HEXA-X-II

A holistic flagship towards the 6G network platform and system, to inspire digital transformation, for the world to act together in meeting needs in society and ecosystems with novel 6G services

Deliverable D3.5
Final architectural framework and analysis

Co-funded by the European Union

6GSNS

Hexa-X-II project has received funding from the Smart Networks and Services Joint Undertaking (SNS JU)Smart Networks and Services Joint Undertaking (SNS JU)Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101095759.

Date of delivery:	28/02/2025	Version:	1.0
Project reference:	101095759	Call:	HORIZON-JU-SNS-2022
Start date of project:	01/01/2023	Duration:	30 months

Document properties:

Document Number:	D3.5
Document Title:	Final architectural framework and analysis
Editor(s):	**Ozgur Akgul** (NFI), Mårten Ericson (EAB), Panagiotis Botsinis (APP), Merve Saimler (EBY), (WIN), Antonio de la Oliva (UC3)
Authors:	Ozgur Akgul (NFI), Bassem Arar (TUD), Sokratis Barmpounakis (WIN), Riccardo Bassoli (TUD), Jaap van de Beek (LTU), Giacomo Bernini (NXW), Giulio Bottari (EAB), Pere Garau Burguera (AAU), Panagiotis Charatsaris (ICC), Maria Diamanti (ICC), Panagiotis Botsinis (APP), Sameh Eldessoki (APP), Mårten Ericson (EAB), Luca Feltrin (EAB), Frank H. P. Fitzek (TUD), Ece Goshi (NGE), Alperen Gundogan (APP), Payal Gupta (LTU), Mohammad Asif Habibi (TUK), Hasanin Harkous (NGE), Hamed Hellaoui (NFI), Selim Ickin (EAB), Paola Iovanna (EAB), Wolfgang John (EAB), Faryal Junaid (LTU), Umur Karabulut (NGE), Bahare M. Khorsandi (NGE), Apostolos Kousaridas (NGE), Karol Kuczyński (OPL), Gerald Kunzmann (NGE), Tero Lotjonen (NFI), Enrique Lluesma Marti (ATO), Amirreza Moradi (LTU), Swaraj S. Nande (TUD), David Navratil (NFI), Nikhitha Nunavath (TUD), Jan Palimąka (OPL), Arled Papa (NGE), Symeon Papavassiliou (ICC), Ignacio Labrador Pavón (ASA), Janusz Pieczerak (OPL), Roberto Querio (TIM), Vignesh Raman (TUD), Luís Santos (UBW), Hans D. Schotten (TUK), Erin E. Seder (NXW), Vivek Sharma (SON), Mohammad Soliman (NGE), Panagiotis Spapis (NGE), Alexandros Stylos (ICC), Halina Tarasiuk (OPL), Nassima Toumi (TNO), Vasileios Tsekenis (WIN), Antonio Varvara (TIM), Stefanos Voikos (ICC), Stefan Wänstedt (EAB), Marcin Ziółkowski (OPL), Milan Zivkovic (APP)
Contractual Date of Delivery:	28/02/2025
Dissemination level:	PU[1]
Status:	Final
Version:	1.0
File Name:	Hexa-X-II_D3.5_v1.0

Revision History

Revision	**Date**	**Issued by**	**Description**
0.1	2024-05-17	Hexa-X-II WP3	Template for Deliverables/IRs
0.2	2024-12-17		Internal review

[1] SEN = Sensitive, only members of the consortium (including the Commission Services). Limited under the conditions of the Grant Agreement

PU = Public

0.3	2025-01-09		External review
1.0	2025-02-28	Hexa-X-II WP3	Final version submitted to EC

Abstract

This is the third public deliverable from Hexa-X-II project work package 3 – “Final architectural framework and analysis”. This deliverable finalizes the data driven architecture to power new services, both for communication and for beyond communications. In addition, the deliverable describes new means for a modular cloud-native network to improve flexibility and reduce signalling, as well as enablers for new access and flexible topologies for improved reliability. The architecture inherently uses AI, both for orchestration and as a service, and incorporates NTN for enabling ubiquitous coverage.

Keywords

6G architecture, data-driven architecture, AIaaS, modular networks, ISAC, flexible topologies, cloud transformation

Disclaimer

Funded by the European Union. The views and opinions expressed are however those of the author(s) only and do not necessarily reflect the views of Hexa-X-II Consortium nor those of the European Union or Horizon Europe SNS JU. Neither the European Union nor the granting authority can be held responsible for them.

Executive Summary

The Hexa-X-II project is a flagship initiative bringing together key stakeholders in Europe for 6G research, continuing the work of the Hexa-X project. Hexa-X-II includes the key industry players in telecom and major research institutes; a combination of innovative knowledge capable of introducing new value chains for future connectivity solutions. Furthermore, the Hexa-X-II project comprises several work packages that span over important parts of the 6G ecosystem. In this report, results from work in WP3, which deals with the 6G architecture design, are presented.

The overarching objective of WP3 is to develop a 6G architecture framework with innovative enablers for beyond communication services. The architecture should be data driven to efficiently power new services, modular to support cloud-native networks and to improve signalling as well as allow new access and flexible topologies for improved reliability. This is the third and final public deliverable from WP3, called D3.5 “Final architectural framework and analysis”.

The main objective of this deliverable is to continue the analyse of the enablers introduced in previous deliverables [HEX223-D32] [HEX224-D33]. The document comprises descriptions of the enablers and studies of the different solutions within the enabler. Each enabler is thereafter summarized, including the benefits and the implications if the enabler would be implemented in the 6G architecture.

An important aspect of the architecture is how to support the migration between 5G and 6G. Therefore, the deliverable analyses the outcome of the 4G to 5G migration, and discusses the lessons learned and how to apply them to the 5G to 6G migration. One proposal is to use spectrum sharing for the 5G and 6G integration and the core network should be based on an evolved 5GC.

Several enablers have been identified for the data-driven architecture, comprising of DataOps, MLOps, and AIaaS. The proposed architecture is based on newly introduced or extended functions, such as the DataOps for handling data lifecycle tasks, the MLOps for end-to-end AI workflow management, the AIaaS for exposing AI functionalities. The deliverable describes a technical analysis focusing on issues such as data exposure and quality assurance, AI model lifecycle management and privacy-preserving learning methods, distributed computing orchestration and failure prediction strategies. All these enablers form the AIaaS Framework.

In a similar manner for beyond communication, several enablers have been defined already from the start. These enablers are ISAC, compute offloading and optimized application placement. The ISAC enabler defines the necessary architecture framework for 6G sensing. New functionality is needed to handle sensing requests, both in the form of an application that sends a sensing request and in the form of a function that receives and processes the request. A sensing management function is thereafter needed to further make use of the information in the request and transform the request information into usable sensing units (e.g. units located in the correct places). There must also be suitable forms of exposure, (radio) resources needed for the actual sensing measurements.

A compute offloading architecture framework is defined in this deliverable. The compute offloading framework can be integrated in cellular network with no significant changes to RAN and NAS protocols, thus making it less disruptive to existing cellular standards, allowing for faster time to market realization. In addition, several procedures are detailed, for example on how to handle the offload node discovery, node registration and offload procedures.

NTN and trustworthy flexible topologies were introduced as part of the network of networks enabler, which includes NTN architectural options and TN-NTN integration, subnetworks with new node roles and node coordination via various architectural options. Regarding the multi-connectivity solutions, proposals for the CA/DC evolution have been stated in detail aiming for one single solution based on CA. Last, solutions for optimizing the use of the transport network infrastructure and packet switching have been presented as part of the context-aware management of transport resources

It is found that there is a trade-off between performance and flexibility when considering the design of modular 6G architecture: More granular design results in higher flexibility in implementing and deploying modules but

at the cost of reduced performance in terms of execution time and state management. Furthermore, procedure-based functional decomposition of the Core network control plane is evaluated. The analysis show that there is a reduction in the total number of messages needed for these procedures by using procedure based functions. However, this design reduces the flexibility in deploying the more coarse-grained NFs. There is a need to improve the cloud friendliness of the interface between RAN and the CN. For this to happen. It is also found that instead of using SBA in the CN, a Data-Centric Networking solution can enable enhanced scalability and flexibility through dynamic stateless NFs and simplified architecture with efficient resource management. Different RAN architectures have an impact on the performance for D-MIMO, indicating that a 6G RAN architecture with lower split may be suitable for D-MIMO operations.

The cloud transformation for 6G includes how to evolve the cloud to be multi-cloud federation enabled. It is shown that a multi-cloud leads to higher resource availability and flexibility also lowers the total network load and latency times in service deployment, as well the ability to leverage computing power and storage that is superior to current solutions. There should be a unified and abstract interface for compute continuum resource management (inventory, provision, operate).

In addition to this, this deliverable has also investigated how to achieve several of the so-called quantified targets. We show that the (user plane) latency target of 1 ms can be achieved using an optimal deployment. Furthermore, the coverage target of >99% of the earth area for an essential service coverage, it is shown that a basic 6G service can be supported (depending on bandwidth).

The deliverable also includes a detailed description of the two proof-of-concepts (PoCs) belonging to WP3.

Table of Contents

List of Tables

List of Figures

Acronyms and abbreviations

Term	Description
3GPP	3rd Generation Partnership Project
5GC	5G Core
6G CSI	6G Channel State Information
6G SA	6G Standalone
6GC	6G Core
6GS	6G System
ADAES	Analytics Enablement Server
ADRF	Access Domain Resource Function
AI	Artificial Intelligence
AIaaS	AI as a Service
AIaaSF	AI as a Service Function
AIMLE	AI/ML Enablement
AMF	Access and Mobility management Function
ANC	Ad-hoc Network Controller
AoA	Angle of Arrival
AP	Access Point
APIs	Application Programming Interface
AR	Augmented Reality
ASN.1	Abstract Syntax Notation 1
ASP	Application Service Provider
AUSF	Authentication Server Function
BCN	Beyond Communication Network
BCS	Beyond Communication Service
BN	Border Node
BS	Base Station
CA	carrier aggregation
CAPIF	Common API Framework
CCN	Compute Control Node
CDF	Cumulative density function
CEI	Cloud-Edge-IoT
CHO	Conditional Handover
CI/CD	Continuous Integration and Continuous Delivery
CN	Core Network

CNF	Cloud native NF
CompN	Compute Node
CP	Control Plane
CPMS	Compute Process Management Service
CRMS	Compute Resource Management Service
CRS	Cell Reference Signal.
CSP	Communication Service Provider
DataF	Data Function
DataOps	Data Operations
DC	Dual Connectivity
DCCF	Data Collection Coordination Function
DCN	Data-Centric Networking
DFP	Data Flow Programming
DLM	Downlink Module
DNS	Domain Name System
DP	Data Plane
DU	Distributed Unit
E2E	End-to-End
E-5GC	Evolved 5G Core
EASDF	Edge Applications Service Discovery Function
EC	Edge Computing
eMBB	Enhanced Mobile Broadband
ENF	Enabler Network Functions
E-RAB	E-UTRAN Radio Access Bearer
ETSI	European Telecommunications Standardization Institute
FIFO	First in, first out
FL	Federated Learning
FMSF	Flexible Mesh Selection Function
FRF	Frequency Reuse Factor
F-RRC	Full RRC
FTM	Fine Time Measurement
FTN	Flexible Topology Node
GH	Group Head
gNB	Next Generation Node B
GPS	Global Positioning System
GUAMI	Globally Unique AMF ID

HAPS	High Altitude Platform Station
HFL	Hierarchical Federated Learning
HLS	High Level Split
IAB	Integrated Access and Backhaul
INC	Integration of Network and Compute
IoT	Internet of Things
IP	Internet Protocol
ISAC	Integrated Sensing and Communication
ISL	Inter-Satellite Link
ISM	Ingress Steering Module
JSON	JavaScript Object Notation
KPI	Key Performance Indicators
KVI	Key Value Indicators
LCM	Lifecycle Management
LEO	Low Earth Orbit
LLS	Low Layer Split
L-RRC	Lean RRC
LSTM	Long Short-Term Memory
LWA	LTE-WLAN Aggregation
LWIP	LTE-WLAN radio level Integration using IPsec tunnel
LWIPEP	LWIP Encapsulation Protocol
MaaS	Mobility as a Service
MAC	Medium Access Control
MEC	Multi-Access Edge Computing
MESE	Minimum Efficient Satisfaction Equilibrium
MgNB	Master gNB
MgtN	Management Node
MIMO	Multiple-Input Multiple-Output
MLOps	Machine Learning Operations
MLOps-aaS	MLOps-as-a-Service
MLOpsF	Model Lifecycle Management Orchestrator
MLP	Multi-layer Perceptron
MN	Master Node
MNIST	Modified National Institute of Standards and Technology
MNO	Mobile Network Operator
MRSS	Multi-Radio Spectrum Sharing

MSC	Message Sequence Chart
MST	Minimum Spanning Tree
mUPFs	Modular UPF
NAS	Non-Access Stratum
NAT	Network Address Translation
NE	Nash Equilibrium
NEF	Network Exposure Function
NF	Network Function
NGAP	NG Application Protocol
NIKSS	Native In-Kernel SDN Switch
NN	Neural Network
NOMA	Non-Orthogonal Multiple Access
non-IID	non-Independently Identically Distributed
NPN	Non-Public Network
NRF	Network Repository Function
NSA	Non-Standalone Access
NSSAI	Network Slice Selection Assistance Information
NTN	Non-Terrestrial Networks
NW	Network
NWDAF	Network Data Analytics Function
OAM	Operations, Administration, and Maintenance
ODM	On Demand Modules
OFDMA	Orthogonal Frequency Division Multiple Access
OPEX	Operating Expenses
OvS	Open vSwitch
P2P	Point to Point
PCF	Policy Control Function
PCT	Procedure Completion Time
PDU	Protocol Data Unit
PLMN	Public Land Mobile Network
PNF	Physical Network Function
PNI	Private Network Interconnect
PP5GS	Per-procedure 5GS
PUCCH	Physical Uplink Control Channel
QoS	Quality of Service
QUIC	Quick UDP Internet Connections

RA	Random Access
RACH	Random Access Channel
RAN	Radio Access Control
RAT	Radio Access Technology
RB	Resource block
REST	Representational State Transfer
RIC	RAN Intelligent Controller
RL	Reinforcement Learning
RLF	Radio Link Failure
RRC	Radio Resource Control
RS	Reference Signal
RSMA	Rate-Splitting Multiple Access
RU	Radio Unit
SBA	Service based arch
SBI	Service based interfaces
SCTP	Stream Control Transmission Protocol
SDL	Supplementary Downlink
SDN	Software Defined Networking
SE	Satisfaction Equilibrium
SEAL	Service Exposure Application Layer
SeMF	Sensing Management Function
SFC	Service Function Chaining
SgMB	Secondary gNB
SIB	System Information Block
SL	Split Learning
SLA	Service Level Agreement
SLAM	Simultaneous Localization and Mapping
SLO	Service Level Objective
SMF	Session Management Function
SN	Secondary Node
snCP	subnetwork Control Plane
SNIP	Secret-Shared non-Interactive Proof
SNR	Signal-to-Noise Ratio
SPF	Sender Policy Framework
SQP	Sequential Quadratic Programming
SubNW	Subnetwork

SVR	Support Vector Regression
TCO	Total Cost of Ownership
TDD	Time Division Duplex
TEF	Trust Evaluation Function
TLNA	Tracking Location and Network Assistance
TN	Transport Network
TS	Traffic Source
TTFB	Time to First Byte
TTL	Time to Live
UAV	Unmanned Aerial Vehicle
UCI	uplink control information
UDM	Unified Data Management
UDR	Unified Data Repository
UDSF	User Data Storage Function
UE	User Equipment
UL	Uplink
ULM	Uplink Module
UP	User Plane
UPF	User Plane Function
UPPCF	User Plane Path Control Function
UPRF	User Plane Registry Function
URA	Uniform Rectangular Array
URLLC	Ultra-Reliable Low-Latency Communications
Uu	Interface between gNB and the User Equipment.
VAL	Value Added Layer
vFL	Vertical Federated Learning
VNF	Virtual Network Function
WRAP	WLAN Relay Adaptation Protocol
XGBoost	Extreme Gradient Boosting
Xn	Interface between gNBs

1 Introduction

The Hexa-X-II project is a flagship initiative bringing together key stakeholders in Europe for 6G research. Hexa-X-II includes the key industry players in telecom and major research institutes; a combination capable of introducing new value chains for future connectivity solutions. In this report, results from work in WP3 are presented, which deals with the 6G architecture design.

