks of adverse consequences for withdrawing participation and you do not need to justify your decision.

In this case your personal data will be destroyed.
You can withdraw your consent to this study at any time, in writing to **Mr. Konstantinos Tsiomos** (see contact details at the beginning of this document).

Use of your views in other research activities

Following the analysis of your views, your views might be included in a piece of work submitted to the European Commission and scientific research papers. These results will be anonymized and will therefore not include your name. Any of these data that are not used in the reporting of results will be destroyed when they are no longer needed, and, in any case, five years after the end of the project in order to be able to fulfil reporting obligations to the European Commission (whichever is sooner).

Potential risks

The 5G-EPICENTRE consortium foresees no physical or psychological risks which could materialise from your participation in this research activity.
You will not be pressured or coerced into participating in this research activity. You are free to leave at any point, with no negative consequences.

Applicable regulations

The protection of natural persons in relation to the processing of personal data is a fundamental right. The law provides specific rights for natural persons (data subjects) and sets specific obligations for those who keep and process such data (controllers). All applicable EU and national legal frameworks and guidelines on the protection of personal data, as derived from the application of the "General Data Protection Regulation (EU 679/2016)", are being considered in this study.

Rights of participants

In accordance with principles of research ethics and EU data protection regulations, you have rights regarding how your personal data is processed.
You have the right to be informed about your personal data collected in and for the purposes of this study and to have access to them, and the right for these data to be in a portable and easily accessible form. You also have the right to request that your personal data be corrected, updated, or deleted, the right to have the processing restricted, and the right to object, with the reservation of any exceptions provided for in existing European and national legislation. We acknowledge also to you, that in accordance with the aforementioned Regulation, you have the right to file a complaint to the corresponding national Data Protection Authority (complaints@dpa.gr).
We aim to fulfil all requests. In accordance with data protection legislation, some requests may be rejected.

Who to contact if you have questions or want to exercise your rights

In order to exercise your rights or you have any questions, as far as this study is concerned, you may contact **Mr. Konstantinos Tsiomos** ([redacted]) or send your requests at [redacted].

Page 1 of 6

Questions marked with a * are required

Certificate of Consent

By submitting this form, you hereby declare that:

1. You agree to participate in this study, in the context of the Project **5G-EPICENTRE** - *"5G ExPerimentation Infrastructure hosting Cloud-nativE Netapps for public protection and disaster RElief"*.
2. You have been explained in writing through this information sheet: the purpose of the study, the respective activities, and your rights.
3. You are participating voluntarily and understand that you can withdraw from the research activities without repercussions, at any time by the end of the study, and have your data deleted.
4. You are satisfied that the assurances of responsible and strict data governance, given by the **5G-EPICENTRE** project, will be upheld.
5. You understand that your personal data are kept and treated as confidential as far as this research program is concerned.
6. You know and understand that your personal data will be kept in a secure environment and that the data controller, as well as any data processors, will take all the necessary and appropriate measures to protect the security, and in particular the confidentiality and integrity, of personal data, according to data protection legislation and the relevant guidelines.
7. You agree with the publication of the results of this study in anonymous form and with the publication of selected screenshots for the promotion of the study in mass media, and / or scientific publications aimed at informing the public and/or the scientific community.

Next Question

* Consent to participate:

- Agree
- Disagree

< Next

Page 2 of 6

Questions marked with a * are required

Note: please proceed only if Informed consent section has been agreed.

Click **"Next"** to proceed with the questionnaire.

Next Question

* Name of your organisation:

Next Question

* In which sector does your organization belong to?

- Public (e.g., public service with PPDR services' needs)
- Private (e.g., 5G Operator, PPDR services implementer, experimentation facilities providers, etc.)
- Academic
- Research
- Regulatory body
- Other

Next Question

If you selected "Other" in the previous question, please specify:

Next Question

* What is the size of your organization, based on the number of employees?

- 1 - 50
- 51 - 250
- 251 - 1000
- more than 1000

Next Question

* What is the main country of your current workplace?

Next Question

* Which of the following roles fits you best?

- [ ] Business driver
- [ ] Strategical driver
- [ ] Technical driver
- [ ] Domain expert

Next Question

* What is your management level?

- ○ Executive officer (i.e., CEO, CTO, CFO, COO etc.)
- ○ Operating officer (i.e., General manager, plant manager, regional manager, divisional manager etc.)
- ○ Administrative officer (i.e., Office manager, shift supervisor, department manager, foreperson, crew leader, store manager, project leader etc.)
- ○ Professor
- ○ Engineer
- ○ Researcher
- ○ Other

Next Question

If you selected "Other" in the previous question, please specify:

Next Question

* How many years of business experience you have?

- None
- Less than 2 years
- 2 to 5 years
- 5 to 10 years
- 10 to 20 years
- more than 20 years

Next Question

* What is the role of your organisation?

- [ ] Application/service/PPDR solutions provider
- [ ] PPDR service buyer
- [ ] PPDR service user
- [ ] Telecommunications network operator
- [ ] Research/Experimental platform operator
- [ ] PPDR Network Planning (RAN)
- [ ] Standardisation body/regulator
- [ ] Other

Next Question

If you selected "Other" in the previous question, please specify:

< Next

Page 3 of 6

On a scale from 1 to 5, do you consider that PPDR services are limited by current networks/solutions?

(1: totally disagree, 5: totally agree)

- 1
- 2
- 3
- 4
- 5

Next Question

Which KPIs/PPDR service aspects you believe are limited by current networks?

Next Question

On a scale from 1 to 5, do you currently test the PPDR services you develop/procure in operational environments/experimentation facilities, before moving to production?

(1: totally disagree, 5: totally agree)

- 1
- 2
- 3
- 4
- 5

Next Question

Please mention the domain of your PPDR services focus:

- Command & Control
- Firefighting

- ○ Mission critical communications
- ○ Testing PPDR services
- ○ Emergency medical services
- ○ Law enforcement
- ○ Automatic vehicle location
- ○ Refineries
- ○ Surveillance
- ○ Ports / Airports
- ○ Transportation / Highways / Railways
- ○ Large-scale events management
- ○ Other

Next Question

If you selected "Other" in the previous question, please specify:

Next Question

Which benefits do you anticipate by utilizing the 5G-EPICENTRE experimentation facility?

- ☐ Test KPIs
- ☐ Test performance in a near operational environment
- ☐ Test compliance to standards
- ☐ Test performance in extreme conditions (simulated)
- ☐ Collaborate with other providers to create a value chain (e.g., VNF implementers, application implementers, equipment providers, etc.)
- ☐ Provide the ability to the end user to run their tests
- ☐ Provide the ability to the operators to test onboarding/operating of the service

- [ ] Provide the ability to define the business collaboration model between the involved parties
- [ ] Other

Next Question

If you selected "Other" in the previous question, please specify:

Next Question

Choose the functions that such an experimentation facility should provide:?

- [ ] Friendly user interface guiding to perform test cycle processes
- [ ] Service onboarding/parametrisation functionality
- [ ] VNF/NETApps repository and ability to use
- [ ] Other services repository and ability to use
- [ ] Network resources repository and ability to use
- [ ] Slices parameters definition
- [ ] Ability to test in various 5G releases
- [ ] Ability to define test cases and KPIs
- [ ] Traffic generator
- [ ] Visualisation of KPIs measure and automated analysis vs required performance/advanced reporting
- [ ] Standards compliance inspection
- [ ] Security inspection and security breach test
- [ ] Interference inspection
- [ ] Simulate extreme operation conditions (physical and digital)
- [ ] Scaling ability inspection
- [ ] Service setup latency
- [ ] Fault tolerance of a service and self-healing

- [ ] Other

Next Question

If you selected "Other" in the previous question, please specify:

Next Question

Please describe the services that the experimentation facility should provide:
(billing, scheduling, training, etc.)

Next Question

Please describe what the experimentation facility should avoid:

Next Question

Should the experimentation facility support cross-domain experiments?

- Yes
- No
- I don't know

Next Question

What are your expectations regarding network slices' configuration during setup phase?
(e.g., regarding maximum uplink/downlink capacity, prioritisation of flows, security schemes)

- I don't mind, 5G-EPICENTRE experts should perform network slice configuration

- I need to be able to fine tune parts of the network slice by invoking appropriate NETAPPs (e.g., for RAN, Core Network, etc.)
- I need to be able to manage the lifecycle of the involved VNFs and fine tune their operation
- I need to be able to manage the VM hosting the VNFs, as well as manage and fine tune these VNFs
- I don't know

Next Question

How do you envisage the reservation of experimentation facilities' resources for conducting iterations of experiments over time?

- Online tools for resource booking should be provided by each experimentation facility infrastructure
- Online tools for resource booking should be provided by 5G-EPICENTRE portal as a "one-stop-shop" for all experimentation facilities
- Resource booking should be done offline by contacting the individual experimentation facilities' administrators
- I don't know

Next Question

What are your expectations regarding security schemes at run-time?

- I don't mind, 5G-EPICENTRE experts should manage security schemes
- Need to be able to define standards security schemes
- Need to be able to customize security schemes
- I don't know

Next Question

What are your expectations regarding cross-layer privacy framework?

- Don't mind, 5G-EPICENTRE experts should manage cross-layer privacy framework
- Need to be able to configure cross-layer privacy framework
- I don't know

Next Question

Please describe your use case:
(**optional**)

Next Question

Please describe the experimentation facility usage business & remuneration model:
(**optional**)

< Next

Page 4 of 6

Which of the following “must-have” functional requirements you regard as the most important for the platform?

- ☐ The system must allow a user to define experiments
- ☐ The system must allow a user to define experiment-specific KPIs
- ☐ The system must support onboarding of network functions (NFs)
- ☐ The system must support testing of NFs
- ☐ The system must be able to manage and orchestrate NFs on top of virtualised infrastructure
- ☐ The system must support service function chaining of NFs into end-to-end services (NetApps)
- ☐ The system must support interoperability with existing network components from the testbeds’ Network Functions Virtualisation (NFV) environments

Next Question

Which of the following “should-have” functional requirements you regard as the most important for the platform?

- ☐ The system should embrace lightweight virtualisation technologies

- [ ] The system should support provision of execution resources (compute and storage) close to the end users
- [ ] The system should support remote access to the definition and monitoring of experiments
- [ ] The system should provide a proper abstraction of the underlying network technologies
- [ ] The system should expose easy-to-consume APIs toward the experimenter
- [ ] The system should support performance testing in (simulated) near-operational conditions
- [ ] The system should be able to simulate extreme network traffic and mobility conditions
- [ ] The system should support the ability of NFs to scale in and out on demand based on a variety of metrics
- [ ] The system should expose a requirements' catalogue for the underlying network resources
- [ ] The system should provide a user with appropriate network resource inventories and means to means to configure and (re)use them
- [ ] The system should allow experimenters to repeat and re-parameterize experiments
- [ ] The system should provide means to customize network slices for NetApp requirements under eMBB, URLLC and mMTC service types
- [ ] The system should support automated experiment lifecycle management
- [ ] The system should provide means to calculate experiment analytics and metric information
- [ ] The system should provide means to validate KPIs
- [ ] The system should provide means to evaluate Quality of Service (QoS)
- [ ] The system should provide means to assess Quality of Experience (QoE)
- [ ] The system should translate analytics into visualization formats suitable for interpretation by a hu man being
- [ ] The system should support the capability of producing customized reports based on users' needs and preferences
- [ ] The system should provide guidance to the user in order to train them on using it

Next Question

Which of the following "could-have" functional requirements you regard as the most important for the platform?