The overarching objective of WP3 is to develop a 6G architecture framework and innovative enablers for a data driven architecture capable of powering new services, such as beyond communications services, a modular cloud-native network for improved signalling performance as well as new access and flexible topologies for improved reliability. This is the third public deliverable from WP3, called D3.5 “Final analysis of architectural enablers and framework and builds on previous deliverable [HEX224-D33].

1.1 Objectives

The main objective of this deliverable is to make the final analysis of the WP3 enablers. The long-term objectives of WP3 are presented in Table 1-1.

Table 1-1 WP3 Objectives

Objective	Objective description	Chapter
WPO3.1: 6G architecture for AI and beyond communications	Develop and analyse a 6G architecture framework and new innovative enablers for the beyond communications and data driven architecture, identify requirements a data-driven architecture will have on protocols, interfaces, data, and network nodes.	Chapter 3
WPO3.2: Combine the cloud technology for a modular, scalable and extendable architecture	Define and analyse solutions that combine cloud technology flexibility with distributed processing nodes into self-contained modules with minimum dependency that can be used to extend and scale the network deployments in stepwise manner.	Chapter 5
WPO3.3: Architecture for flexible topologies	Develop and analyse new access for flexible topologies and local communications, including different types of multi-connectivity, node roles and node coordination, as well as design control and management solutions for programmable and context-aware transport.	Chapter 4

1.2 Enabler definition and Methodology

In this deliverable, the term “enabler” is used extensively; enabler is defined as a technical area, usually aiming to improve the same KPIs. The technical area (or enabler) may contain several different types of solutions (or components) aiming for the same goal or KPI. For example, the enabler Network of networks in Section 4.3, aims to develop a seamless and ubiquitous communication system, targeting improved coverage and reliability. To solve this, several different solutions (or components) are necessary, for example the use of Non-Terrestrial Networks (NTN) and different types of (terrestrial) sub-networking solutions.

The methodology of this deliverable is to investigate and analyse the architectural implications for each enabler and the different components belonging to the enabler. The architectural implications describe how the architecture needs to be modified in order to implement and introduce the enabler in the 6G architecture. For some of the enablers there are also dedicated evaluation sections with more focus on simulation results, etc. Next, the enablers are summarized based on their benefits and implications in a 6G system. As can be seen in Table 1-1, the WP3 objectives cover broad areas (technically broader areas than the enablers in many cases), and this final deliverable, comprises an attempt to analyse how different enablers will work together in order to fulfil the WP3 objectives.

1.3 Structure

This document is structured as follows: Chapter 2 gives a brief overview of the envisioned architecture and 5G to 6G migration. Chapter 3 describes the Novel services for 6G, including AI enablers for a data driven architecture, sensing and compute offloading. Chapter 4 describes new 6G radio access for flexible topologies and local communications. Chapter 5 is about the Transformed architecture, i.e., how to build an architecture of modules that can scale based on current needs as well as the cloud transformation. Chapter 6 is a summary of how the quantified targets defined by Hexa-X-II is achieved. Chapter 7 summarizes and concludes the deliverable. Chapter 8 contains the references and Annex A provides more details from some of the enablers and studies. Finally, Annex B shows the details on how the quantified targets are achieved, including numerous simulations results.

2 6G Architecture introduction

This chapter introduces the 6G architecture from WP3. Figure 2-1 shows the so called 6G End-to-End (E2E) system blueprint [HEX224-D23] with the area of the WP3 objectives highlighted in red. The blueprint consists of four layers: Application layer, Application enablement platform layer, Network functions layer, and Infrastructure layer as well as a set of "Pervasive functionalities", which can reside in multiple layers. Figure 2-1 is used throughput the deliverable as a way to explain the specific topics in the context of the overall view.

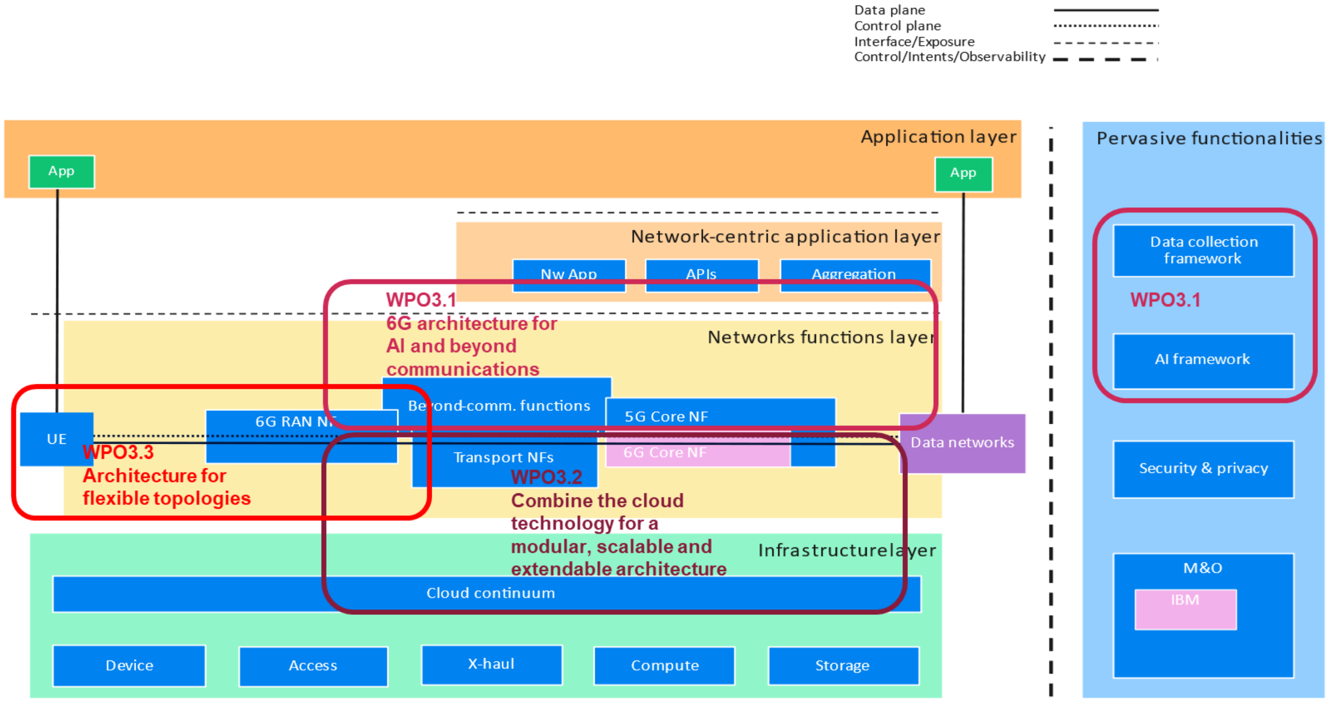


Figure 2-1 Illustrative mapping of WP3 objectives to the system blueprint

The objective WPO3.1 "6G architecture for AI and beyond communications" includes data driven functionalities, such as Machine Learning Operations (MLOps), Data Operations (DataOps), exposure of AI services, and AI as a Service (AIaaS). It also contains the Beyond Communication Network (BCN) functions, such as sensing and compute offloading as well as means for how to expose their services. The proposed architecture frameworks differs between AI, compute and sensing. The AI functions are pervasive and can belong to any layer (see section 3.1.4 for more details), while the compute framework is using a "over the top" solution with minimal impact on the network function layer (see section 3.3), while the sensing architecture framework belongs to the network function layer (see section 3.2).

The next objective WPO3.2 "Combine the cloud technology for a modular, scalable and extendable architecture" addresses how the RAN and CN NFs can communicate with each other in an efficient manner as well as how to optimally (from an E2E view) place the functions, see section 5.1. The objective also addresses the cloud transformation and the so-called cloud continuum. As can be seen in Figure 2-1, this objective maps to both the infrastructure (in particular the cloud continuum) and the Network functions layer.

The last objective is WPO3.3 "Architecture for flexible topologies". This objective maps to the Network functions layer (see section 4.1) and develops existing and new 6G sub-networks, including NTN, mesh networks and device networks. The objective is to develop new multi-connectivity solutions.

Another important aspect is the migration from 5G to 6G, since this can affect how the 6G architecture needs to be designed. This is treated in section 2.1.

2.1 Migration from 5G to 6G

In [HEX224-D33] several migration options were discussed. It was there concluded that Option 2 is the most viable option. Option 2 includes a 6G RAN that is deployed standalone and connected to a so called "Evolved

5GC (E-5GC)" (also referred to as "6GC" in the present document). The E-5GC (as Hexa-X-II terminology) is a trade-off between a pure evolution of the 5GC (Option 1) and a completely new 6G Core (Option 3) in the sense that "E-5GC" (Option 2) allows re-use of existing 5GC NFs, while introducing dedicated 6G NFs where justified. Multi-Radio Spectrum Sharing (MRSS) as a spectrum migration solution can support interworking between 5G and 6G [HEX223-D43]. The main reason to not use a dual connectivity solution for interworking between 5G and 6G is that it leads to a non-standalone (NSA) 6G deployment model in practice, and this is expected to delay the introduction of new 6G services (as experienced with 5G NSA), see section 2.2 for further analysis. Furthermore, the use of dual connectivity in 5G also caused numerous architecture options which led to unnecessary complexity. The pros and cons are further evaluated below in section 2.3.

Another aspect to consider is that 6G aims to support new services and use cases. Although AI and ISAC are the most visible service examples to place new requirements on the E2E network, there will be others not yet anticipated. Especially regarding the first two services (ISAC and AI), there may be a need for new KPIs or characteristics of the radio access, in addition to what is provided through current QoS requirements. Building upon the Service Based Architecture (SBA) of 5G, 6G can target streamlined NF design by collocating or refactoring 5G NFs as well as new 6G NFs. This streamlined design can target different aspects of the network, e.g. reduced signalling, increased flexibility or communication overhead, etc. Modular networks also bring out challenges in the interactions between different modules and orchestration of the modules and entities.

Key Requirements

Sustainability covers social, economic and environmental aspects [ref to wp1]. The foundation of 6G design will integrate all of those aspects. From the social point of view, 6G systems need to support digital inclusion for the society which may call for TN – NTN integration and/or coexistence. The 6G Core Network Functions (6GC NF) need to not only support this coexistence but further extend it to a cooperation. From an economical perspective 6G CN design needs to ensure simple migration from 5G to 6G as well as business potential. In particular, in order to ensure the gradual migration from 5G to 6G, in early 6G, 5G NFs should be used as much as possible. New 6G NFs should be defined only if required and justified. Finally, ensuring the environmental sustainability is one of the key success factors of 6G. The ongoing climate change as well as rising energy prices force an integration of the native energy efficiency to the 6G era network design. Therefore, the 6G NFs need to be streamlined to be energy efficient both in their operation (e.g., procedures, operation etc.) and design.

From the network performance perspective, 6G is envisioned to exceed 5G performance which puts additional requirements to the CN design. In addition to the well-known KPIs, such as lower latency or high throughput, the 6G network needs to ensure streamlined operations with decreased number of standardized APIs, network functions and signalling. Similar to 5G, 6G needs to ensure backwards compatibility and interoperability on hardware and software. Finally, 6G needs to further enhance the deployment and operational flexibility by revisiting the 5G CN design where needed.

2.2 Single 6G architecture option

The main architecture options that had been evaluated for 6G migration were detailed in [HEXX223-D33]. It was concluded that Option 2 (cf., Figure 2-2) is the preferred 6G architecture design. In particular**,** 6G RAN is deployed stand-alone and connected to an E-5GC. 6G intra-RAT multi-connectivity using an enhanced carrier aggregation (CA) and/or 6G-6G DC can be used to combine capacity and coverage bands that are dynamically shared with 5G via MRSS [HEX223-D43]. Mobility/inter-working between 5GS and 6G System (6GS) is core-based and realized via inter-RAT hand-over.

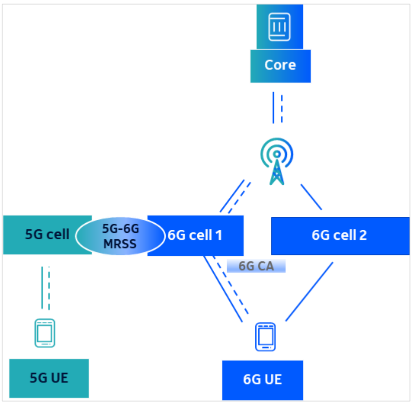


Figure 2-2 Preferred architecture option for 6G, i.e., SA with E-5GC and using MRSS.

To avoid delays in introducing key 6G services, the 6G CN is recommended to be based on the 5G CN (i.e. avoid a disruptive approach), as most of the NFs can be reused between different generations. This design approach follows characteristics towards a G-agnostic core. In this design, the 6G CN shares selected NFs with 5GC, while it allows for introducing new NFs and non-backward compatible changes to existing NFs. Introducing a new NF service (for an existing NF) may further be done in a backward- & 5G- compatible manner. For the smooth interworking between generations, some 6G dedicated NFs (e.g., UPF, SMF or AMF) may be collocated with their 5G counterparts.

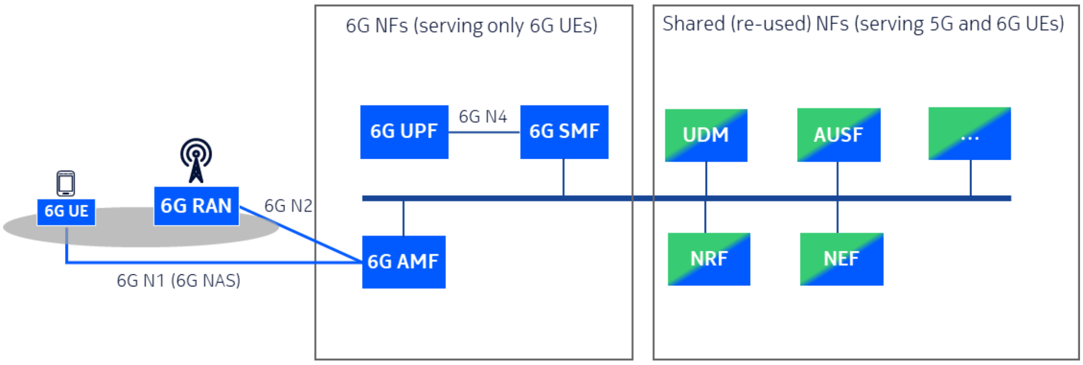

6G NFs (serving only 6G UEs)
Shared (re-used) NFs (serving 5G and 6G UEs)
6G UPF
6G N4
6G SMF
UDM
AUSF
...
6G UE
6G RAN
6G N2
6G AMF
NRF
NEF
6G N1 (6G NAS)

Figure 2-3 6G Core design as an evolution of 5GC with 6G NFs and Shared NFs

The solution for inter-RAT 6G and 5G handover with single registration is assuming an architecture with shared NFs (e.g., for UDM, AUSF, etc.) and combo NFs deployments where 5G and 6G NFs are collocated for smooth interworking (e.g., for SMF, UPF, AMF etc.; 5G-6G combo NFs are not depicted in the figure) to preserve IP session during mobility. The solution for inter-RAT handover depends on several other solutions for 6G which might evolve in 6G in comparison to 5G, such as the interface between control and user plane, the interface between RAN and CN and an evolution of NAS [HEX23-D53].

While the inter-RAT handovers are not required to provide lossless mobility, data forwarding between the source and the target RAN needs to be performed to minimize the packet loss during the handover. 3GPP standardized direct and indirect forwarding methods for the 4G and 5G interworking [23.502]. The indirect forwarding was standardized first in Rel-15, while the direct forwarding was introduced to the specification only one release later, i.e. Rel-16. It is assumed that initial 6G deployments will be with 5G and 6G hardware collocated on sites where networking infrastructure can support the direct forwarding. The direct forwarding does not load the backhaul and has a lower latency in the collocated deployments. Considering the expected rollout of 6G, the direct forwarding should be prioritized in 3GPP work. The indirect forwarding should be standardized only if there is a clear need coming from deployments, i.e. when the transport network does not allow for setting forwarding tunnels.

There are some basic principles that have been used to guide the standardized solutions for handovers in 3GPP [38.300], [23.502]. Principles of baseline handover between 5G and 6G RATs should follow these principles and (i.e. excluding fast handover) are summarized as follows:

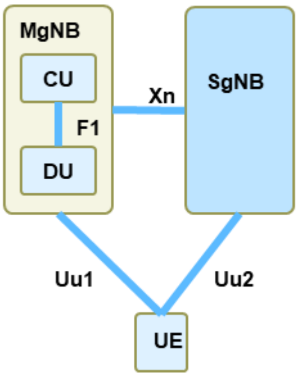
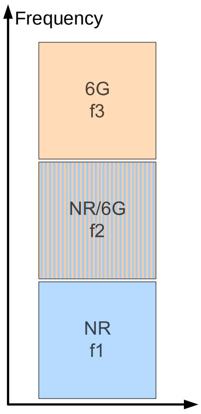
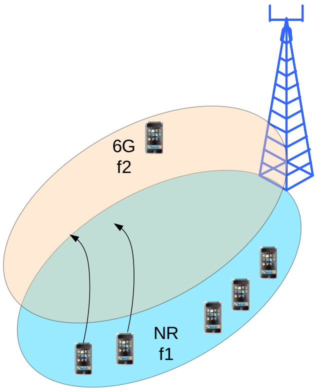
- The source RAT configures target RAT measurements and reporting.

- The source RAT decides on the preparation initiation and provides the necessary information to the target RAT in the format required by the target RAT.
- Radio resources are prepared in the target RAT before the handover.
- The Radio Resource Control (RRC) reconfiguration message from the target RAT is delivered to the source RAT via a transparent container and the message is passed to the UE by the source RAT in the handover command.
- In-sequence and lossless handovers are not supported.
- Security procedures for handover between 5G and 6G should be based on mapping between 5G and 6G security context by the 6G CN and the UE because of single registration.

2.3 5G - 6G interworking analysis

Table 2-1 shows the main alternatives for 5G to 6G interworking solutions, i.e. solutions that ensure that the initial 6G deployment can use the 5G network or resources to ensure good coverage.

Table 2-1 5G-6G interworking solutions

Aspect	Dual-connectivity	MRSS	Inter RAT Handover
			
Method	UE is connected to both 5G RAT and 6G RAT.	6G RAT can be employed in 5G spectrum dynamically; UE can use 6G in 5G spectrum	UE can switch to 5G if 6G coverage is bad (and vice versa)
UE configuration	UE is configured with both 5G and 6G	UE configured with only 6G	UE either uses 6G or 5G
Migration	A "spotty" coverage of the 6G cells can rely on 5G coverage	If MRSS is employed, the time-frequency resources in a cell are dynamically assigned to either 5G or 6G according to traffic demands.	Needs new spectrum for 6G, difficult to reallocate legacy spectrum.
Spectrum	Fixed spectrum split between 5G and 6G.	Dynamic spectrum split	Fixed spectrum split between 5G and 6G.
Reliability	Good	Good, if best frequency carrier has MRSS	Decent, there might be occasions when the HO fails
UE aspects	UE needs to split power in UL; UL coverage is always limited by the *worst* carrier	No impact	No impact
Challenges	Slow process to add new secondary, often session is completed before the SCG is connected. Limited in coverage by worst carrier in UL. Flow control between MN and SN	For best result, MRSS needs to be employed on low frequency bands.	May lead to inefficient use of spectrum due to fixed spectrum split. Control signal overhead may be an issue.

	causes stalling (see [HEX223-D32]). May lead to inefficient use of spectrum due to fixed spectrum split.		

Based the analysis shown in Table 2-1, DC is probably the least preferred option due to the rather many challenges and the complexity of the solution. Further on, MRSS the preferred option for 5G and 6G migration as well as interworking, mainly for the dynamic and granular use of spectrum between 5G and 6G. However, as a fallback for spectrum bands where MRSS is not activated, inter RAT handover can be used. For MRSS, it is of interest to understand the potential overhead from the 6G signals to the 5G system when using 5G spectrum. For MRSS, the legacy 5G UEs should be able to avoid all 6G signals, i.e., the 6G signals are "hidden" by reusing 5G locations in the time frequency grid. This means that the 5G UEs will not be affected by these 6G signals. For example, the 6G CSI-RS locations should be a superset of 5G CSI-RS time/frequency locations. Further on, the impact on the network performance and updates to the 5G standard configuration should be minimized. From the 6G perspective, MRSS should preferably not restrict design and operation of a 6G-only carrier and new 6G UEs should be able to avoid 5G signals.

In general, the 6G UL signals can be handled by scheduling 6G within the 5G carrier and thereby avoid overlap with 5G signals. However, the uplink control information (UCI) signal for 5G and 6G needs to be coordinated. The 5G UCI is essentially scheduled by 5G and any periodic PUCCH remains as in 5G. Similarly, the (possible) 6G UCI is scheduled by 6G. For RACH resources, it is possible to reuse the 5G RACH structure as long as the 6G RACH is similar enough to 5G RACH. All in all, MRSS in the UL should be possible with negligible overhead.

In DL, there are several signals that need to be considered. To estimate the signalling overhead incurred by MRSS, consider e.g., a scenario with no 6G traffic, a 20 MHz carrier, and 15 kHz subcarrier spacing. The fundamental 6G downlink signals which are always present are then as shown in Table 2-2, which shows a rough estimate of the DL MRSS overhead. Assuming a system with a carrier bandwidth of 20 MHz (100 Resource blocks) where the 5G SSB typically occupies 20 Resource blocks (RBs) and 4 out 14 symbols which are transmitted every 20 ms [Sam21]. Further, assuming the same SSB allocation for 6G, this means that the SSB signal gives roughly 0.3% overhead (see Table 2-2). If the SSB periodicity in 6G will be even longer, the overhead will decrease even more. In the same manner, the SIB1 gives and overhead of 1.2%. Similarly, if the bandwidth is increased, the overhead will decrease even further. Note that there is no CRS in either 5G or 6G as there is in 4G, so the overhead from the CRS signals can be avoided between 5G and 6G.

Table 2-2 Rough estimate of the DL MRSS overhead (no active 6G users)

6G signal	Resources assumed	Overhead (20 MHz, 100 RBs)
SSB	20 RBs and 4 OFDM symbols every 20ms	(20/100) * (4/14) * (1/20) = 0.3%
SIB1	48 RBs, 14 OFDM symbols, every 40 ms	(48/100) * 14 / (40*14) = 1.2%
SIB2	Only on demand	0%

All in all, this means that the overhead from an empty 6G carrier on top of 5G is ~1.5%. However, for scenarios with even higher bandwidths, the overhead will be even lower. For example, for a 100 MHz carrier (mid-band TDD), the corresponding overhead for MRSS is 0.3%. This clearly outperforms DSS (spectrum sharing between 4G and 5G) which incurs an overhead of at least 5% [Sam21].

2.4 6G RAN Core Network interfaces overview

Figure 2-4 shows the architectural design of the 6G End-to-End (E2E) system blueprint without the detailed interfaces. Taking the considerations of the migration aspects and using evolved 5G CN, the expected the main interfaces between the UE, RAN and CN in 6G is depicted in Figure 2-4. As can be seen, a Lower Layer Split (LLS) in the RAN will likely be used for 6G, instead of the High Level Split (HLS) used in 5G. The main reason for this is to simplify the RAN architecture, as well as to support D-MIMO. This is to some extent further analysed in section 4.2 about multi-connectivity for 6G and section 5.2.3.3 analysing cell free MIMO. Furthermore, it is expected that the N2 and N3 interfaces will be kept but probably need to be evolved to handle the demands for a more cloud friendly environment, see section 5.2.2 for a detailed analysis. Further, "AMF for 6G" can be an evolved AMF potentially connecting to both 5G RAN and 6G RAN, or it could be a dedicated 6G NF with revised functionalities if architectural benefits motivate it.

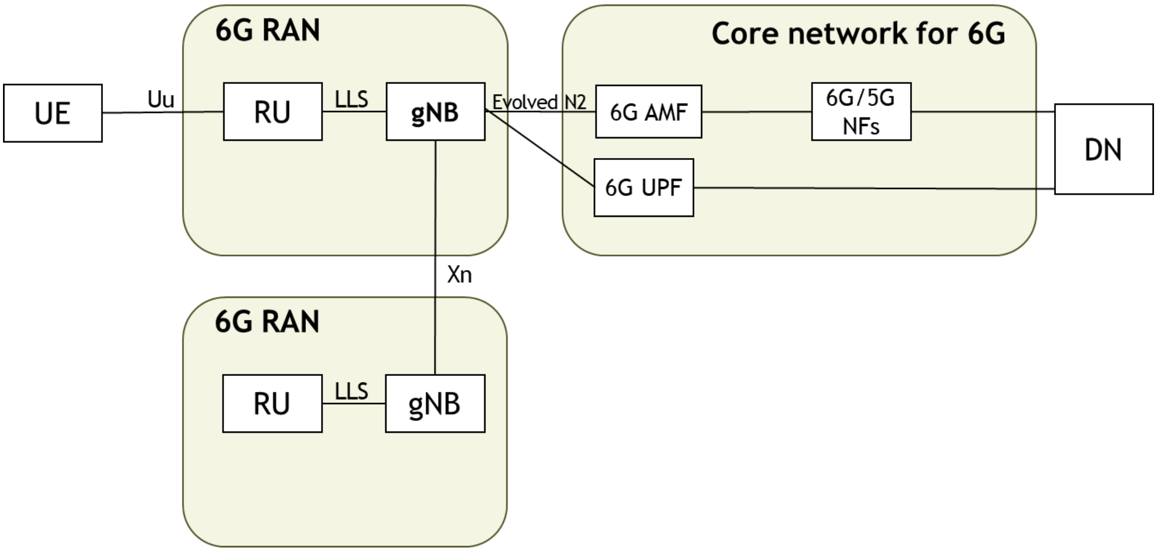


Figure 2-4 Overview of the expected UE-RAN–CN interfaces of 6G.

3 Novel services

This chapter addresses WPO3.1 “6G architecture for AI and beyond communications 6G architecture for AI and beyond communications” (see section 1.1) by developing and analysing a comprehensive 6G architecture framework tailored for beyond communications and AI and data-driven environments. Also, the chapter describes and outlines the necessary requirements for protocols and interfaces. The enablers for the novel services extend the traditional communication capabilities of the mobile network, introducing possibilities for new use cases such as physical awareness and collaborative robots [HEX223-D12]. These uses cases require extensive use of beyond communication enablers, such as, sensing, compute and AI. Each one of the novel services enablers are mapped to the 6G E2E system blueprint in Figure 3-1, where the green boxes represent the sensing enabler, the red boxes the compute and placement enablers, and finally the dark green the AI and data driven network enablers.

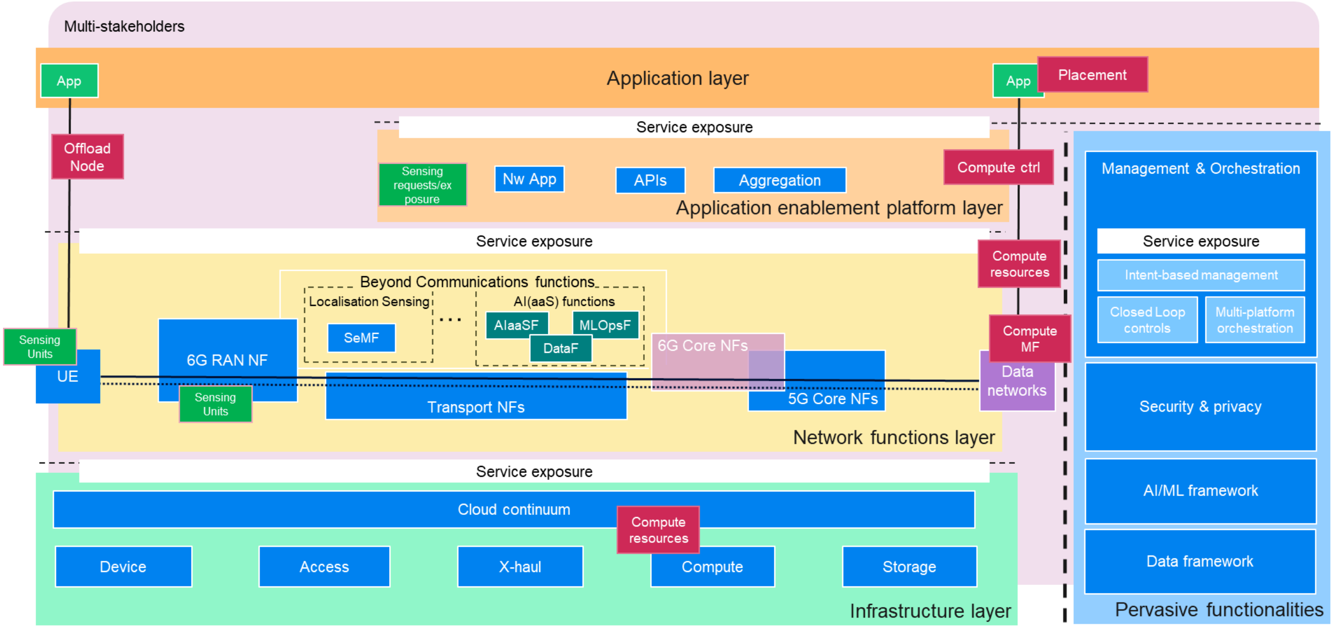


Figure 3-1 Mapping of the novel services enablers to the 6G E2E system blueprint of [HEX223-D23]

To enable AI for use in-network and exposed to consumers, the enablers Data Operations (DataOps), Machine Learning Operations (MLOps), and AI-as-a-Service (AIaaS) are required, see section 3.1. These enablers are interconnected and serve distinct roles in integrating AI-driven capabilities and compute services efficiently. DataOps provides the foundation for reliable, efficient, and scalable data pipelines, supporting MLOps and AIaaS by ensuring the availability of high-quality data. Building on this, MLOps specializes in operationalizing machine learning workflows, offering tools and methodologies for deploying and maintaining distributed AI functions. MLOps supports advanced techniques like Federated Learning (FL) and Split Learning (SL) for scalable model training and inference. AIaaS extends these capabilities further by delivering AI-native services within the 6G architecture, incorporating comprehensive MLOps functionalities and additional APIs for data exposure and Quality of Service (QoS). Additionally, AIaaS goes beyond MLOps by offering application-level services, exposing APIs, and enabling dynamic orchestration of AI models and related components in response to user and network demands.

AI-supported optimisation of placement of services and application across the network can improve latency, resource availability, overall performance and energy efficiency, see section 3.4.1. This dynamic placement enhances the efficiency and responsiveness of applications, meeting the stringent demands of next-generation services.

ISAC will utilize the existing 6G networks communication signals for sensing, e.g. locating obstacles on a road to help improve traffic safety. To enable ISAC, several functions are developed. Central to this integration is the Sensing Management Function (SeMF), see section3.2, which facilitates efficient coordination of sensing procedures by managing sensing requirements, capabilities, and constraints.