- [ ] The system could provide means to calculate security analytics

- [ ] The system could support cross-domain/cross-testbed experiments
- [ ] The system could enable the calibration of individual testbed components from a singular control point
- [ ] The system could support role-based access control (RBAC) policies

Next Question

Which of the following “must-have” non-functional requirements you regard as the most important for the platform?

- [ ] The system must be secure
- [ ] The system must comply to relevant standards
- [ ] The system must be privacy-compliant
- [ ] The system must be performant/responsive
- [ ] The system must be reliable

Next Question

Which of the following “should-have” non-functional requirements you regard as the most important for the platform?

- [ ] The system should be user-friendly
- [ ] The system should be cloud-native

Next Question

Which of the following “could-have” non-functional requirements you regard as the most important for the platform?

- [ ] The system could be adaptable
- [ ] The system could be documented

< Next

Page 5 of 6

Questions marked with a * are required

* Did you participate in the 1st round of the survey?

- Yes
- No

* Would you like us to contact you to further discuss your view on the experimentation facility?

- Yes
- No

If you replied *YES*, please kindly fill the following information:

First Name

Last Name

Title

Organization

Office phone number

Mobile phone number

Email Address

Skype

Other

Preferred interview method:
(e.g., in person, over the phone, online conference platform, messaging application, etc.)

< Done

Page 6 of 6

Annex II: Towards cloud-native enhancement of the federated platforms

The contents of this Annex correspond to the D1.3 Section 3.4 "Towards cloud-native enhancement of the federated platforms".

NFV-MANO RA

In the early 2010s, Network Functions Virtualization (NFV) essentially addressed time and cost-saving provisions for the deployment of new network services, by means of decoupling software from purpose-built hardware (where one piece of hardware performed one software operation) to a single piece of hardware running several virtual machines (VMs) each performing a dedicated function known as a virtual network function (VNF). This paradigm shift has helped propel the development of 5G technology.

In 2014, the European Telecommunications Standards Institute (ETSI)-established Industry Specification Group for NFV (ISG NFV) defined the NFV RA framework, which identified the standards-compliant functional components and relationships between those components. In addition, the framework specified the base information elements involved in the NFV management and orchestration (MANO) system. These are summarized in Figure 42.

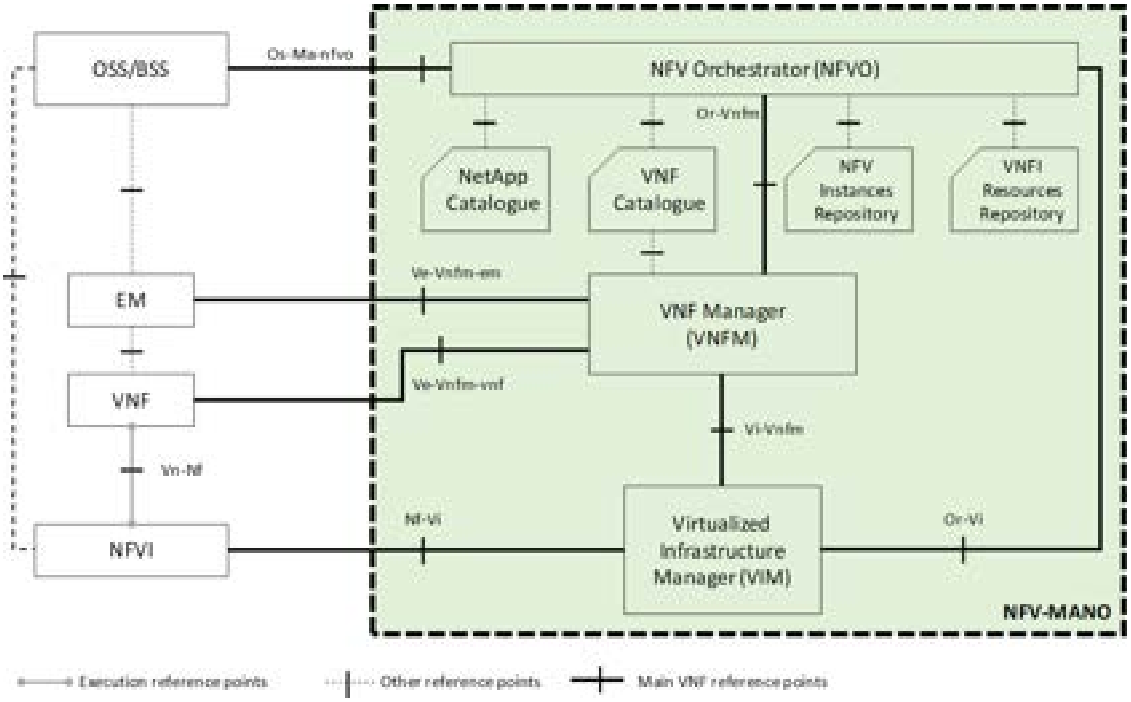


Figure 42: NFV-MANO reference architectural framework (Adapted from [A1]).

As can be seen in the ETSI NFV RA diagram, three subsystems are defined inside the NFV-MANO component:

- The **Virtualized Infrastructure Manager(s) (VIMs)**, which control and manage the interaction of VNFs with the physical resources (computing, storage, network) under their supervision.
- The **NFV Orchestrator (NFVO)**, which is responsible for the orchestration and management of the NFVI and software resources, along with the onboarding and managing of the life-cycle of NetApps.
- The **VNF Manager(s) (VNFMs)**, which is assigned lifecycle management duties of VNFs managed by the NFVO, coordinating the VIM for to facilitate the instantiation, update, scaling, querying and termination of the VNFs in its care.

- The **NetApp Catalogue** (specified in [A1] as the 'NS Catalogue') is a repository of onboarded NetApps, which facilitates creation and management of all NetApp related templates (*e.g.*, the NetApp descriptor).
- The **VNF Catalogue** is a repository of onboarded VNFs, which facilitates creation and management of all NetApp related templates (*e.g.*, the VNF descriptor).
- The **NFV Instances repository**, which contains information on VNF and NetApp instances stored as records, updated during each instance lifecycle.
- The **NFVI Resources repository**, which stores information on the NFVI resources that are important for the reservation, allocation and monitoring.

The NFV RA further identifies the main reference points (RPs) that provide the interfaces facilitating exchanges of data between the aforementioned subsystems. The main RPs inside the NFV-MANO block are:

- The NFVO – VNFM RP (**Or-Vnfm**), which carries resource requests by the VNFM to the NFVO (*e.g.*, for the reservation of resources); configuration data by the NFVO to the VNFM for proper configuration of VNFs in the NetApp chain; and VNF state information queried by the VNFM to carry out lifecycle management of VNFs.
- The VIM – VNFM RP (**Vi-Vnfm**), which carries resource allocation requests by the VNFM to the VIM; along with virtual hardware state information and resource configuration to facilitate interaction of VNFs with the physical resources.
- The NFVO – VIM RP (**Or-Vi**), which carries resource reservation requests made by the NFVO as well as virtual hardware state information and resource configuration data.

In addition, inbound/outbound RPs include the following:

- A RP between the Operations and Business Support System (OSS/BSS) and the NFV-MANO block, and more particularly, the NFVO (**Os-Ma-nfvo**), which is used for requests for NetApp/VNF lifecycle management; NFV state information; policy management and data analytics exchanges; NFV related accounting and usage records; and NFVI capacity and inventory information.
- A RP between the Element Management (EM) and the VNFM (**Ve-Vnfm-em**). The EM is responsible for typical management functionality (configuration, fault management, accounting, performance measurement, security) for one or several VNFs. Hence, this RP is used for forwarding requests related to VNF lifecycle management (instantiation, run-time information querying, scaling, termination); and bidirectional exchange of configuration and events information between the EM and VNFM regarding the VNF. In case the EM is not aware of virtualisation, a direct RP between the VNF and VNFM (**Ve-Vnfm-vnf**) fulfils the above exchanges.
- A RP between the NFVI and VIM (**Nf-Vi**), which is used for assigning virtualized resources to address resource allocation requests by the VIM; the exchange of virtualised resources and hardware resources configuration and state information.

The NFV-MANO RA presented above specifies a high-level architectural reference frame along with implementation guidelines and interoperability via open interfaces, and is accommodated by the underlying 5G-EPICENTRE experimentation infrastructure platforms. The aim of 5G-EPICENTRE is to maintain those elements that can continue to serve their purpose when applying the transformation to cloud native Network Functions Virtualization (NFV), enabling augmentations and their replication to be introduced wherever necessary to accommodate the shift to cloud native solutions.

Cloud-native NFV-MANO

The ETSI NFV-MANO Group Specification on NFV-MANO describes a common frame of reference for the implementation of MANO capabilities on hypervisor-based virtualization environments, supporting deployment of VNFs as VMs. The majority of practical implementations of this architecture are built mostly on hypervisor-based virtualization environments, where the VIM is mostly based on VM orchestration tools, such as OpenStack[19] and VMware[20]. Evolution to cloud-native will involve deployment of network functions in the cloud in the form of microservices containers, introducing added complexity but also several advantages to the ETSI-compliant NFV-MANO systems [A2]. Hence, in the ETSI GR NFV-IFA 029 [A3], the ISG NFV identified additional components that the NFV-MANO should support for MANO of container-based VNFs:

- The **Container Infrastructure Service (CIS)**, which is responsible for delivering the runtime environment for one or more container virtualisation technologies that can be ran on top of both bare metal and hypervisor-based virtualisation environments.
- The **Container Infrastructure Service Management (CISM)**, which manages the deployment, monitoring, and lifecycle of containerized workloads and infrastructure resources containing computing, storage, and network resources.
- The **Container Image Registry (CIR)**, which stores container images and exposes them to other functions.

The ETSI ISG NFV presents various use cases for the utilization of containers, specifying that *"the CISM does not mean any other functional block than the existing NFV-MANO functional blocks, but a logical entity with functionality to be integrated in NFV-MANO architecture"* [A3]. Several options for the enhancement of the NFV-MANO architecture with CISM functionality towards utilization of containers have been envisioned. Each enhancement has its pros and cons, leading to a comparison of architectural options but no concrete conclusion on CISM to NFV-MANO RA mapping [A4]. Particular focus is placed on derivative architectures of the NFV-MANO RA block to support backward compatibility, as well as mapping to existing de-facto standards. Within 5G-EPICENTRE, it is proposed that the cloud native infrastructure will be deployed as an overlay on top of the NFVI. The NFV-MANO underlay then acts as the resource orchestrator, allocating computational resources at Kubernetes clusters.

Kubernetes (K8s) is the de-facto platform for automating container deployment and managing container lifecycle. It is responsible for scheduling and running containerized applications over one or more containers, while it is responsible for scheduling and running containerized applications over one or more containers, while simultaneously automating container operational tasks, such as creating, starting, organizing, monitoring and destroying containers. K8s internal high-level architecture is shown in Figure 43. Containers are grouped into a K8s *pod*, providing shared storage and network resources. Pods are assigned to run on *worker nodes*, which are sets of IT resources (*e.g.*, physical or virtual machines). A set of nodes running containerized applications is called a K8s *cluster*. A master node is assigned to be responsible for handling control of events in the K8s cluster.