Compute offloading mechanisms allow mobile devices to dynamically offload computational tasks to more capable environments within the cellular network or to traditional data centers. This offloading not only optimizes performance and energy efficiency but also enhances scalability and resilience by balancing workloads across edge and cloud resources. Section 3.3 describes a possible compute offloading framework and some of the necessary high-level procedures, as well as methods for when to perform offloading.

3.1 AI Enablers for Data Driven Architecture

3.1.1 DataOps

3.1.1.1 Introduction

Building on established principles of DevOps and MLOps, DataOps introduces advanced functionalities tailored to manage the complexities of 6G. These include data quality assurance, automated data pipelines, integration of distributed data sources, and governance mechanisms. With these capabilities, DataOps facilitates the consistent preparation and provisioning of datasets for applications such as AI-driven network management, Digital Twins, and predictive analytics. It also ensures compliance with stringent data privacy and security standards, particularly critical in federated and multi-stakeholder environments.

To realize such a DataOps framework, a robust architecture is essential, incorporating components such as data provenance, historical data management, data cleansing and normalization, real-time pipeline monitoring, and feedback loops for continuous optimization. These elements enable the effective handling of federated data across heterogeneous edge and cloud sites, ensuring high-quality and timely data availability for AI/ML processes. Moreover, the standardization of DataOps functionalities and APIs is vital for interoperability and scalability.

3.1.1.2 Secure and Privacy Preserving DataOps for a 6G Network Slice

DataOps plays a crucial role in managing the lifecycle of data used for machine learning models within 6G network slicing, where network resources are dynamically allocated to serve specific applications. In 6G, federated learning offers an efficient approach to enable network slices to autonomously improve their performance using locally sourced data. By incorporating DataOps practices, the management of these data pipelines becomes more efficient, scalable, and secure. The 6G slicing is a multi-stakeholder framework, where each player is responsible for its own tasks defined either by an E2E or domain-level orchestrators. Such a multi-stakeholderism brings its own advantages as well as challenges to the framework, which are discussed in a detailed manner later in the document.

In a 6G slicing framework, data (both management and network data) is distributed across multiple edge and centralized locations, each representing different network slices with specific use cases, such as Internet of Things (IoT), enhanced mobile broadband (eMBB), or ultra-reliable low-latency communication (URLLC). Federated learning allows each node to train a machine learning model locally without transferring raw data to a centralized server, thus preserving privacy and reducing network overhead. DataOps optimizes this process by establishing standardized processes for data collection, preprocessing, quality assurance, and governance across these distributed edge nodes. These practices facilitate consistent data handling and reduce model drift due to inconsistent or low-quality data input. Figure 3-2 depicts the proposed architecture for DataOps in a 6G network slice.

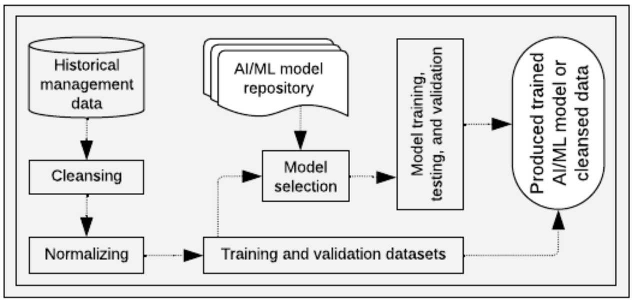

Historical management data
Cleansing
Normalizing
AI/ML model repository
Model selection
Training and validation datasets
Model training, testing, and validation
Produced trained AI/ML model or cleansed data

Figure 3-2 DataOps for 6G Network Slicing Framework

The architecture is composed of the following components:

Historical Management Data: This component serves as the central repository for historical data collected from multiple sources across network slices. It includes records of past data inputs, model performance metrics, and environmental conditions relevant to different slices (e.g., latency and bandwidth usage). Historical management data provides the foundation for creating effective data preprocessing rules and enables insights into long-term trends that can optimize model performance over time. It also assists in understanding the changing patterns in network demands, which helps refine data handling and model training strategies.

Cleansing: Data collected in a federated learning environment is often noisy and inconsistent due to the distributed nature of 6G networks. The cleansing component is responsible for identifying and rectifying anomalies, such as missing values or duplicate entries, ensuring that the data passed to the next stages is accurate and reliable. Cleansing mechanisms may involve outlier detection, deduplication, and the application of predefined filters to retain only relevant data for each network slice. This step is crucial for maintaining data integrity across all nodes in the federated setup.

Normalizing: Once cleansed, data normalization aligns various data formats and scales to a standard structure, facilitating seamless integration across the network's diverse nodes. This component standardizes data units and formats to ensure compatibility, enabling the model to process data from heterogeneous sources without bias or inconsistencies. For instance, normalization may convert timestamps to a unified format or standardize values such as bandwidth and latency, ensuring that the input data adheres to the model's requirements and prevents skewed predictions.

Training and Validation Datasets: After data has been cleansed and normalized, it is split into training and validation datasets, prepared for the federated learning process. The training dataset is used to train the model locally at each node, while the validation dataset allows for an independent evaluation to fine-tune the model and verify that it generalizes well to new data. The separation of datasets at each edge location is essential to the federated learning paradigm, as it helps prevent overfitting and ensures robust model performance across different network slices.

Model Repository: This centralized or distributed repository stores all versions of the trained models. The model repository maintains model checkpoints, metadata, and performance metrics, providing a structured way to manage the lifecycle of each model. It enables easy access to previously trained models for comparison, retraining, or rollback. This repository ensures model continuity and accountability, supporting reproducibility and compliance with federated learning protocols and privacy standards.

Model Selection: In federated learning, multiple models may be trained across different nodes, necessitating an efficient selection process. The model selection component evaluates model performance on validation datasets and identifies the optimal model that meets predefined criteria (e.g., accuracy, latency, or resource efficiency). This component selects either a single best-performing model or a group of top models for aggregation, ensuring that the final federated model reflects the highest standards of accuracy and adaptability required by the 6G slice.

Model Training and Testing: The selected model is iteratively trained and tested within the federated environment, leveraging data from distributed nodes to improve performance while respecting data privacy. This training and testing component integrates local training results from each node, often using federated averaging or similar techniques to combine the models. After each training round, the updated model is tested against validation data to confirm that it meets the quality benchmarks. This continuous loop ensures that the federated model adapts to new data patterns and maintains high performance across different slices.

Produced Trained Model: The output of the training and testing process is the produced trained model, which represents the aggregated knowledge from all participating nodes in the federated learning system. This model is now prepared for deployment across network slices, enabling autonomous, intelligent decision-making tailored to each slice's specific requirements. The produced trained model can be further stored in the model repository for future reference, monitoring, or retraining as network conditions evolve.

This architecture ensures that data management and model training are efficient, consistent, and secure in a federated learning environment. By addressing the complexities of data quality, privacy, and model management, it provides a structured approach to building adaptive machine learning models capable of supporting dynamic and heterogeneous 6G network slicing.

3.1.1.3 High-level procedures for privacy preserving data collection

The privacy-preserving architecture and novel architectural components for data collection, learning, and analytics are presented in [HEX223-D32]. As introduced in [HEX224-D33] and also described in Annex, a potential implementation may be based on privacy-preserving cryptographic protocols like Prio [CB17].

The potential deployment options of privacy-preserving (i.e., both data and model) aggregation procedures based on Prio protocol in Network Data Analytics Function (NWDAF) framework and corresponding high-level message exchange are further introduced.

Core network deployment of privacy-preserving data collection

An example of the core network deployment of privacy-preserving aggregation procedure in future cellular network is shown in Figure 3-3. Let us consider a simplified scenario by assuming two UEs and two aggregators, where the network function (NF) registers to Network Repository Function (NRF) for NWDAF together with DCCF (Data Collection Coordination Function) and Analytics Data Repository Function (ADRF) data collection service. Aggregators are implemented as AFs, while Collector is implemented in NWDAF as a service consumer. A Leader, an aggregator responsible for coordinating the data aggregation, resides in the network domain, and a Helper, an aggregator assisting the Leader in deriving the aggregate data statistics, is located in the Application Service Provider (ASP) domain. Both aggregators register to the NRF. Based on AF configurations and available resources, Leader could reside in network domain, while Helper could be realized in ASP domain.

NF initiates data collection service from the NWDAF. Based on the NF request for data collection service and registered AF characteristics, the NWDAF performs discovery of AFs from NRF and selects and subscribes to Leader and Helper(s). Furthermore, based on the selected AF and NF characteristics, NWDAF requests for PDU session establishment or modification for UEs participating in data aggregation [23.288].

Optionally, if requested by the NF, NWDAF performs data collection from the RAN and other NFs. Based on the properties of the secure aggregation AFs, i.e., Leader/Helper(s) configuration, UEs create data shares and secret-shared non-interactive proof (SNIP) [GPP+24], used for validation of aggregators, and send them to the Leader (AF1) and the Helper (AF2). Here, privacy is preserved under the assumption that aggregators do not contain the same data share. Before creating aggregated shares, the Leader (AF1) and the Helper (AF2) exchange and validate SNIP proofs to ensure validity of the data. The Collector (NWDAF) collects aggregate shares and calculates the aggregated statistics of the UE data. Finally, the NF consumes statistics of the collected data, maintaining UE privacy, i.e., without having direct access to privacy sensitive data).

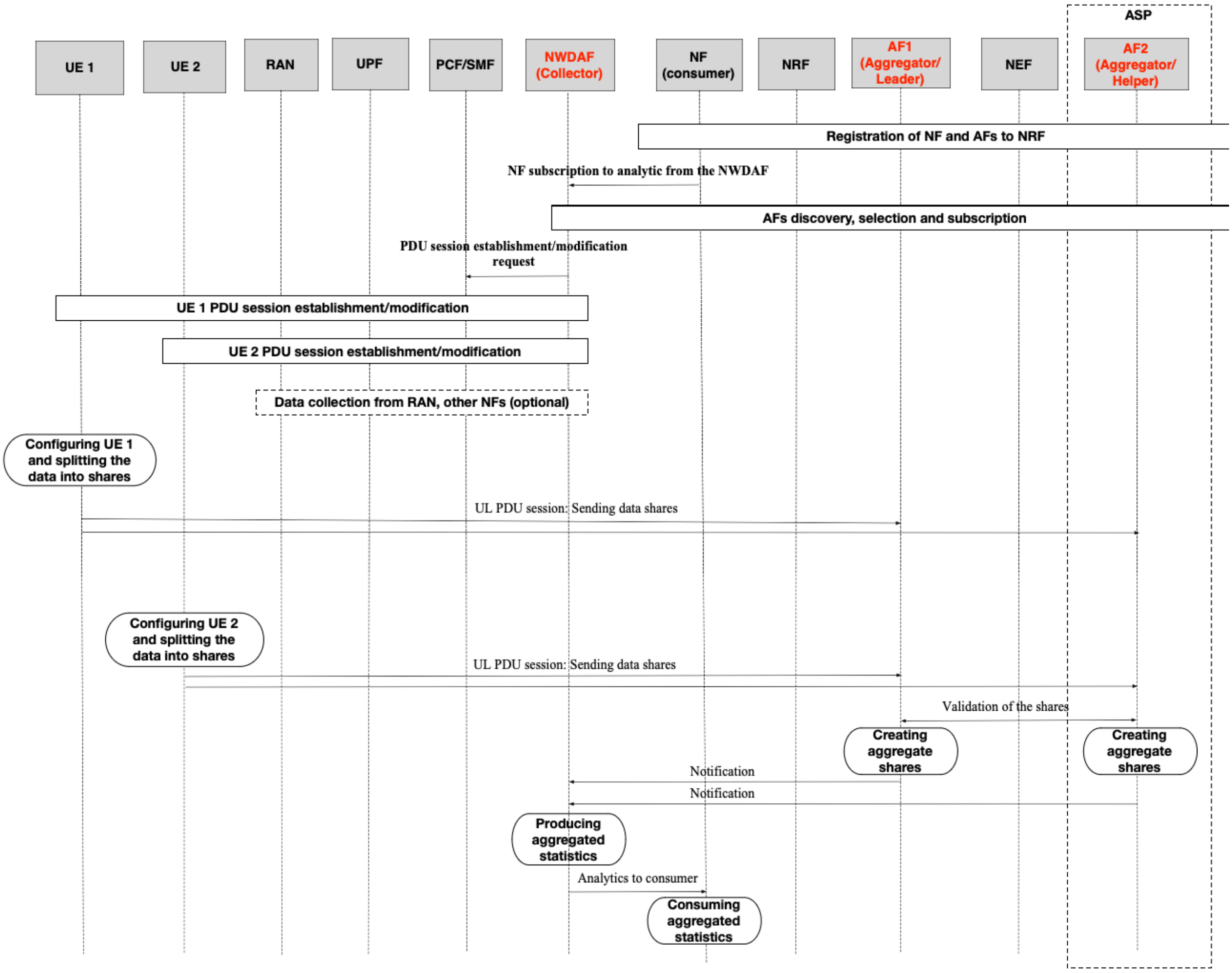

ASP
UE 1
UE 2
RAN
UPF
PCF/SMF
NWDAF (Collector)
NF (consumer)
NRF
AF1 (Aggregator/ Leader)
NEF
AF2 (Aggregator/ Helper)
Registration of NF and AFs to NRF
NF subscription to analytic from the NWDAF
AFs discovery, selection and subscription
PDU session establishment/modification request
UE 1 PDU session establishment/modification
UE 2 PDU session establishment/modification
Data collection from RAN, other NFs (optional)
Configuring UE 1 and splitting the data into shares
UL PDU session: Sending data shares
Configuring UE 2 and splitting the data into shares
UL PDU session: Sending data shares
Validation of the shares
Creating aggregate shares
Creating aggregate shares
Notification
Notification
Producing aggregated statistics
Analytics to consumer
Consuming aggregated statistics

Figure 3-3: An example of the core network deployment privacy-preserving aggregation procedure in future cellular network.

RAN deployment of privacy-preserving data collection

RAN deployment of the Prio protocol can be used for the collection of aggregate statistics of data, which could provide useful information (e.g., application type, number of applications, etc.) for optimizing different network procedures (e.g., mobility, network load, etc.). This contextual data, shared with the RAN, would allow for correlation of different configurations, contextual features, KPIs and learnt histograms. Further details on content and characteristics of contextual data are provided in the Annex A.

Two potential RAN deployments of privacy-preserving data collection are discussed. The first deployment, where the Collector is located in NWDAF, may be leveraged for more passive analytics, is introduced below.

The second deployment, where Collector is placed in RAN, may be used for the collection of live analytics, i.e., for obtaining the aggregated statistics in a smaller time scale [HEX224-D33], and is detailed in the Annex A.

RAN deployment with Collector in NWDAF

An example of the RAN deployment privacy-preserving aggregation procedure in future cellular network and corresponding high-level message exchange is shown in Figure 3-4.

Let us consider a simplified scenario by assuming two UEs and two aggregators, where the Collector is implemented in the NWDAF. The aggregate statistics Consumer in the RAN (e.g., network control unit/optimizer) registers to NRF for data collection~~.~~ The Leader, an aggregator responsible for coordinating the data aggregation, is deployed in the RAN and the Helper, an aggregator assisting the Leader in deriving the aggregate data statistics, resides in the ASP domain and registers to the NRF.

Consumer initiates data collection querying the UE and NW contextual data by sending *ppaCapabilityRequest* as an RRC message. UE responds with *ppaCapabilityResponse* RRC message containing UE and NW contextual data and the corresponding aggregation capabilities (i.e., granularity, need for Prio, need for DP, DP noise level, etc.). The Consumer then subscribes to the data collection service from NWDAF. Based on the Consumer request for data collection service and registered aggregators, NWDAF performs discovery of aggregators from NRF and subscribes to Leader and Helper.

NWDAF sends analytics request to RAN, which then sends *ppaContextDataReportRequest* RRC message, containing the content of UE and NW context to collect, as well as aggregators destinations. Moreover, based on selected aggregators and Consumer, NWDAF requests PDU session establishment or modification for UEs participating in data aggregation. If requested by the Consumer, NWDAF performs data collection from RAN and other NFs. Based on the properties of the secure aggregation, i.e., Leader/Helper(s) configuration, the UEs create a *ppaContextDataReport* containing data shares and secret-shared non-interactive proof (SNIP) [GPP+24], used for the validation of the aggregators. The UEs then send a *ppaContextDataReport* to the Leader via a RRC Message, and to the Helper (AF2) as an UL PDU load. Here, privacy is preserved under the assumption that aggregators do not contain the same data share. Before creating aggregated shares, the Leader and the Helper exchange and validate SNIP proofs to ensure validity of the data. The Collector (NWDAF) collects aggregate shares and calculate the aggregated statistics of UE and NW contextual data. Finally, the Consumer in the RAN leverages statistics of the collected data for network optimization without revealing UE privacy, i.e., without having direct access to privacy sensitive data.

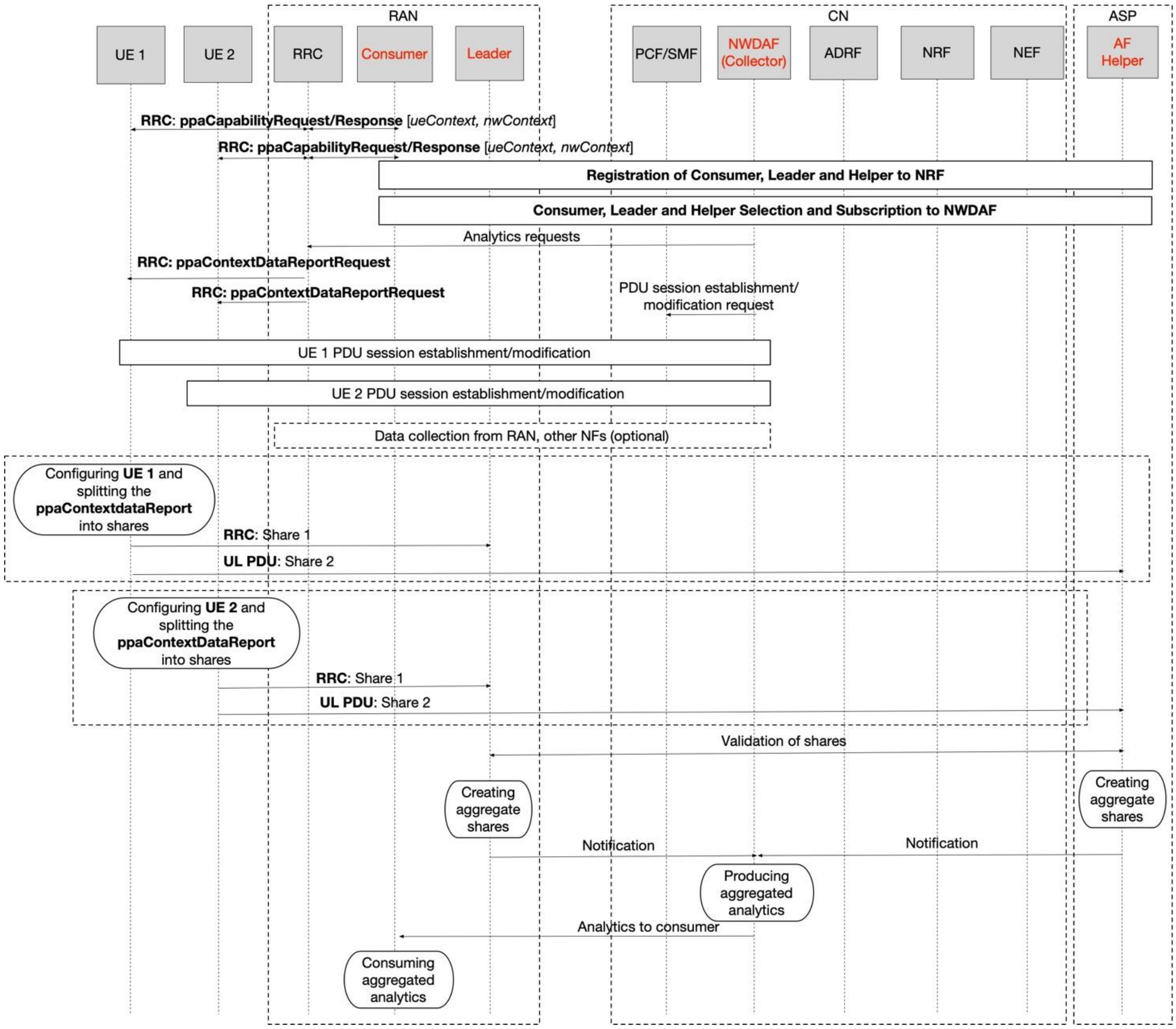

RAN
CN
ASP
UE 1
UE 2
RRC
Consumer
Leader
PCF/SMF
NWDAF (Collector)
ADRF
NRF
NEF
AF Helper
RRC: ppaCapabilityRequest/Response [ueContext, nwContext]
RRC: ppaCapabilityRequest/Response [ueContext, nwContext]
Registration of Consumer, Leader and Helper to NRF
Consumer, Leader and Helper Selection and Subscription to NWDAF
Analytics requests
RRC: ppaContextDataReportRequest
RRC: ppaContextDataReportRequest
PDU session establishment/ modification request
UE 1 PDU session establishment/modification
UE 2 PDU session establishment/modification
Data collection from RAN, other NFs (optional)
Configuring UE 1 and splitting the ppaContextdataReport into shares
RRC: Share 1
UL PDU: Share 2
Configuring UE 2 and splitting the ppaContextDataReport into shares
RRC: Share 1
UL PDU: Share 2
Validation of shares
Creating aggregate shares
Creating aggregate shares
Notification
Notification
Producing aggregated analytics
Analytics to consumer
Consuming aggregated analytics

Figure 3-4: An example of the RAN deployment privacy-preserving aggregation procedure in future cellular network (Option 1 – Collector in NWDAF).

3.1.1.4 Kubernetes-based failure detection and prediction data sourcing for 5G Core

Reliability requirements in considered use cases for 5G/6G networks (such as from robots to cobots, massive twinning, immersive telepresence for enhanced interactions) constitute the necessity for failure detection and prediction. The main motivation of the contribution was to provide a minimum set of metrics for a 5G procedure, based on which failure (risk of losing service continuity) detection or prediction can be carried out. For this purpose, the 5G Core Observability Platform (called 5GCOP) was developed in an experimental testbed. 5GCOP allows to collect, process and expose data sourced by the Observability and Monitoring tools. Figure 3-5 shows the 5GCOP architecture and its mapping to the 6G E2E system blueprint. Platform designed for 5G Core could be translated for 6G system in the future. The Linkerd [LIN24] Service Mesh was chosen for the experimental environment, due to the extensive number of available metrics [LIM24].

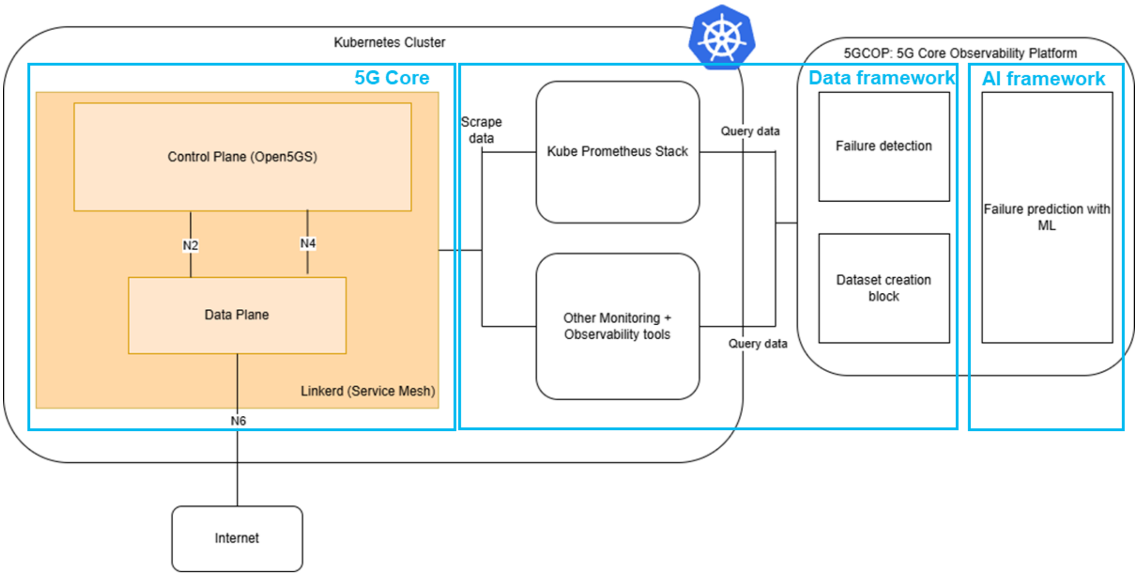
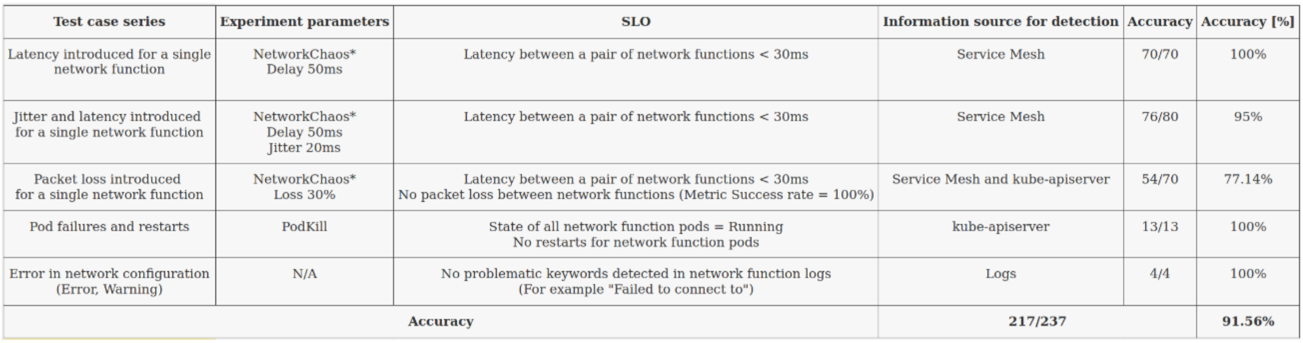


Figure 3-5 5GCOP architecture and its mapping to the 6G E2E system blueprint.

More details about the data sourcing and failure detection processes are included in the Annex A, section A.1.1 Table 3-1 and Table 3-2 present failure detection accuracy results obtained for experiments performed with the NetworkChaos tools, from the Chaos Engineering tool Chaos Mesh [CHE24]. Table 3-2 contains detailed results for the case of Jitter and Latency introduced for a single network function. A description of the experiments is included in Annex A, section A.1.1 .

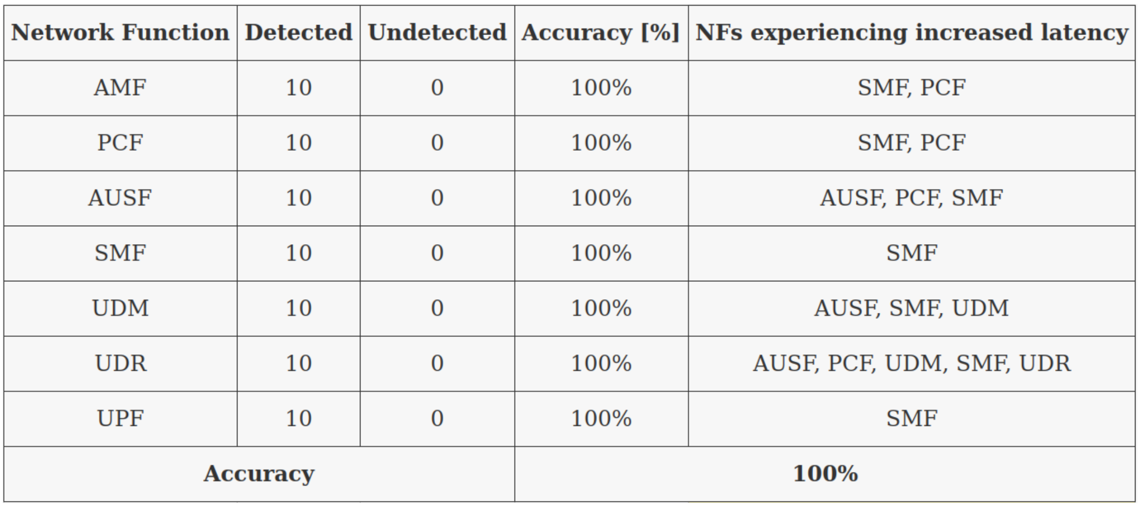
Table 3-1 Failure detection by 5GCOP – Accuracy results



Test case series	Experiment parameters	SLO	Information source for detection	Accuracy	Accuracy [%]
Latency introduced for a single network function	NetworkChaos* Delay 50ms	Latency between a pair of network functions < 30ms	Service Mesh	70/70	100%
Jitter and latency introduced for a single network function	NetworkChaos* Delay 50ms Jitter 20ms	Latency between a pair of network functions < 30ms	Service Mesh	76/80	95%
Packet loss introduced for a single network function	NetworkChaos* Loss 30%	Latency between a pair of network functions < 30ms No packet loss between network functions (Metric Success rate = 100%)	Service Mesh and kube-apiserver	54/70	77.14%
Pod failures and restarts	PodKill	State of all network function pods = Running No restarts for network function pods	kube-apiserver	13/13	100%
Error in network configuration (Error, Warning)	N/A	No problematic keywords detected in network function logs (For example "Failed to connect to")	Logs	4/4	100%
Accuracy			**217/237**		**91.56%**

* NetworkChaos - Fault type name from Chaos Engineering tool, Chaos Mesh

Table 3-2 Failure detection for Delay experiment



Network Function	Detected	Undetected	Accuracy [%]	NFs experiencing increased latency
AMF	10	0	100%	SMF, PCF
PCF	10	0	100%	SMF, PCF
AUSF	10	0	100%	AUSF, PCF, SMF
SMF	10	0	100%	SMF
UDM	10	0	100%	AUSF, SMF, UDM
UDR	10	0	100%	AUSF, PCF, UDM, SMF, UDR
UPF	10	0	100%	SMF
Accuracy			**100%**	

Based on the obtained results it can be observed, that introducing a Service Mesh in a data-scarce environment has been proven to be a valuable data source, allowing for failure detection accuracy to reach levels of over 90%. At the same time, the increase in Procedure Completion Time created by the Service Mesh sidecar containers warrants a need to evaluate existing Sidecarless Service Mesh tools as a replacement for Linkerd. Three extensive data sets (Idle 5G, 5G during the registration procedure, Failure in 5G during a procedure) were collected[2] for Machine Learning models for training purposes for the Failure Prediction Model.

Conclusions:

DataOps offers insights into state of the 5G/6G Core, supplying Machine Learning models with data necessary for Failure Prediction. The search for early indicators of a possible failure in the 5G Core network needs to minimize the impact of introduced tools on the resource usage and performance of the 5G Core. The addition of Linkerd Service Mesh tool supplies data, which is exposed and transformed to supply both Failure detection and prediction modules. This data makes it possible to achieve high failure detection accuracy, but the additional resource cost and scalability concerns warrant the need for different data sources (i.e. sidecarless Service Mesh, new Monitoring and Observability tools). The Failure Prediction Machine Learning module was implemented in order to improve the 5G Core reliability (see section 3.1.2.7).

3.1.1.5 Summary DataOps enabler

Table 3-3 summarizes the DataOps enabler.