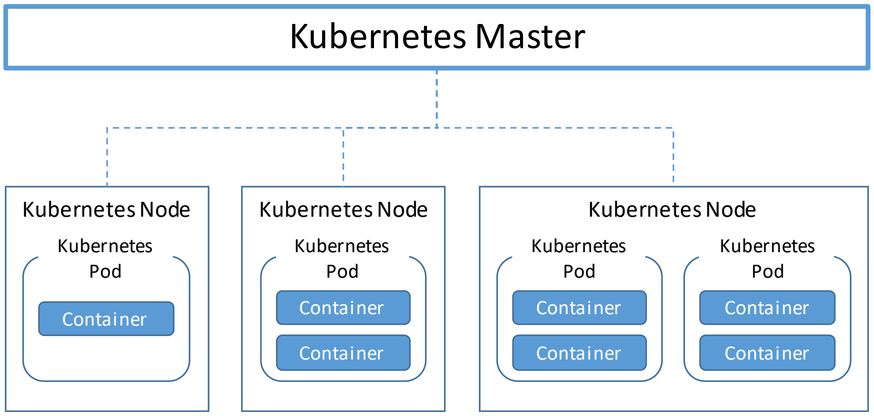


Figure 43: Kubernetes high level master node system architecture.

[19] https://www.openstack.org/
[20] https://www.vmware.com/

Because of this extensible architecture, K8s is a MANO-compliant platform can be mapped to the functions of the ETSI NFV-MANO architecture into a single, yet extensible solution. According to the NFV Release 4 FEAT17 [A4], the de-facto standard solution analogy for the above terms correspond to K8s nodes and services, as specified in the Table below (Table 11).

Table 11: Mapping of [A3] terms to de-facto standard solutions.

Component/Functionality	Analogy
CIS	K8s services exposing the Container Runtime Interface (CRI), Container Network Interface (CNI) and Container Storage Interface (CSI).
CIS instance	K8s worker node.
CISM	K8s master node.

Extensibility in K8s is ensured via plugins, which enable a K8s cluster to adapt to the needs of any work environment. For instance, a Container Network Interface (CNI) plugin is used to setup network connectivity between pods on different nodes, allowing them to communicate with one another.

Within 5G-EPICENTRE the goal will be to smoothly evolve the existing NFV-MANO architectures in each testbed into a K8s-based orchestration system for automating VI and VNF/CNF deployment, scaling and management. This will be achieved by extending K8s with the appropriate plugins for extending its functionality. An important such extension toward the cloud-native transformation, adaptation, reconfiguration and evaluation of the testbeds should facilitate support for MANO of both container-based and VM-based VNFs. This would facilitate backward compatibility, making it easier to map the NFV-MANO functional model augmented with CISM functionality to K8s. The latter supports such mixed VM-container workloads through the KubeVirt VM management add-on[21], which essentially enables the launch of both containers and VMs on the same cluster, or even the same node, using the same networks and same storage infrastructure (Figure 44). Recent evidence by the 5G-PPP supports this technology as a potential disruptor in the NFV-MANO architecture [A5].

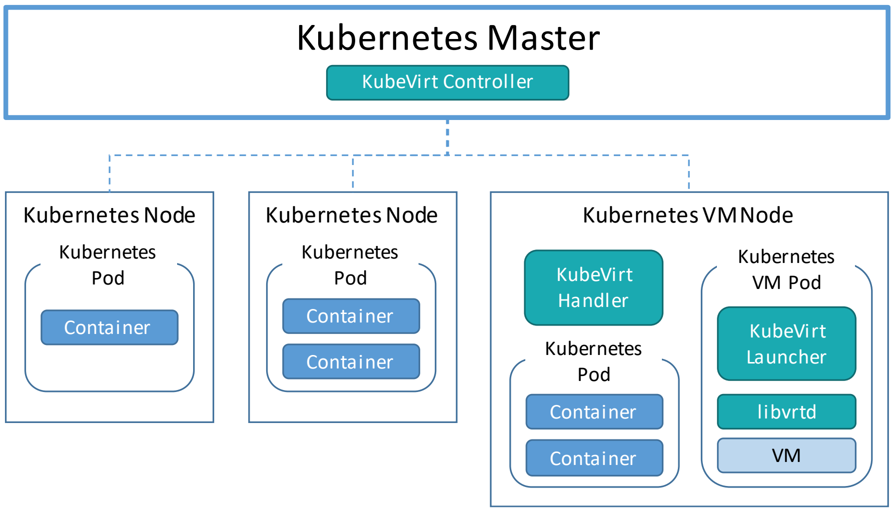


Figure 44: KubeVirt components for orchestrating/managing VM-based workloads alongside container-based workloads.

[21] http://kubevirt.io/

Taking the above into account, Figure 45 describes an early common frame of reference for the augmentation of the current NFV-MANO block by adding support to manage both network functions and virtual infrastructure on Kubernetes, essentially instigating embedding CISM functionality inside both/either the VIM and the VNFM. An updated RP between the between the CNF and VNFM (**Ve-Vnfm-cnf**) is expected to fulfil the same exchanges as Ve-Vnfm-vnf (see previous Section), albeit utilizing K8s and KubeVirt add-on for management and control of both CNFs and VM-based VNFs (running inside regular pods managed by K8s alongside container pods) respectively.

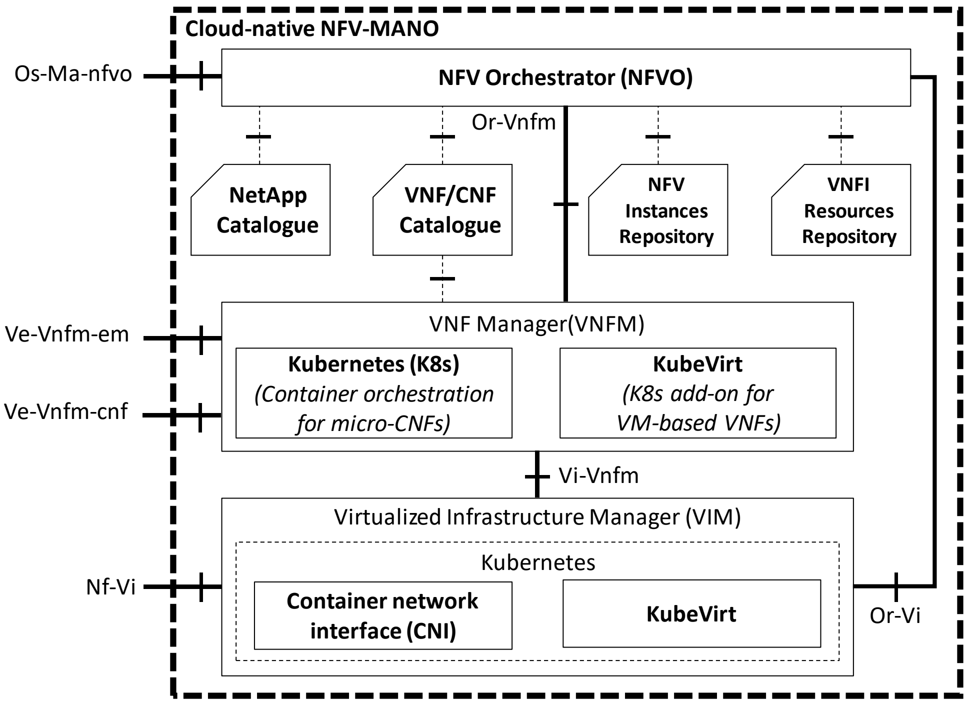


Figure 45: CISM functionality embedded in the ETSI NFV-MANO RA via Kubernetes and add-ons (KubeVirt, CNI). The figure focuses on the portion of NFV-MANO architecture relevant for the architectural impacts. Figure adapted from [A1].

Cloud-native VNF architecture

The ETSI GS NFV-SWA 001 [A6] describes a high-level internal architecture of a VNF. In principle, a VNF consists of one or more VNF Components (VNFCs), each occupying a single VM and providing appropriate links (by means of the SWA-2 interfaces) to the other VNFCs so as to deliver the desired functionality of the VNF. The VNF block further exposes well-defined interfaces toward the NFVI (Vn-Nf, providing the VNF with access to a slice of the NFVI resources), the VNFM (Ve-Vnfm-vnf, providing the VNF with lifecycle management by the VNFM) and EM (SWA-4, enabling the EM to communicate with the VNF). VNFs can be linked together by means of SWA-1 interfaces to form a network service, service function chain or NetApp. This RA is iterated in Figure 46.

This Section aims at providing a high-level overview of how the shift to cloud-native or containerized NFs (CNFs) will be accommodated. In addition, an overview of hosting CNFs alongside legacy VNFs will be provided to facilitate developments in the context of WP2 through WP4.

Micro-VNF/CNF architectural enhancements

As specified in the previous Section, 5G-EPICENTRE is based on the augmentation of the current NFV-MANO block to which it will integrate support to manage NFs and VI using a container engine (Kubernetes) enhanced with the appropriate add-ons to accommodate VM-based VNF implementations dictated by the Figure 46 specification. This will impact the NFV-MANO architecture specification [A1] in the following manners:

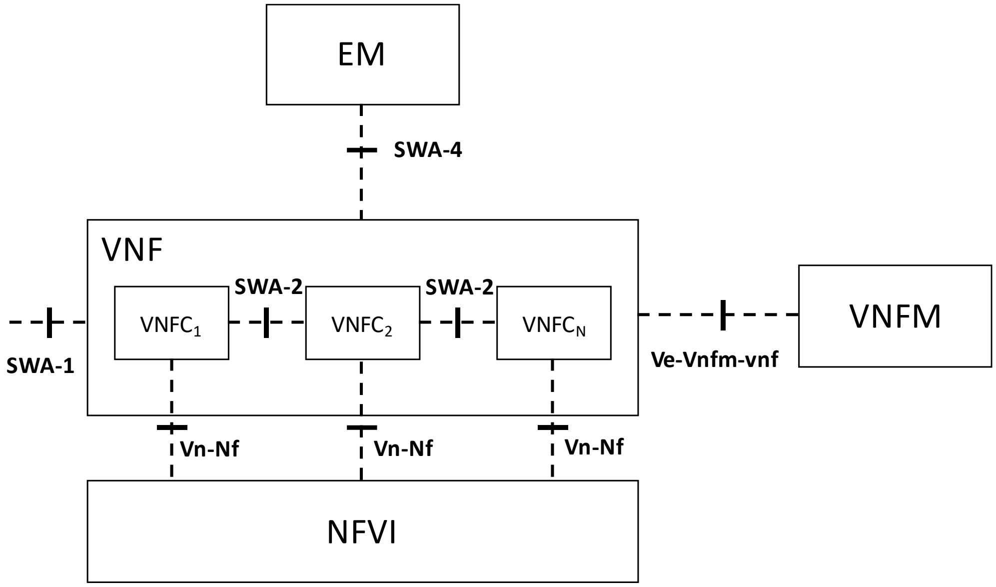


Figure 46: Internal functional architecture of VNF, adapted from [A6].

- It will require infrastructure for the network services to be deployed as containers in a hybrid cloud environment (both public and private clouds), thus accommodating a Cloud-native NFVI (CNFVI). It will contain the hardware needed for compute, storage and networking along with a containerization layer on top of which CNF containers can be spun up [A2].
- It will require VNFCs to be realised in a microservices-based approach, foregoing the 1:1 correspondence between VNFC-VM and yielding instead a one-to-one microservice to container analogy to form a CNF Component (CNFC).

These impacts, along with the considerations for cloud-native NFV-MANO elaborated in the previous Section are highlighted in Figure 47. Internal implementation of microservices-based VNFs is a responsibility assigned to the function developer, and consists of identifying the CNFCs from a microservices perspective (either by developing NFs that correspond to generic functionality that a wide range of CNFs should support , or decomposing existing VNFs into smaller units [A7]), and specifying well-defined interfaces for facilitating communication among CNFs by means of the SWA-2 interface (Figure 47).

Reference implementation NFV ecosystem in Kubernetes

In order to accommodate the hybrid deployment envisioned within 5G-EPICENTRE, VNF/CNF orchestration and lifecycle management should accommodate both VM-based monolithic VNFs as well as CNFs. The addition of the KubeVirt add-on to the K8s orchestration engine enables treatment of such VMs as K8s-managed resources in a similar way to container resource management across the same K8s cluster, as shown in Figure 44.

Utilizing this information, a reference implementation of the functional augmented NFV-MANO RA depicted in Figure 47 can be seen in Figure 48. CNFs are broken down into CNFCs each hosted in a container on a K8s Pod assigned to a worker node. Similarly, legacy VNFs utilize K8s VM pods running the *virt-launcher* and *virt-handler* KubeVirt components along with an instance of *libvirtd* to manage the lifecycle of the VM process. The K8s master, as the access point responsible for managing scheduling and deployment of the CNF/VNF containers is assigned to the VNFM block, storing the state and configuration of the entire cluster. The VNFM further incorporates the *virt-controller*, which is responsible for creating and scheduling the VM pod.

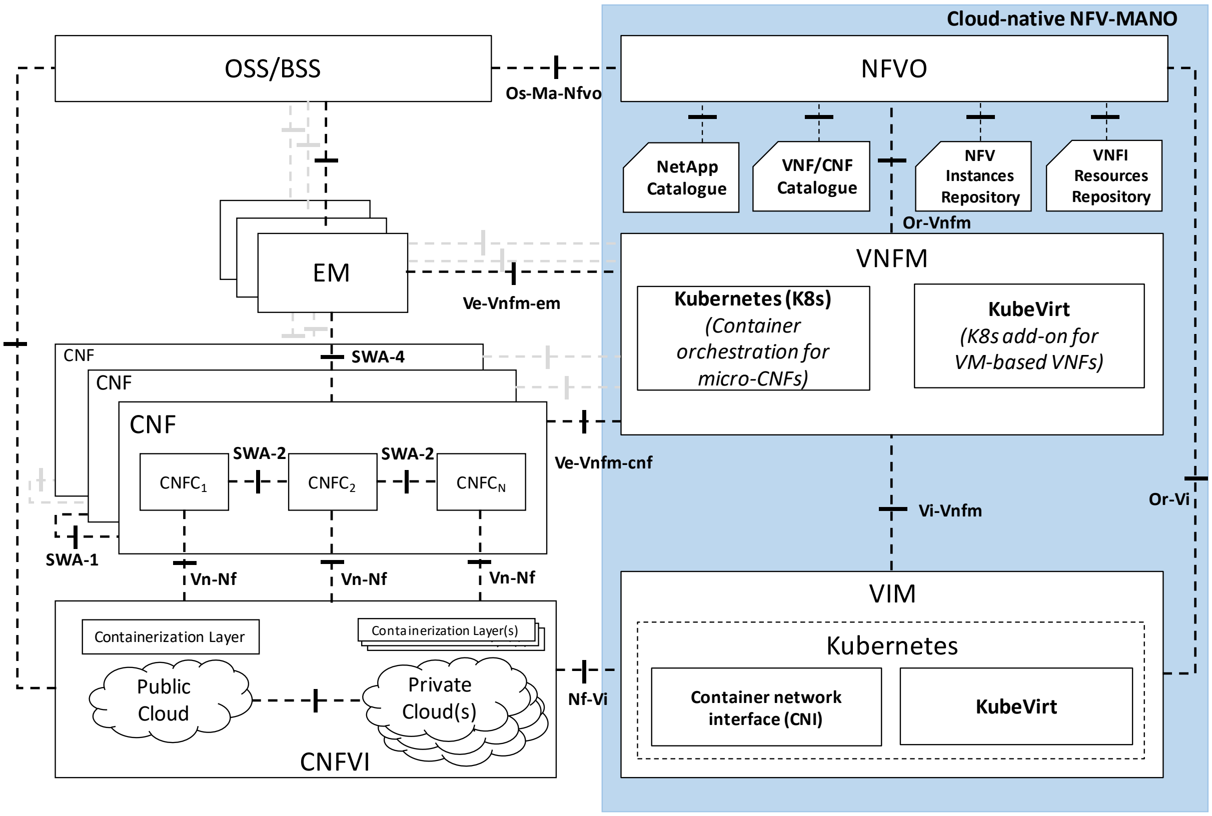


Figure 47: Cloud-native implications for NFV-MANO RA.

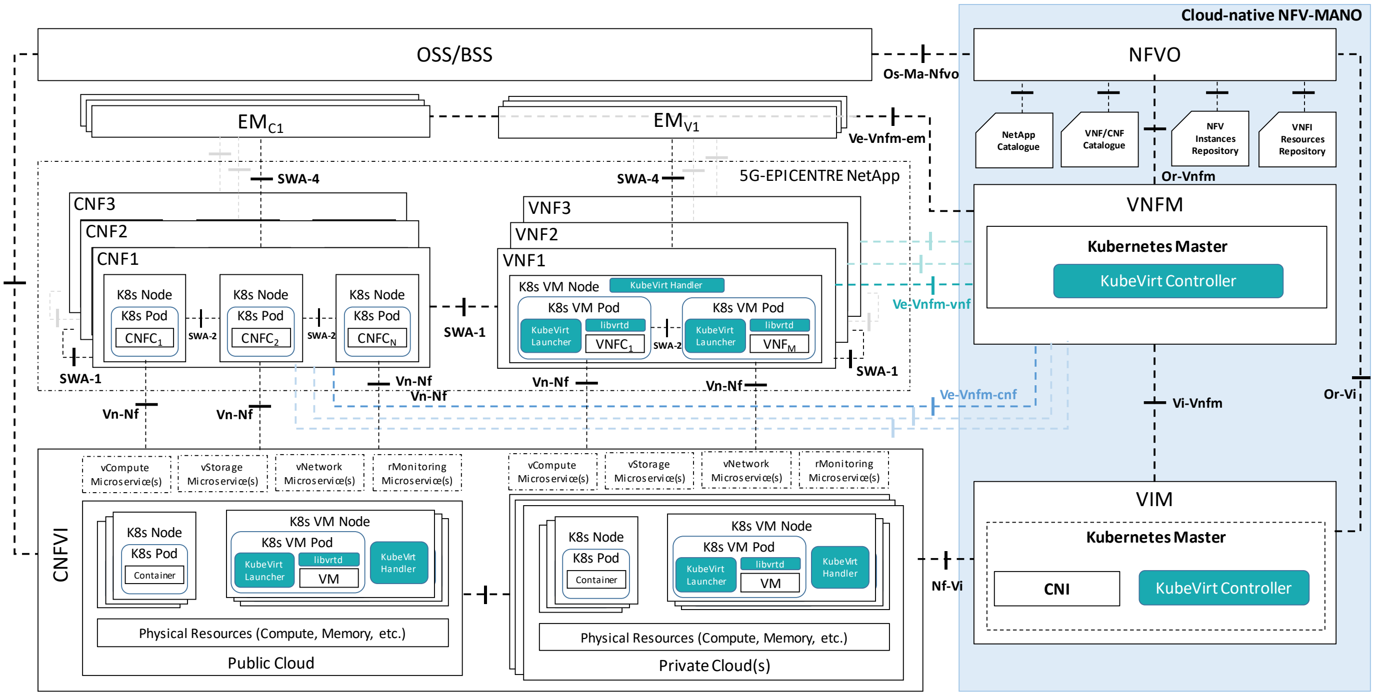


Figure 48: Reference implementation architecture proposal for cloud-native NFV-MANO with K8s and KubeVirt add-on.

On the CNFVI side, a container cloud platform is deployed, responsible for management and capability virtualization of the underlying (public/private cloud) hardware resources, exposing resource management and software microservice support capabilities for the upper layer. The VIM, centred around K8s and KubeVirt add-on, provides the necessary container resource management for managing compute, network, storage and resource utilization in support of the service-based architecture evolution of the upper-layer NetApps and NFs.

References

[A1] ETSI Industry Specification Group (ISG) Network Functions Virtualisation (NFV). (2014). ETSI GS NFV-MAN 001 V1.1.1 (2014-12) - Network Functions Virtualisation (NFV); Management and Orchestration. https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf

[A2] Chen, Y. L., & Bernstein, A. (2017). Bridging the gap between ETSI-NFV and cloud native architecture. In *Proc. SCTE/ISBE Fall Tech. Forum* (pp. 1-27). https://www.nctatechnicalpapers.com/Paper/2017/2017-bridging-the-gap-between-etsi-nfv-and-cloud-native-architecture

[A3] ETSI Industry Specification Group (ISG) Network Functions Vitualisation (NFV). (2019). *ETSI GR NFV-IFA 029 V3.3.1 (2019-11) - Network Functions Virtualisation (NFV) Release 3; Architecture; Report on the Enhancements of the NFV architecture towards "Cloud-native" and "PaaS"*. https://www.etsi.org/deliver/etsi_gr/NFV-IFA/001_099/029/03.03.01_60/gr_NFV-IFA029v030301p.pdf

[A4] ETSI NFV IFA & SOL WG (2020, March 9). *NFV Release 4 FEAT17 – CNF management concepts*. https://nfvwiki.etsi.org/images/NFVIFA%2820%29000163_NFV_release_4_FEAT17_CNF_management_concepts.pdf

[A5] 5G-PPP Technology Board WG & 5G-IA's Trials WG (2021). *Edge Computing for 5G Networks* [White paper]. https://zenodo.org/record/3698117#.YKu4xagzZaQ

[A6] ETSI Industry Specification Group (ISG) Network Functions Virtualisation (NFV). (2014*). ETSI GS NFV-SWA 001 V1.1.1 (2014-12) - Network Functions Virtualisation (NFV); Virtual Network Functions Architecture*. https://www.etsi.org/deliver/etsi_gs/NFV-SWA/001_099/001/01.01.01_60/gs_NFV-SWA001v010101p.pdf

[A7] Chowdhury, S. R., Salahuddin, M. A., Limam, N., & Boutaba, R. (2019). Re-architecting NFV ecosystem with microservices: State of the art and research challenges. *IEEE Network, 33*(3), 168-176. https://doi.org/10.1109/MNET.2019.1800082

Annex III: Bottom-up component refinement template

This Annex summarises the component refinement template booklet shared with technical partners toward specification of their components' functional descriptions and APIs.