Table 3-3 Summary of Key Aspects of DataOps in 6G Networks

Description	The DataOps enabler implements a data-centric platform seamlessly integrated within the 6G network architecture to manage the end-to-end data lifecycle efficiently. It provides a superset of traditional data management functionalities, incorporating automated data pipelines, data governance, and quality assurance mechanisms. By offering tailored data services through standardized APIs, it ensures seamless data preparation, provisioning, and management to meet the diverse requirements of 6G-enabled applications and services.
Key take-aways	Measurable KPIs: Data pipeline efficiency (e.g., latency, throughput), data quality metrics, operational cost optimization Non-measurable KPIs: Data accessibility, seamless integration, developer satisfaction DataOps contains mechanisms for privacy preserving data and model aggregation. Combining differential privacy with secure aggregation techniques allows that reliable statistical information

[2] https://zenodo.org/uploads/14616256

	of aggregated UE data can still be learned without revealing personally identifiable information, neither to network nor application provider. A framework is presented for providing data in a heterogeneous 6G network slicing environment. Alignment with 3GPP and Figure 3-2 standards ensures interoperability and adoption of standardized data operation practices. DataOps can be used to gain insights of the 5G/6G Core network performance, by supplying Machine Learning models with data necessary for Failure Prediction.
Requirements	Complexity of data operations to address heterogeneous use cases and data requirements (e.g., for preprocessing constraints, real-time processing, and compliance needs) Need to regulate data ownership and access in multi-stakeholder scenarios, ensuring secure sharing and usage of data assets across federated environments Requirement for defining new APIs, protocols, and standards for seamless data lifecycle management and integration with existing 6G architectures, addressing the current lack of standardization Necessity to manage distributed and federated data pipelines, ensuring scalability and reliability across the edge-to-cloud continuum The addition of DataOps tools, such as those supplying data to ML models, shall not adversely impact the performance of 5G/6G Core network
Standard relations & regulations	3GPP TS 28.105 ETSI GS ZSM 004 3GPP TR 28.858 phase 2 3GPP TR 28.858

3.1.2 MLOps

3.1.2.1 Introduction

MLOps is a key enabler for managing the entire ML development lifecycle, including data preparation, model training, deployment, and monitoring. The MLOps enabler encompasses a set of techniques, mechanisms, and solutions, specifically designed to efficiently manage distributed AI functions across the network, aiming to minimize the communication, computation, storage, and energy costs during data collection, training, and inference.

In more detail, in Section 3.1.2.2, a significant focus is placed on i) cross-domain training and inference that eliminates the need to transfer datasets between distributed nodes, ii) model generalization for multi-task learning, and iii) model layer offloading when computational and energy resources of the device performing the training are limited. The previous deliverable [HEX224-D33] evaluated the performance gains achieved in reducing computation and memory costs in split learning applications, while the particular emphasis of this deliverable is on the energy consumption gains introduced when applying the aforementioned techniques for distributed model training and inference is placed.

Furthermore, this deliverable introduces distributed solutions for radio and compute resource allocation to enable efficient FL-based ML training. Specifically, Section 3.1.2.3 summarizes different radio resource allocation solutions for wireless hierarchical FL networks, evaluated in terms of the achieved communication time and energy consumption between the distributed clients and the central entity coordinating the FL procedure, as well as the achieved global model accuracy. Complementary to this, Section 3.1.2.4 introduces a pricing and computing resource allocation approach for multi-server multi-model FL, which is subsequently evaluated in terms of computing resource utilization across the distributed clients and the achieved FL model accuracy.

Apart from optimizing communication, computing, storage, and energy costs, MLOps also encompasses the design of protocols and applications. In more detail, previous work in [HEX224-D33] characterized UE data privacy-sensitivity levels to define data-sharing models. Building on this, Section 3.1.2.5 of this deliverable proposes a cooperative learning framework for 6G architecture, introducing cooperative interactions for privacy-preserving training and inference using hierarchical FL across UE groups. Additionally, Section 3.1.2.6 presents a realistic FL platform, enabling privacy-preserving and distributed ML training across city verticals. Finally, Section 3.1.2.7 explores failure prediction in the 5G/6G core network using the 5G Core Observability Platform, offering key insights for implementing an AI/ML-specific view.

3.1.2.2 Distributed Model Training and Inference

One of the ML enabler technologies for MLOps is Split Learning (SL) based Vertical Federated Learning (vFL). The ML enablers should ensure ML models to be accurate, energy-efficient, reusable, generalized, modular, and flexible. In this study, the possibility of reaching above properties with SL is investigated. In the scope of component PoC #B.2 [HEX224-D33] (which is described in details later in Section 3.4.2), three main technical properties are demonstrated: i) cross-domain training and inference to improve model accuracy, without necessitating to move dataset between distributed data nodes; ii) model generalization that learns common representation of data to serve multi-tasks; iii) model layer offloading to reduce computation overhead. This deliverable deep-dives into model layer offloading with the focus on energy consumption. The distributed model architecture is illustrated in Figure 3-6, and it consists of split Neural Network (NN) models deployed in the network and in the application. The advantages of having such model architecture are privacy preservation, removing the need for model and feature sharing between the participating entities, reducing computation and memory cost as described earlier in [HEX224-D33]. In this section, the goal is to quantify the *energy consumption* in model training and inference in SL. In the presented scenario, network and application are collaboratively training a QoE estimation model, where one part of the NN model is in the network and the other is in the application. The application (referred to as output consumer) is assumed to be running on a battery-powered user device, and the network (which hosts the generalization node) is connected to an electric grid.

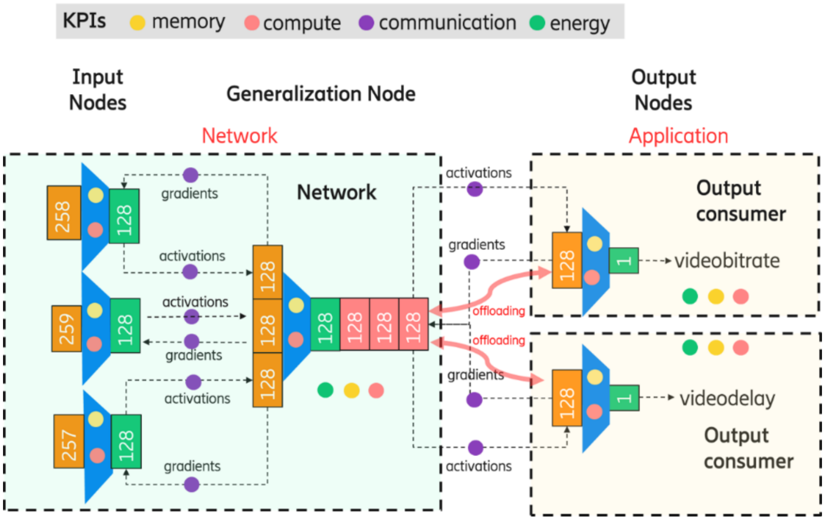


Figure 3-6: Distributed model architecture of component PoC #B.2 [HEX224-D33].

In addition to the measurement points to quantify memory allocation, computing availability, and communication cost, measurement points for quantifying the *energy* consumption (as depicted as green dots in Figure 3-6) are also deployed at the generalization and the output nodes. This helped to quantify energy consumption variation caused by offloading NN model layers from an output node, e.g., application to the network.

In order to quantify energy savings, there is a need for an energy consumption estimation tool. The energy consumption of a Kubernetes pod while executing ML tasks with NN models is measured via the Kepler tool [ACC+23]. This requires installation of the Kepler tool on every Kubernetes cluster where a measurement of energy consumption is intended. An alternative way is to collect measurements from Kepler tool while running various experiment of different settings, and then training an ML model with the collected data to obtain an energy consumption estimator. Then this obtained model can be deployed within the intended ML applications including both the network and the application. Experiments were performed on different scenarios where each scenario had different parameters.. The corresponding energy consumption measurement values are recorded.

These parameters are batch size of the dataset fed into the NN model, number of rounds of training, number of input attributes that are fed into the NN model, number of layers of the NN model, number of neurons per layer, and total model parameter size (i.e., number of weights in the NN model).

Once the measurement data is obtained, then an Extreme Gradient Boosting (XGBoost) machine learning model is trained on this dataset such that the model learns to estimate energy consumption of an NN model related to its computation overhead given different parameters mentioned earlier. The estimation performance of the trained model is evaluated on the test set, an R2-score of 0.7. Figures 3.6, 3.7, 3.8, and 3.9 illustrate the impact of number of layers in the NN, number of rounds, number of input attributes, batch size, and number of neurons at the intermediate layer of the NN models on the energy consumption. The dots, and the lines represent the actual and estimated values, respectively. In overall, the energy consumption increases linearly with the number of layers and model parameters of the NN model. This clearly motivates the need for more attention for developing shallow models or models with low number of parameters. Of course, such motivation should also be supported by targeting high accuracy. In Figure 3-7 neuron count in the intermediate layers was set to 128, number of input attributes was set to 128, and the batch size was set to 256. Figure 3-7 depicts that the energy consumption during training increases approximately linearly with the number of rounds. This motivates us to improve the vFL model such a way that it converges to a high accuracy in minimum number of training rounds. In Figure 3-8, the impact of number of input attributes is illustrated. Reducing the number of input attributes is expected to yield energy savings. Therefore, the input features that are not expected to improve the model accuracy should be removed. There are ways to autonomously select the contributing features in the context of vFL such as in [Ick23]. Figure 3-9 presents the impact of batch size on the energy consumption with varying number of NN layers. Larger batch size increases the energy consumption due to the increased number of operations in the computation units, on the other hand it helps to obtain faster convergence. Therefore, a good trade-off between the two (energy vs accuracy) needs to be established. In Figure 3-10, the impact of number of neurons at the intermediate layers on the energy consumption is visualized. Observe that y-axis is in logarithmic scale. The significant reduction in energy consumption with reduced number of neurons indicates that smaller sized NN layers helps to reduce energy consumption, which may come with a trade-off in accuracy. Again, a trade-off needs to be well managed depending on the goal and prioritized KPIs in model training.

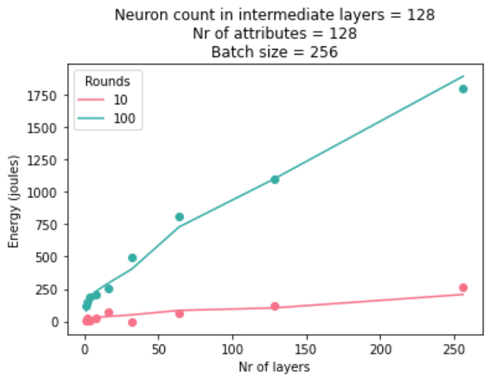


Figure 3-7: The impact of the number of layers training rounds on the energy consumption.

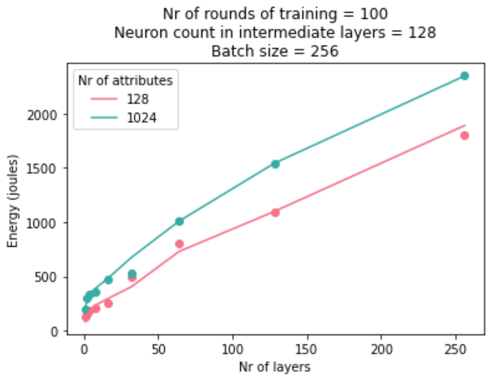


Figure 3-8: The impact of the number of layers and attributes on the energy consumption.

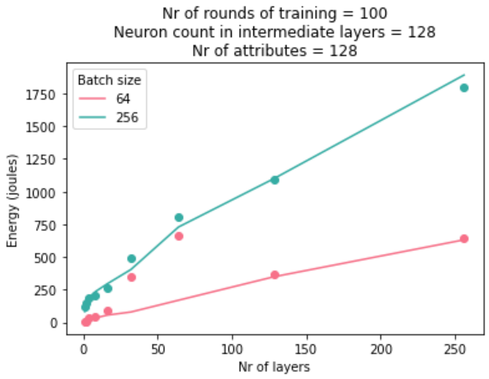


Figure 3-9: The impact of the number of layers and batch size on the energy consumption.

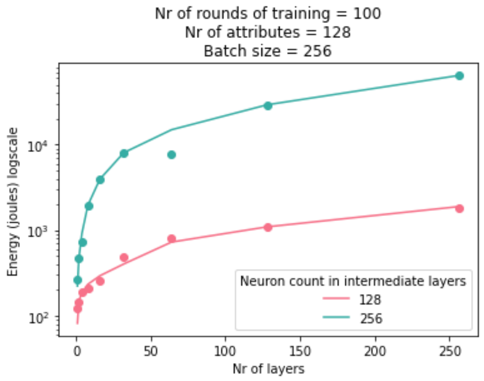


Figure 3-10: The impact of the number of layers and neuron count at the intermediate layer on the energy consumption.

Moreover, the impact of number of layers on the energy consumption during training and inference is observed to be different due to the missing backward propagation step in the inference. This is illustrated in Figure 3-11.

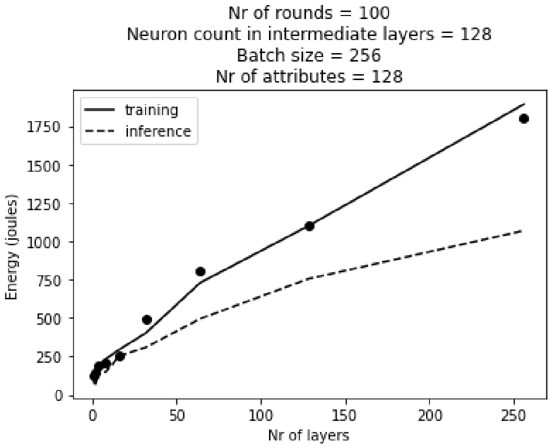


Figure 3-11: Impact of number of layers on the energy consumption is illustrated in training and inference phase.

In the above, impact of model hyper parameters and other factors on the energy consumption related to the computation overhead of NN model is quantified. Moreover, an ML model is trained to estimate energy consumption from the parameters. This pre-trained energy estimation model is deployed on the generalization node, and the output nodes. The properties of the NN model (number of layers, number of neurons, number of input attributes, batch size) are fed into the pre-trained energy estimation model to obtain the approximate energy consumption values. More details on the component PoC #B.2 is given in Section 3.4.2.

3.1.2.3 On the accuracy-energy tradeoff in wireless hierarchical federated learning

In the context of MLOps, the previous deliverable [HEX224-D33] examined the impact of radio resource allocation on the accuracy and overall performance of wireless Federated Learning (FL), as a key technique for distributed AI/ML. Specifically, it addressed the joint problem of user association and uplink transmission power allocation to multiple edge servers in a wireless Hierarchical FL (HFL) setting. The problem was modeled as a Game in Satisfaction Form, enabling each user to autonomously balance the tradeoff between local model accuracy and the time and energy overheads incurred by wireless transmissions during the HFL process. Deliverable [HEX224-D33] provided a detailed problem description, proposed a Game Theory-based solution, and presented initial numerical results demonstrating the effectiveness of optimized radio resource allocation in improving FL accuracy.

This deliverable further examines the effectiveness and efficiency of the proposed solution based on Games in Satisfaction Form, with focus on comparing it to other equilibrium concepts from Game Theory to promote sustainable MLOps, while maintaining a good tradeoff between global model accuracy and time and energy overheads. In more detail, the following comparative scenarios are considered:

- Satisfaction Equilibrium (SE) – Each user satisfies its targeted accuracy-time-energy tradeoff value.
- Minimum Efficient Satisfaction Equilibrium (MESE) – Each user satisfies its targeted accuracy-time-energy tradeoff value with the minimum possible energy cost.
- Random Association – The users randomly select an available edge server to associate with and a power level that lies within their maximum transmission power budget.
- Closest Server – The users are associated with their closest available edge server and their transmission power is selected as the SE of the game in satisfaction form that is similar to the proposed framework, with the difference that only one variable (i.e., the power) is optimized.
- Closest Server MESE – Similar to Closest Server scenario with the difference that the MESE point is selected for the users' uplink transmission power when playing the game in satisfaction form.