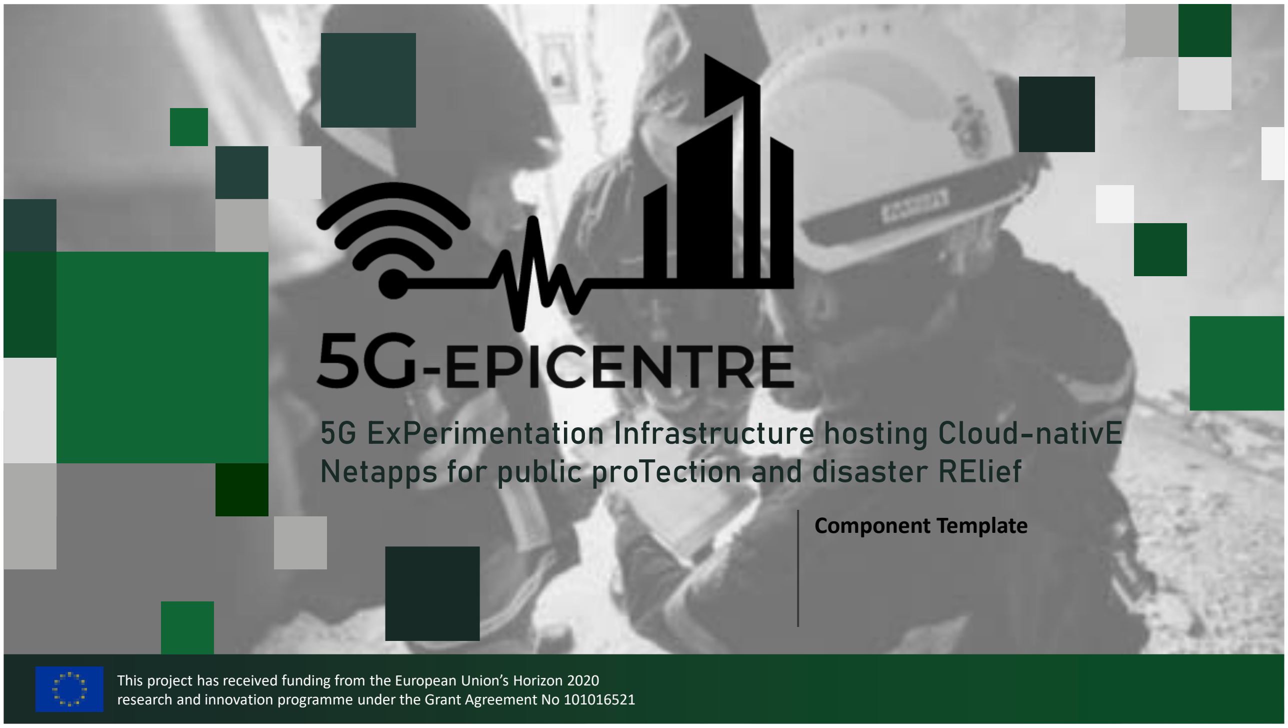


Figure 49: Architecture refinement – Component template distributed to partners (Cover page).

Figure 50: Architecture refinement – Component template distributed to partners (page 2).

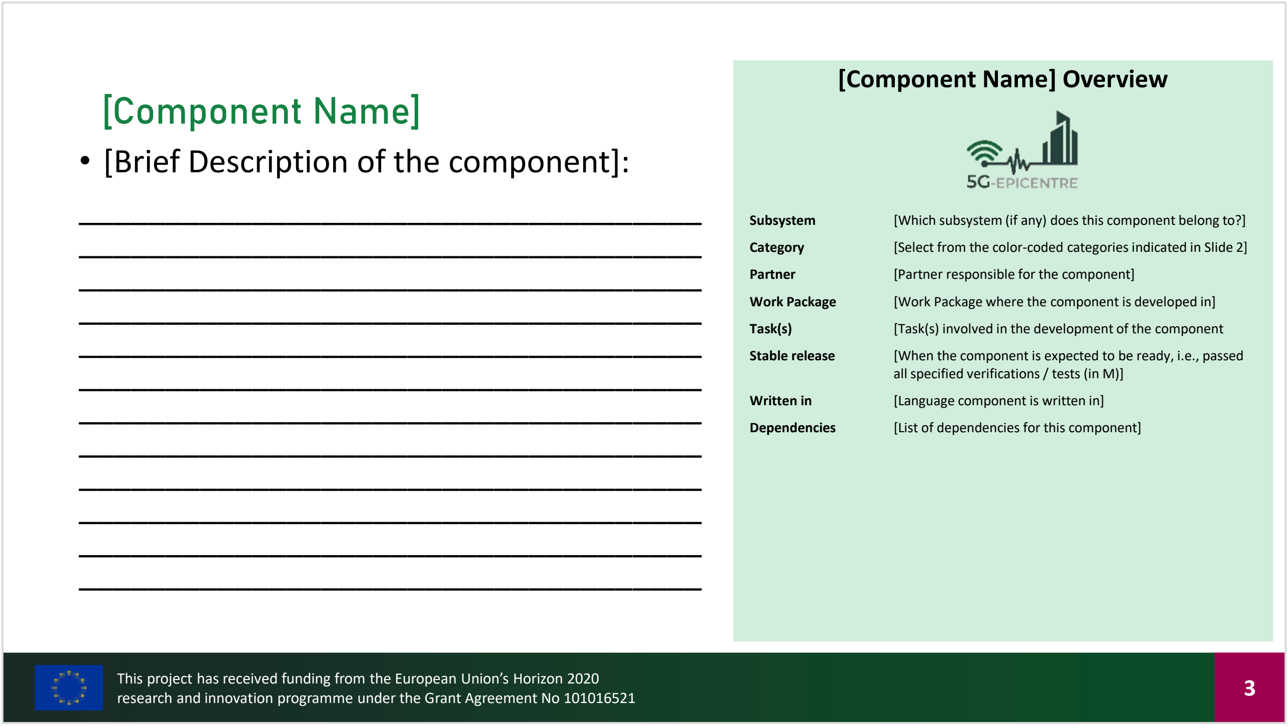


Figure 51: Architecture refinement – Component template distributed to partners (page 3).

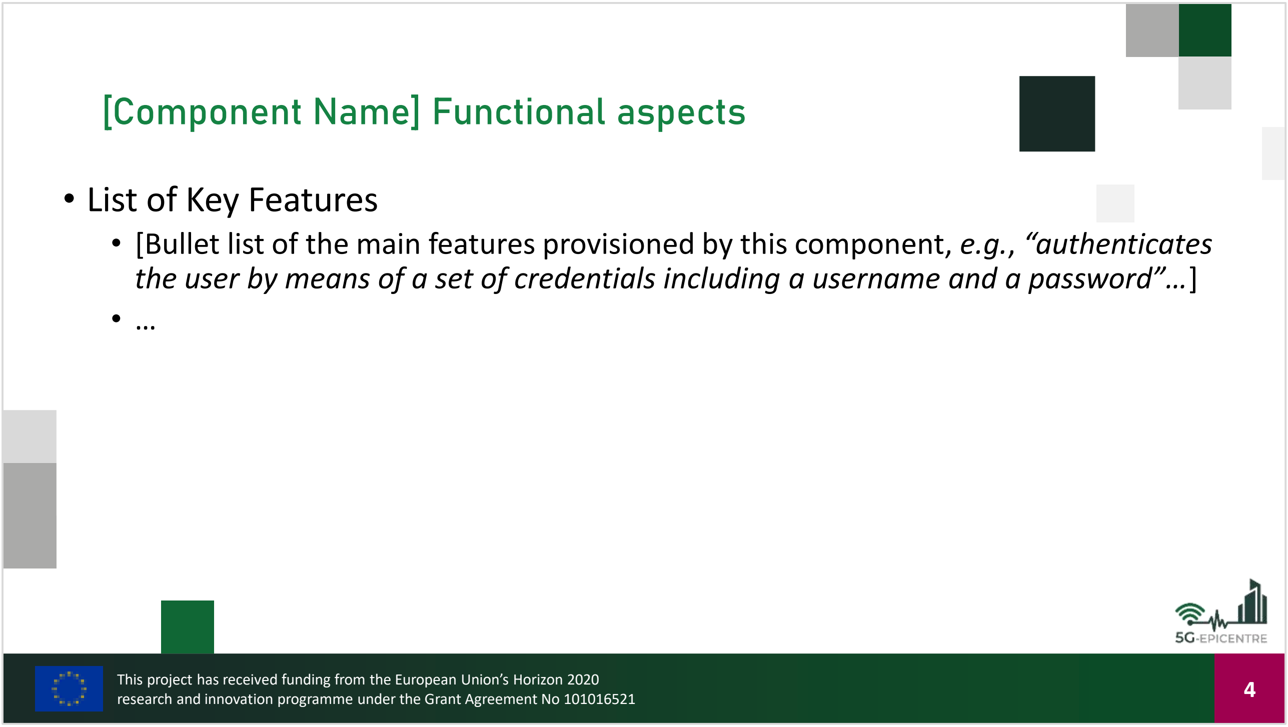


Figure 52: Architecture refinement – Component template distributed to partners (page 4).

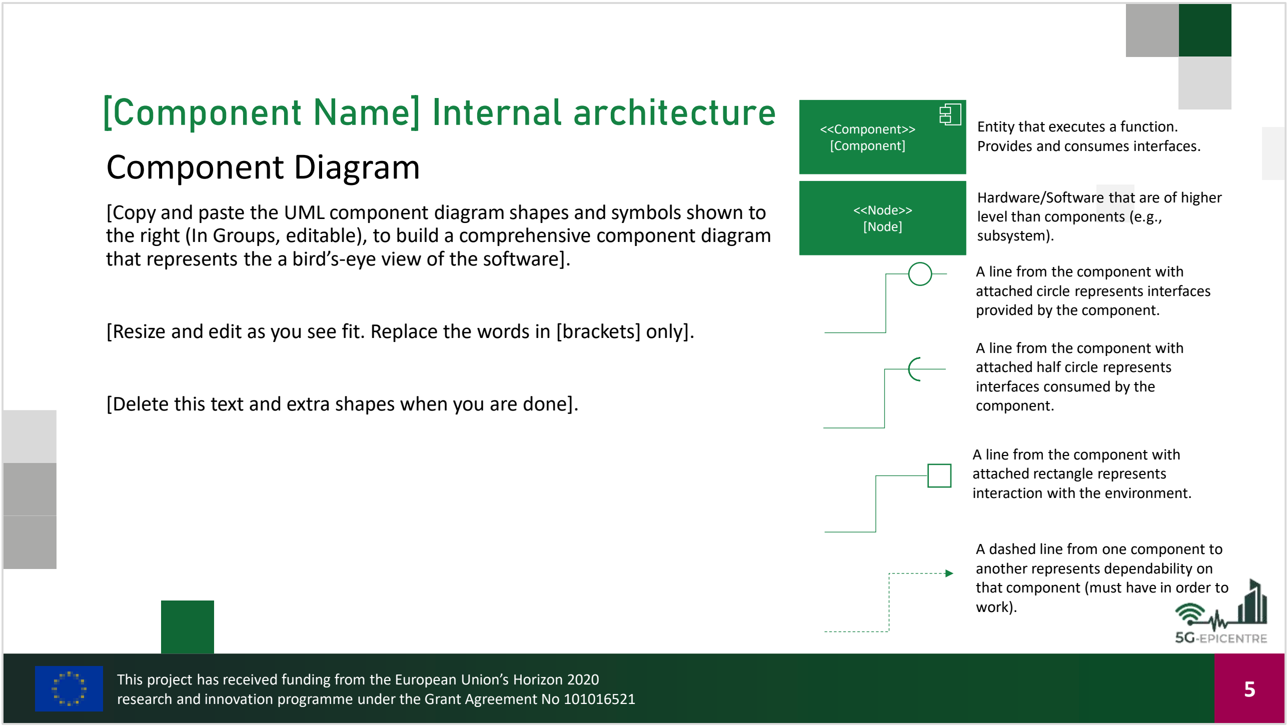


Figure 53: Architecture refinement – Component template distributed to partners (page 5).

Figure 54: Architecture refinement – Component template distributed to partners (page 6).