The simulation setup used for experimentation is as follows. A wireless HFL network is considered, arranged within a circular area of 300 m radius. 3 edge servers are randomly positioned along the perimeter of the circular area, and 10 user devices are uniformly distributed within the area, unless otherwise explicitly stated. The total network bandwidth is 10 MHz and the maximum transmission power of each user device is 1 W. For the implementation of the HFL procedure, we consider 6 local model updates, 10 edge model updates, and 100 overall HFL iterations. The training learning rate is 10^{-3}, and the data size of the local model parameters is 28.1 kbits. The Modified National Institute of Standards and Technology (MNIST) dataset is used, and the 60000 total samples are equally divided among the users following in a non-Independently Identically Distributed (non-IID) manner by ensuring that no individual user possesses samples of all classes included in the dataset. More details about the problem, solution, and numerical evaluation can be found in [CDP24].

First, the performance of the proposed satisfaction game-based framework is investigated in terms of the tradeoff achieved between the targeted quantities (network entropy impacted by the user-to-edge-server association, energy consumption, transmission time). The results in Figure 3-12 and Figure 3-13 reveal that the proposed framework under both types of equilibria (i.e., SE and MESE) attains the highest model accuracy (approximately 80%-90%) among all alternatives, confirming its superior performance in the learning process. The HFL procedure's convergence is achieved after approximately 20 iterations, with the accuracy gradually reaching its final value. Once again it is highlighted that the MESE point sacrifices concluding the best possible accuracy for the sake of its energy cost minimization objective, yielding a lower accuracy of about 10%, outperforming however all the rest benchmarking scenarios in terms of the achieved accuracy. Concerning the networks' entropy, it precisely follows the same trend as the global model's accuracy in Figure 3-12, verifying that capturing the user data distribution across the different edge servers constitutes a representative metric for the achieved global model's accuracy. The lowest total network cost and energy consumption are achieved under the proposed framework using MESE and Closest Server MESE, which also employs MESE point for uplink transmission power allocation. Among these, the Closest Server MESE further reduces energy consumption by assigning users to the nearest edge server but at the expense of reduced global model accuracy.

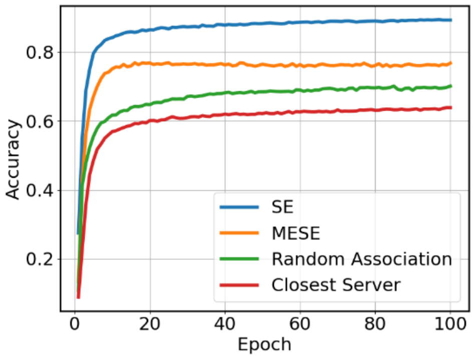


Figure 3-12: Achieved trade-off in terms of global model's accuracy under the SE, MESE, and the benchmark scenarios.

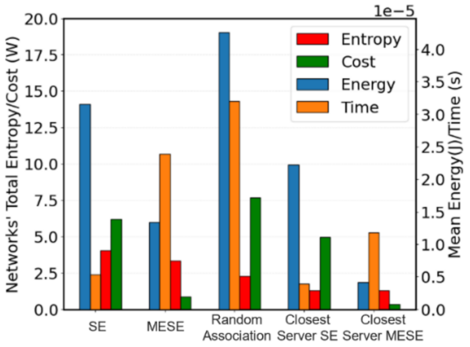


Figure 3-13: Achieved trade-off in terms of network's total entropy, cost, mean consumed energy, and time under the SE, MESE, and the benchmark scenarios.

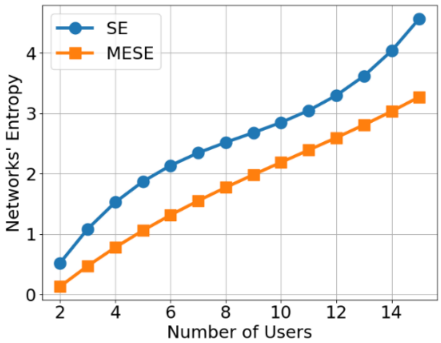


Figure 3-14: Achieved network's entropy under SE and MESE points for different numbers of users.

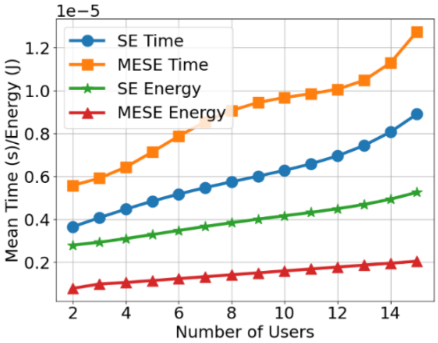


Figure 3-15: Achieved mean consumed energy and time under SE and MESE points for different numbers of users.

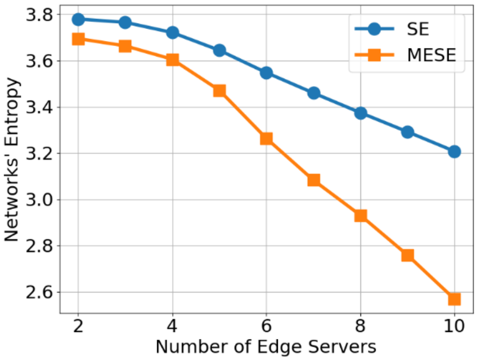


Figure 3-16: Achieved network's entropy under SE and MESE points for different numbers of servers.

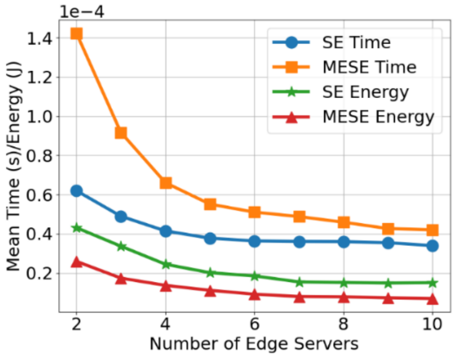


Figure 3-17: Achieved mean consumed energy and time under SE and MESE points for different numbers of servers.

To further examine the impact of the number of network entities in the studied framework in terms of both end users and edge servers, we conducted a scalability evaluation. Specifically, Figure 3-14 and Figure 3-15 present the behavior of the framework as more users are added to the network, ranging from 2 to 15 users. An increase is shown as more users enter the HFL procedure owing to the inclusion of more (and potentially unique) data samples during model training. Nevertheless, apart from the increase in the network's entropy, an increasing trend is also observed in the mean energy consumption and transmission time of the users as their number gets higher under both SE and MESE. Figure 3-16 and Figure 3-17 regard the same network metrics, considering however the case that the number of edge servers increases instead, ranging between 2 and 10 for a fixed number of users equal to 10. In this simulation case, the trend is entirely the opposite. The presence of a larger number of edge servers implies that the same amount of data samples is shared among more servers, which prevents the network from reaching an Independent Identically Distributed (IID) data distribution case.

Overall, the scalability analysis reveals that increasing the number of users enhances model diversity but leads to higher energy consumption, while increasing the number of edge servers improves energy efficiency but introduces challenges in achieving IID data distribution.

3.1.2.4 Incentive mechanism design for wireless federated learning networks

The allocation of computing resources is a critical factor to enable efficient distributed ML/AI services within the context of MLOps. In the previous deliverable [HEX224-D33], the problem of incentivizing distributed clients via monetary or resource-related rewards to participate in FL and practically investing their computing resources for local training was considered and discussed. The computing resource allocation problem was formulated as a Fisher market [BCD+14], where servers act as buyers and the clients' computing resources represent the goods being purchased. The servers aim to create models by leveraging the data and computing resources of the clients. Clients, who possess locally available data, independently train models on their devices. These locally trained models are then aggregated and coordinated by the servers to produce the final models in the multi-server FL process. Each server relies on fractions of the clients' computing capacity, allocated specifically to support this collaborative model training. The market determines prices for each client to compensate for the local model training they provide, while also considering the budget constraints of each server. As a result, the formulated market is characterized by a set of constraints that govern both the resource allocation and the pricing. The computing resource allocation from clients to servers is subject to two key constraints: the clients’ total computing power and the servers’ monetary budgets. By modeling the problem as a Fisher market, the solution concept reached is the so-called Market Equilibrium (ME) point. At this point, servers maximize their utility, and the market achieves clearance, meaning all clients’ computing resources are fully allocated, and the servers’ budgets are entirely utilized. Any redundant resources remain unpriced. Finally, the servers’ utility is expressed as a linear function of the resources allocated by the clients and the resulting model accuracy, which improves as resource allocation increases, since more data samples are processed by the clients within a given time frame.

In this deliverable, the aim is to evaluate the behavior of resource allocation and pricing under the ME. To this end, a simulation setup with 10 clients is considered, each characterized by a different number of local training samples and total computing resource capacity, represented by the client ID. A higher ID indicates a greater amount of data and computing resources. First, in Figure 3-18, the pricing of the respective client is illustrated as a function of the client ID in the horizontal axis, verifying that clients investing more effort in terms of training samples and computing resources to the overall multi-model FL process are priced more. To further examine how clients allocate their computing resources for training models across different servers, Figure 3-19 analyzes three distinct budget ratio scenarios from the perspective of the servers. Considering 4 servers denoted as S1, S2, S3, S4, the budget ratio refers to the relative budget availability of each server compared to server S1. For example, a budget ratio of (1/2/3/4) implies that the budget of server S2 is twice that of server S1, the budget of server S3 is three times the budget of server S1, and the budget of server S4 is four times that of server S1. Across the three budget ratio cases examined, Figure 3-19 illustrates the equilibrium resource allocations as a function of the client ID along the horizontal axis. The results show that servers with higher IDs and larger budgets can afford to engage more clients, ensuring greater data heterogeneity for improved global model accuracy and faster training times, reflecting their model's criticality. By varying the budget ratios of servers, specifically increasing the budget of server S3 while decreasing that of server S4, we demonstrate that the proposed model effectively prioritizes service allocation, where servers with higher budgets receive resources from more clients.

The proposed computing resource allocation and pricing solution based on the ME is further compared against the following resource sharing schemes from the literature:

- Max-Min Fairness: The utility of the lowest utility server is maximized while ensuring the resource allocation respects the clients' total computing power constraints.
- Proportional Sharing: Each server is allocated a fraction of each client's computing resources proportionally to the server's budget.
- Max Social Welfare: The total utility of all servers is maximized subject to the clients' total computing power constraint, neglecting budget constraints.
- Equal Sharing: All servers are allocated an equal fraction of each client's computing resources.

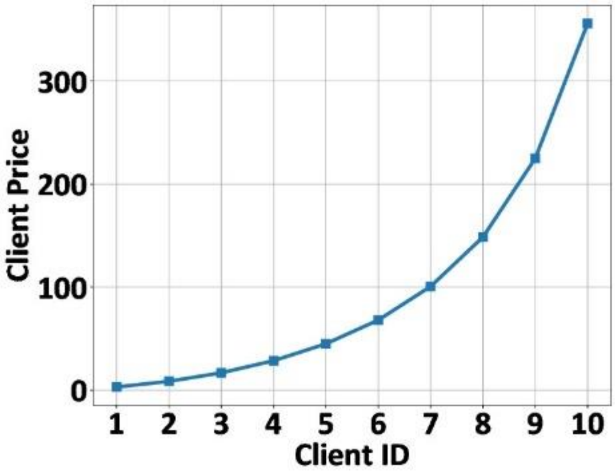


Figure 3-18: Equilibrium prices for clients with increasing training data and computing resources.

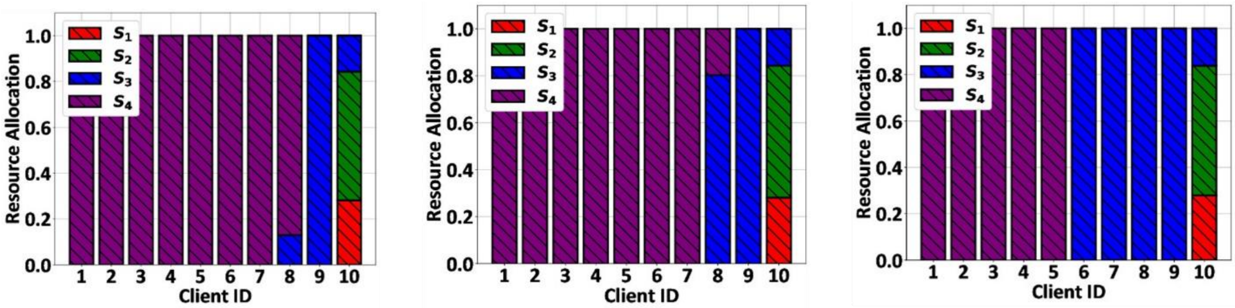


Figure 3-19: Equilibrium resource allocations for clients with increasing training data and computing resources.

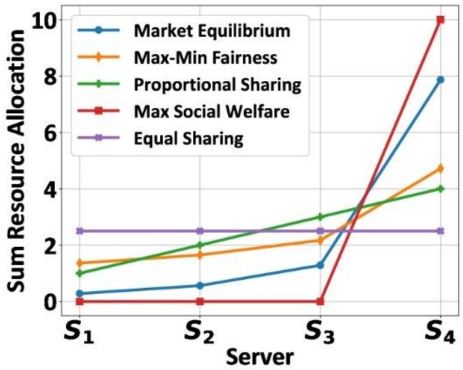


Figure 3-20: Equilibrium resource allocations under comparative scenarios.

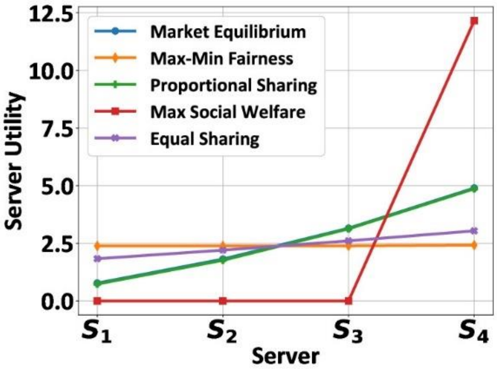


Figure 3-21: Server utilities under comparative scenarios.

In the experiment, a baseline budget ratio of (1/2/3/4) between the servers is considered, along with 10 clients, each having increasing values for both data and computing resources, as previously described. Figure 3-20 and Figure 3-21 illustrate the total resource allocations from different clients to the various servers along the x-axis, and the server utilities achieved while accounting for the various comparative scenarios, respectively. As expected, the maximum average utility is achieved in the Max Social Welfare scenario (Figure 3-21), albeit at the cost of highly uneven resource allocations from the client side (Figure 3-20), failing to fairly respect the service criticality compared to the rest of the scenarios. Equal Sharing, Proportional Sharing, and Max-Min Fairness do not explicitly account for budget constraints and model criticality. Therefore, although the proposed ME framework performs approximately close to these approaches in terms of server utility and achieved model error, it effectively allocates resources on the client side to align with the servers' model criticality, ensuring that each server's budget is fully utilized.

3.1.2.5 High-level procedures and signalling for cooperative learning

Cooperative learning is an ML-based cooperative intelligence technique, where network nodes collaboratively share data and/or models towards achieving a common task, by taking advantage of each other's knowledge and experience. The learning task requires participation of multiple nodes at the same time and needs multiple successive steps. The potential scenarios may include cooperative rendering, where each UE shares partial result to the aggregator that combines them and shares the full result back to the UEs.
Based on the privacy-aware data classification and trust levels between UEs and network, UE data is shared either partially or totally or owned locally. The characterization of UE data privacy-sensitivity levels that determines the data sharing cooperative model is introduced in [HEX223-D33].

Here, the cooperative learning in 6G architecture is proposed by introducing cooperative session concept to enable:

- Training over a distributed network of UEs, grouped in cooperative groups, e.g., geographically in subnetworks, that are collaboratively learning a model in a privacy preserving manner.
- Cooperative inference over multiple (groups of) UEs given a pre-trained model.