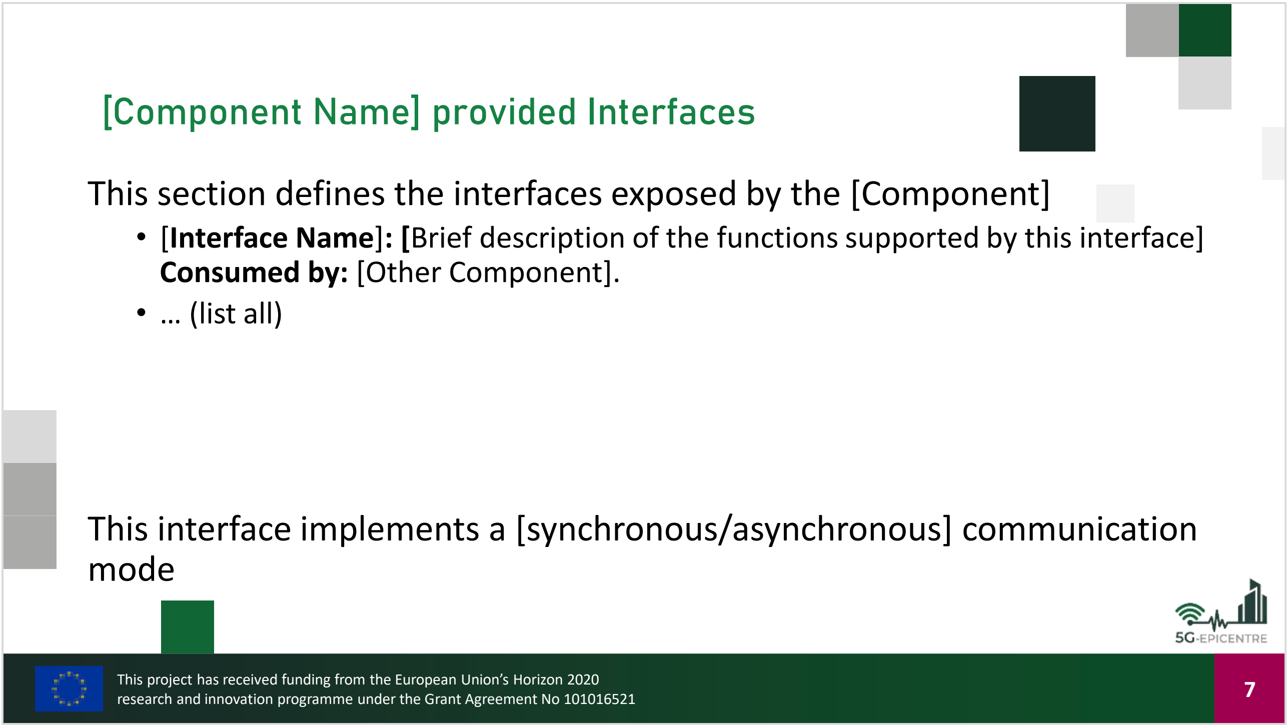


Figure 55: Architecture refinement – Component template distributed to partners (page 7).

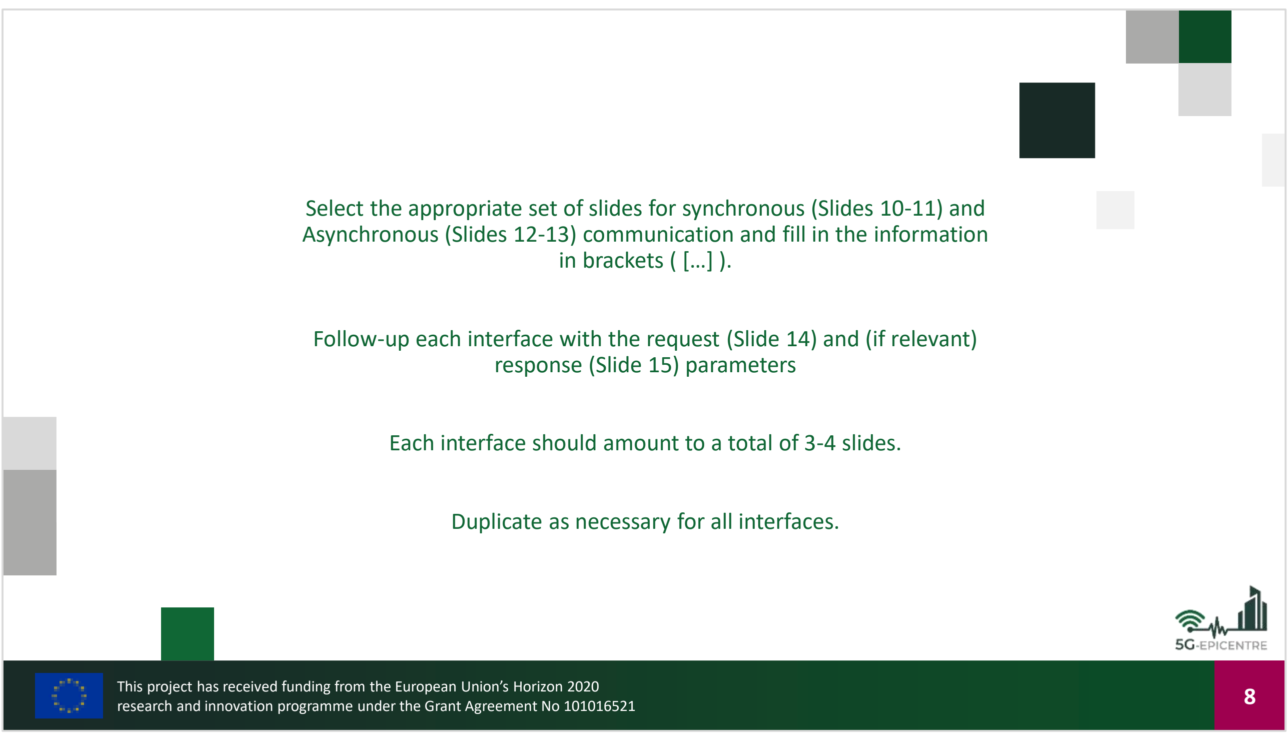


Figure 56: Architecture refinement – Component template distributed to partners (page 8).

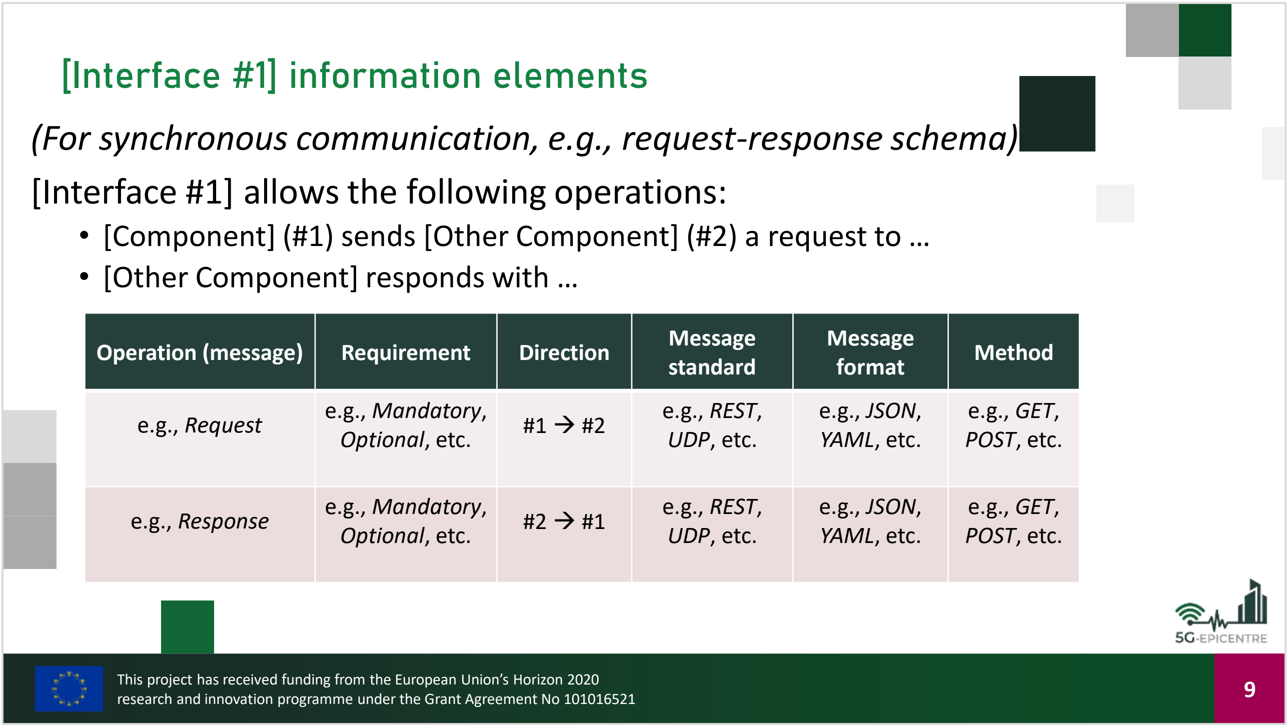


Figure 57: Architecture refinement – Component template distributed to partners (page 9).

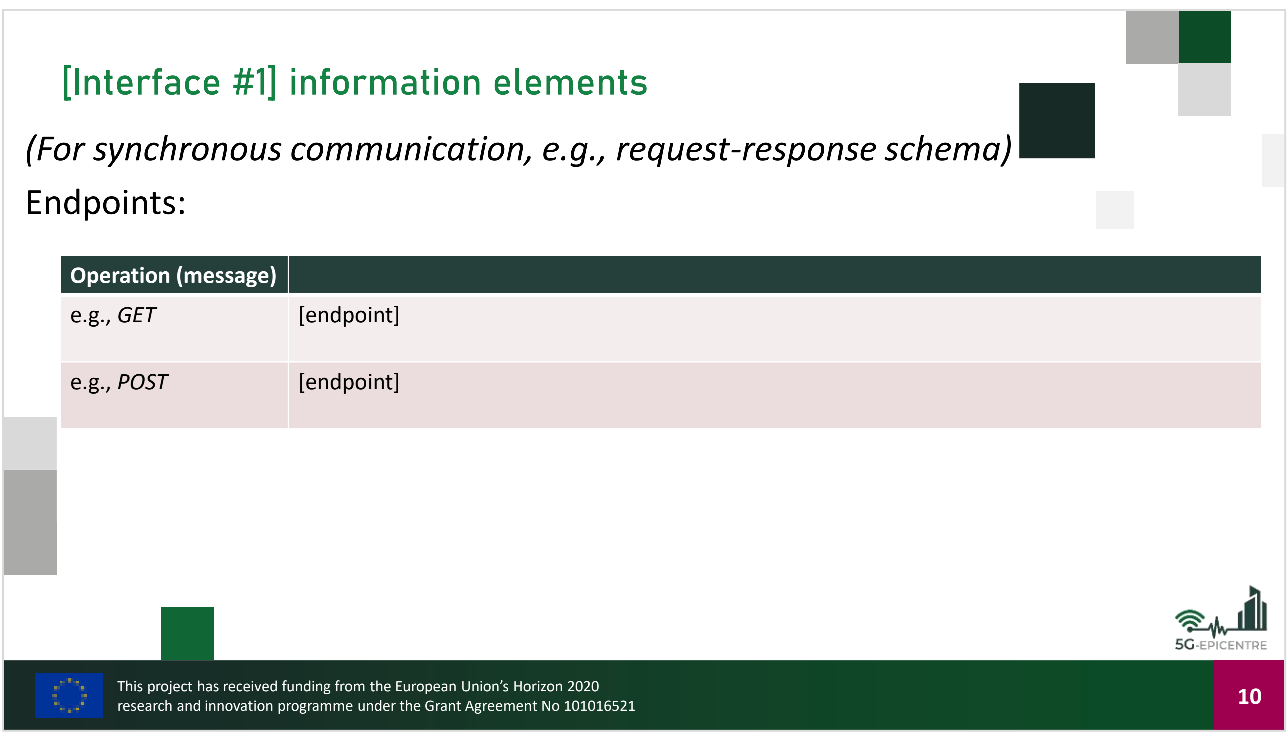


Figure 58: Architecture refinement – Component template distributed to partners (page 10).

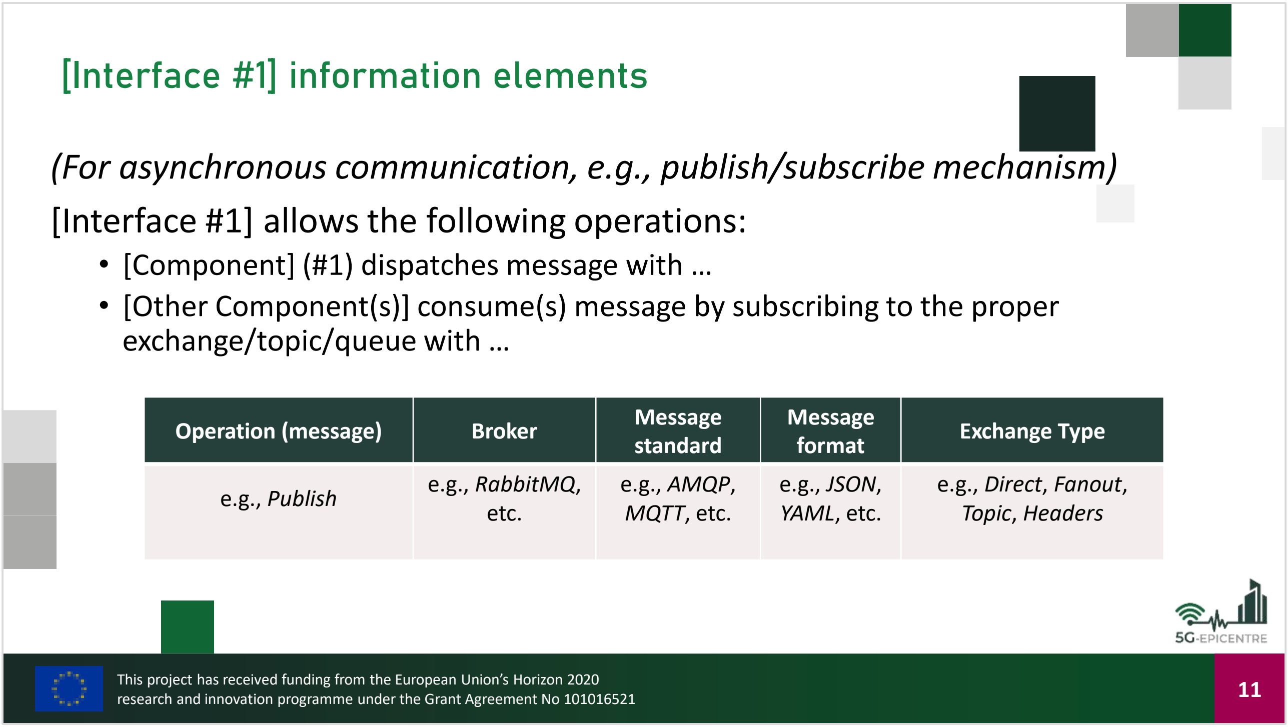


Figure 59: Architecture refinement – Component template distributed to partners (page 11).

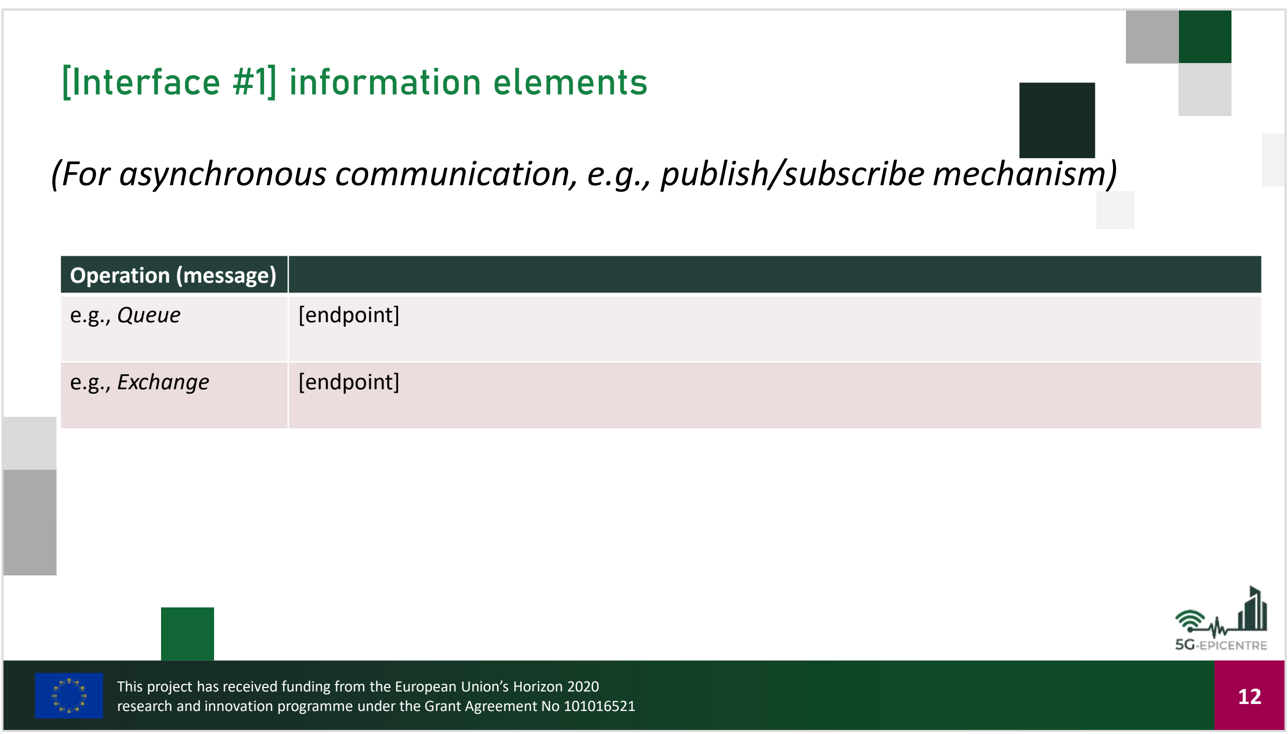
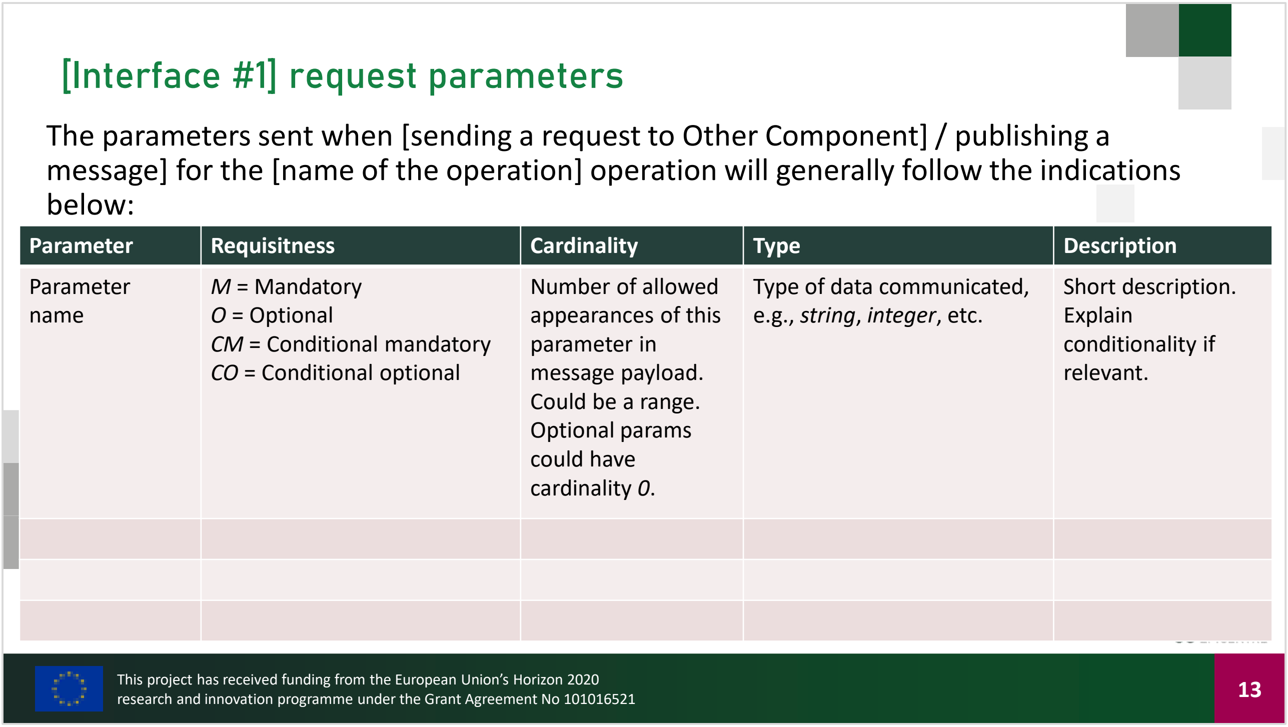


Figure 60: Architecture refinement – Component template distributed to partners (page 12).



[Interface #1] request parameters

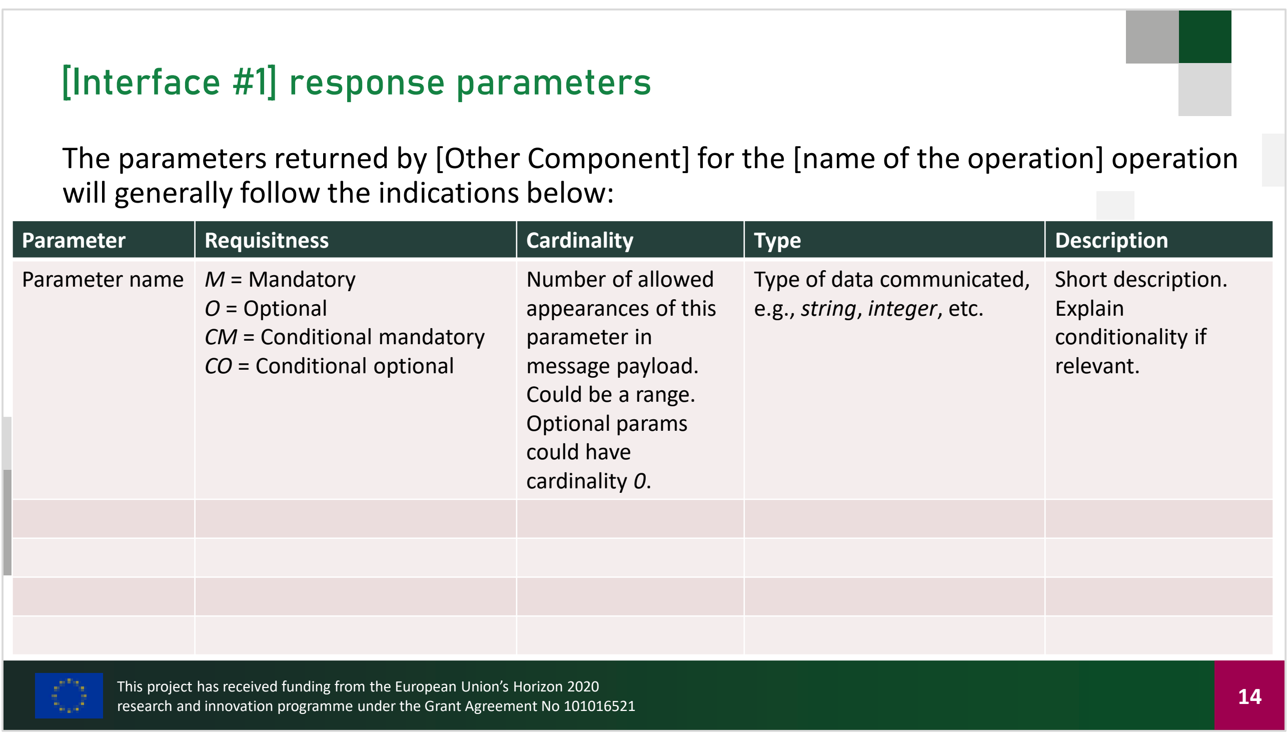
The parameters sent when [sending a request to Other Component] / publishing a message] for the [name of the operation] operation will generally follow the indications below:

Parameter	Requisitness	Cardinality	Type	Description
Parameter name	*M* = Mandatory *O* = Optional *CM* = Conditional mandatory *CO* = Conditional optional	Number of allowed appearances of this parameter in message payload. Could be a range. Optional params could have cardinality *0*.	Type of data communicated, e.g., *string*, *integer*, etc.	Short description. Explain conditionality if relevant.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No 101016521

13

Figure 61: Architecture refinement – Component template distributed to partners (page 13).



[Interface #1] response parameters

The parameters returned by [Other Component] for the [name of the operation] operation will generally follow the indications below:

Parameter	Requisitness	Cardinality	Type	Description
Parameter name	*M* = Mandatory *O* = Optional *CM* = Conditional mandatory *CO* = Conditional optional	Number of allowed appearances of this parameter in message payload. Could be a range. Optional params could have cardinality *0*.	Type of data communicated, e.g., *string*, *integer*, etc.	Short description. Explain conditionality if relevant.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No 101016521

14

Figure 62: Architecture refinement – Component template distributed to partners (page 14).

Figure 63: Architecture refinement – Component template distributed to partners (page 15).

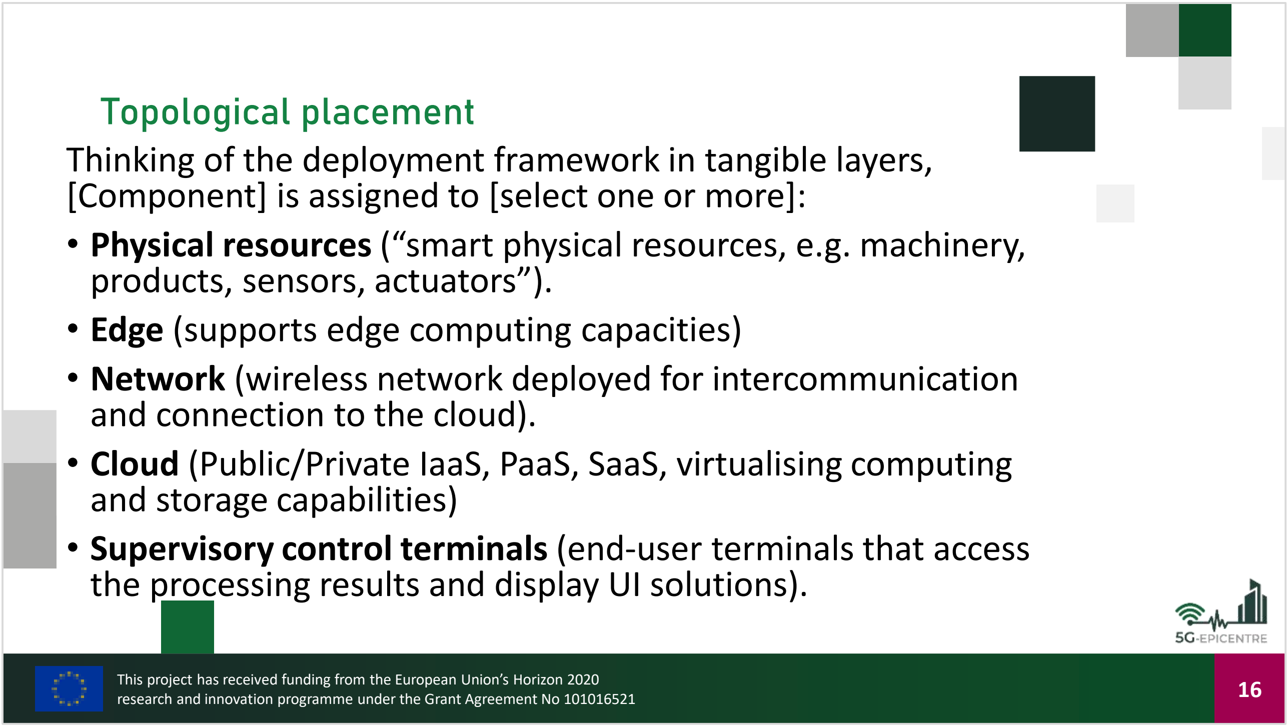


Figure 64: Architecture refinement – Component template distributed to partners (page 16).

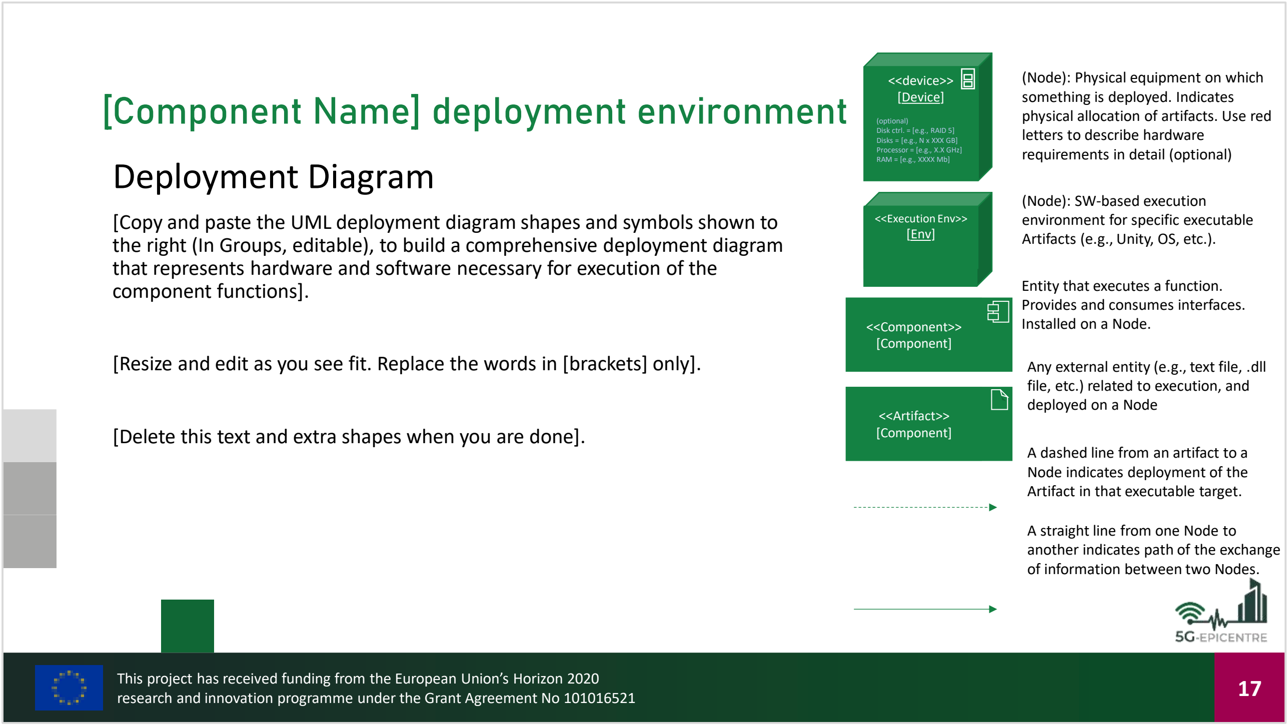


Figure 65: Architecture refinement – Component template distributed to partners (page 17).

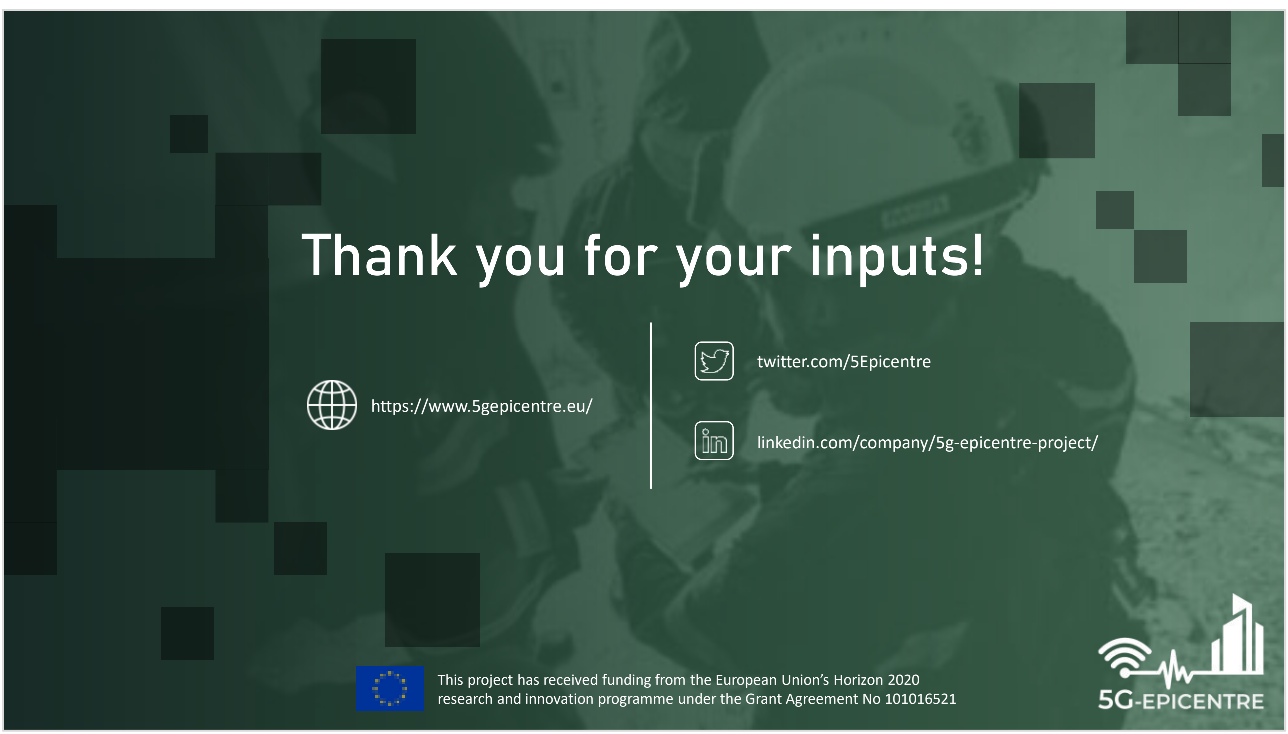


Figure 66: Architecture refinement – Component template distributed to partners (Back cover).