To enable a privacy preserving model training, UEs may adapt a federated learning (FL) framework in which UE data remains locally, while in each training iteration the UEs share only the model parameters (gradients) with the model aggregator at the network side. However, the model parameters may still expose sensitive information of training data to the outside adversaries, i.e., by means of gradient-based inversion attacks. Therefore, instead of a classical centralized FL, a privacy preserving hierarchical FL can be utilized. Here, the UEs create cooperative groups based on different criteria such as geographical distribution, task requirements, privacy levels, etc. Each cooperative group has a group head (GH), which aggregates the intra-group results and shares the cooperative group aggregated results (like model parameters) with other cooperative groups through the NW.

Assuming that the trust level among UEs within cooperative group excludes the possibility of potential adversarial attacks, the model can be trained using FL among UEs in cooperative group, aggregating the results/model at GH. GH(s) can share this aggregated data with NW, by using privacy-preserving aggregation, e.g. by using Prio-protocol [CB17], by splitting the aggregated data into shares, such that NW obtains the aggregated data. The potential realization of privacy-preserving (both data and model) aggregation procedure based on Prio protocol in NWDAF framework and corresponding high-level message exchange is introduced in Section 3.1.1.3.

The cooperative groups are not necessarily formed based on distances using proximity-based services (e.g. SNs), but they can be formed based on privacy levels or the type of environment, e.g., in a collaborative remote gaming session, creating collaboration groups. Here, network can help with the creation of collaboration groups based on the information shared by the UEs such as UE communication and computation capabilities and their privacy budget.

UEs register to NWDAF by sending its *data privacy levels, cooperation budget and computing resources*. The establishment of the cooperative session can be requested by an application via AF that acquires the list of UEs with capability or are willing to cooperate (this can be signalled from UE to AF in an individual UE PDU session by setting the CooperationFlag). The application then requests a cooperative training/inference session from the network by sending to NWDAF the *CooperationRequest* containing *TaskID*, *ModelID*, *number of iterations, number of participating UEs/list of desired UEs, cooperation group characteristics (same location, privacy budget), list of desired UEs to share result with*, and *update flag for NWDAF*. In this way, the training/inference service will be provided by the network directly so the application will just get the result. NWDAF then decides on the formation of cooperative group(s) based on certain criteria, e.g., by using established local subnetworks (if location dependent). NWDAF also chooses group members and GHs, which would perform aggregation of inferred data. Based on application (AF) characteristics and selected cooperative group, NWDAF requests Cooperative session establishment from PCC / Session Management Function (SMF) by sending CooperationSessionEstablishmentReque***st*** containing *CooperativeSessionID* (related to the *Task ID* from the CooperationRequest), *list of IPs of selected cooperative UEs*, and *required QoS*. Single *TaskID* could be also associated with multiple Cooperative Sessions.

Individual PDU sessions of cooperating UEs are associated with CooperativeSessionID, such that the modification of cooperative session could change individual PDU sessions of UEs participating in the Cooperative session. In this way, the established Cooperative session offers/enables the service of providing cooperative learning results.

The cooperative session can be also requested by a UE, such that NWDAF determines the cooperative group of UEs, GH, and cooperative session characteristics based on registered UEs and available network resources, similarly to the response to AF request.

Cooperative learning

The high-level message exchange for cooperative learning is shown in Figure 3-22. Leader (AF1) and Helper (AF2), serving as leader and helper aggregators, and NWDAF, acting as the collector, are registered to NRF. AF3 requests the cooperative training task and sends training CooperationRequest via Network Exposure Function (NEF). GH UE can subscribe to cooperative learning task from NWDAF. NWDAF performs discovery of AFs from NRF and selects and subscribes to Leader and Helper(s). Based on the selected AFs, and the CooperationRequest, NWDAF sends *CooperativeSessionEstablishmentRequest* for training to PCF/SMF. After establishing the cooperative session for the participating cooperative groups, NWDAF retrieves the model from ADRF and sends it to GH. GH then shares model to cooperating UEs in the cooperating group. UEs are training the model on local data and then send model updates (gradients) to GH. The privacy is preserved under the assumption that aggregators do not contain the same model share. Before creating aggregated shares, Leader (AF1) and Helper (AF2) exchange and validate SNIP proofs to ensure validity of the data. The Collector (NWDAF) collects aggregate model shares, stores the model in ADRF and sends it to consumers.

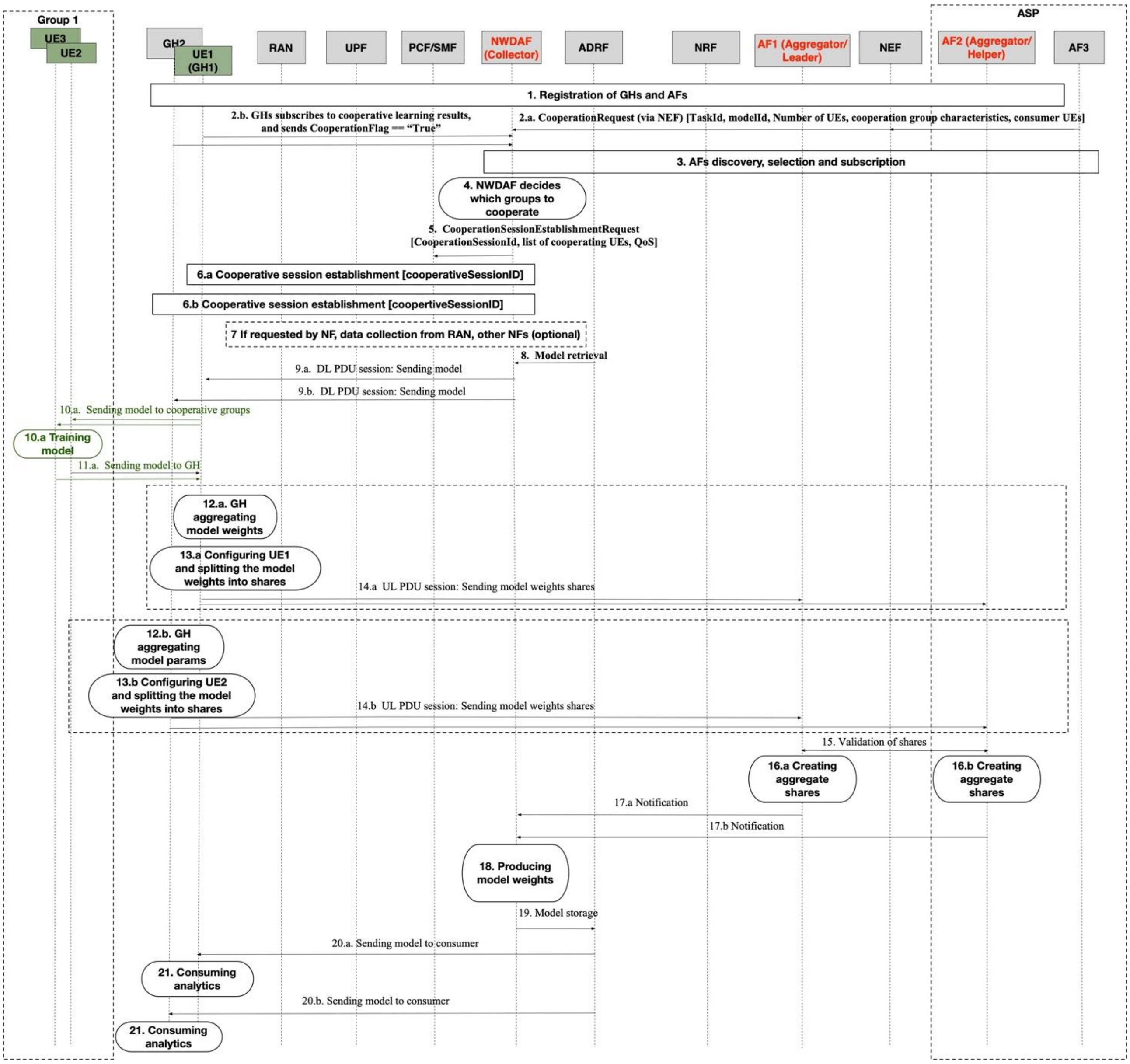

Group 1
UE3
UE2
GH2
UE1 (GH1)
RAN
UPF
PCF/SMF
NWDAF (Collector)
ADRF
NRF
AF1 (Aggregator/ Leader)
NEF
ASP
AF2 (Aggregator/ Helper)
AF3
1. Registration of GHs and AFs
2.b. GHs subscribes to cooperative learning results, and sends CooperationFlag == "True"
2.a. CooperationRequest (via NEF) [TaskId, modelId, Number of UEs, cooperation group characteristics, consumer UEs]
3. AFs discovery, selection and subscription
4. NWDAF decides which groups to cooperate
5. CooperationSessionEstablishmentRequest [CooperationSessionId, list of cooperating UEs, QoS]
6.a Cooperative session establishment [cooperativeSessionID]
6.b Cooperative session establishment [coopertiveSessionID]
7 If requested by NF, data collection from RAN, other NFs (optional)
8. Model retrieval
9.a. DL PDU session: Sending model
9.b. DL PDU session: Sending model
10.a. Sending model to cooperative groups
10.a Training model
11.a. Sending model to GH
12.a. GH aggregating model weights
13.a Configuring UE1 and splitting the model weights into shares
14.a UL PDU session: Sending model weights shares
12.b. GH aggregating model params
13.b Configuring UE2 and splitting the model weights into shares
14.b UL PDU session: Sending model weights shares
15. Validation of shares
16.a Creating aggregate shares
16.b Creating aggregate shares
17.a Notification
17.b Notification
18. Producing model weights
19. Model storage
20.a. Sending model to consumer
21. Consuming analytics
20.b. Sending model to consumer
21. Consuming analytics

Figure 3-22: Cooperative learning.

Cooperative inference

Whether the cooperation groups are formed based on distances using proximity-based services or on other criteria, such as based on UE communication and computation capabilities or privacy budgets, there can be different variants of cooperative inference, introduced in more details in Annex, where the aggregation of inferred data is performed in SN using local communications among UEs, while GH is one of the UEs in SN, and Cooperative inference in collaborative group, where the aggregation of inferred data is performed using communications over NW, while GH is one of the UEs or it is located in NWDAF.

3.1.2.6 Federated Learning approach between city verticals

Integrating the people in service of the city and having city resources in service of the people is one of the goals of the Smart City of the future. By using IoT sensors and other devices that are able to gather data (such as smartphones and smart watches), people are able to produce valuable information that can be collected, processed and used on different ML models that are the backbone of the services behind a Smart City.

As a branch of AI, ML can be understood as a way of "using data and algorithms to enable AI to imitate the way that humans learn" [IBM24]; by using a known dataset, it is possible to design an algorithm that takes this information to predict a certain result from this data. This process is further improved by using an error function that constantly evaluates the result of the assessment and an optimization function that picks better data points in the chosen dataset, if they produce a better prediction. With pre-processing the dataset and iterating the algorithm with multiple evaluation-optimization steps, it is possible to reach high values of accuracy in predicting certain events, developing what is called an ML model. This ML model can then be applied on data collected on edge devices of the city network to generate useful forecasts.

Edge Computing is a distributed computing framework that bridges the gap between applications/services and data sources. By tapping into the IoT and edge devices distributed along the network, it is possible to not only collect relevant data but also use the computational capabilities of said devices towards running and splitting specific workloads that are to be executed on the network, effectively raising the total computational potential of the network and available data sources.

FL is an approach to this distributed framework that integrates ML models and privacy-preserving features. By uploading the ML models from the central server and running them locally, raw data is never uploaded for training (only the gradient and weights of the model are uploaded), which enhances the security of the data used for ML models with privacy-preserving needs such as data related to health or social issues. FL can be done using a central server for model aggregation and global update; or serverless, by iterating the model results with global updates happening among the edge devices on the network.

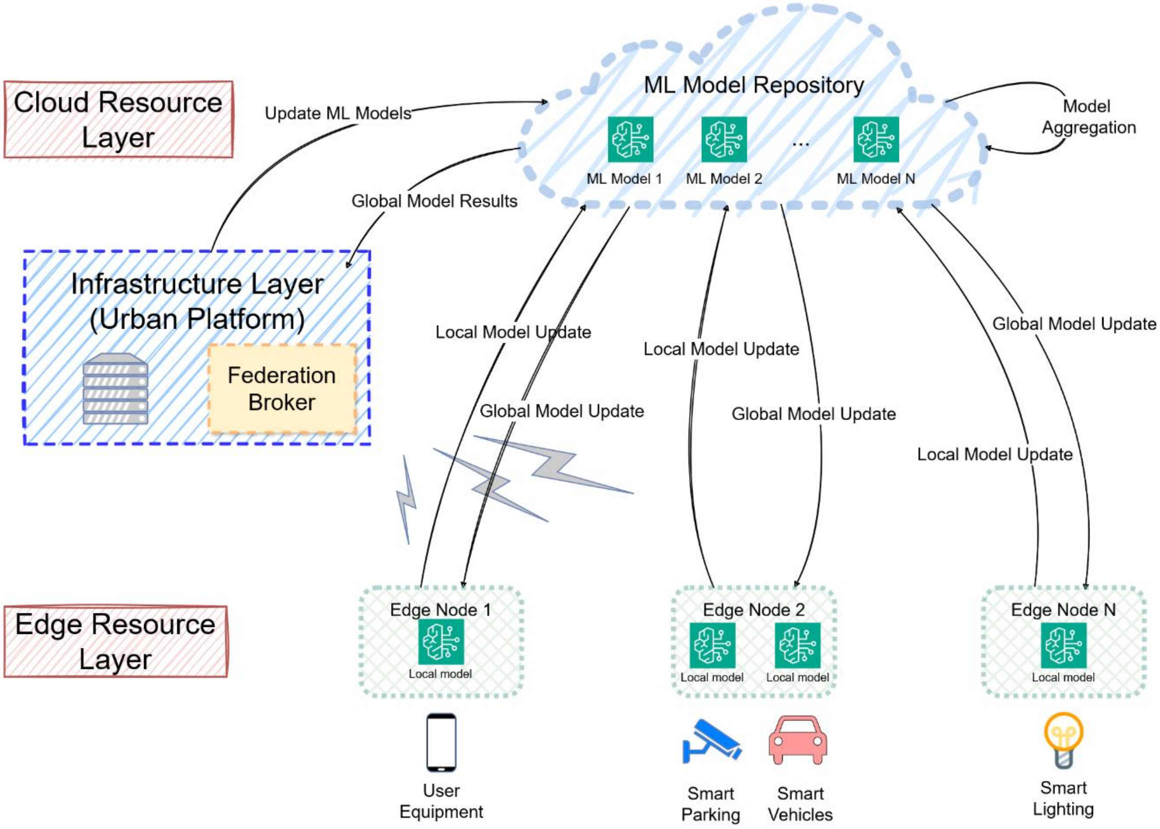


Figure 3-23: Federated Learning approach between city verticals.

This Federated Learning model approach in Figure 3-23: Federated Learning approach between city verticals. is planned to be integrated on Urban Platform [UBW24]. Using a Federation broker to control Authentication, Authorisation and Accounting (AAAs) processes, a Federation of edge devices is setup between the city verticals monitored by this Urban Platform; the data collected by these edge nodes can be used to host the training process of ML models. In doing so, it's possible to bring the benefits of Federated Learning to city monitoring, making full use of the computational capabilities of edge nodes, as well as the distributed nature of city verticals for data collection. Local Model and Global Model update communications are encrypted in transit, to ensure model training is less vulnerable to attacks on the upload of model gradients.

ML Model Aggregation is offloaded towards the Cloud, making full use of the hybrid resource pool available in order to reduce infrastructure costs, and trained ML models are hosted on a Model Repository, that can be accessed through the Urban Platform. This ML Model Repository can be accessed by multiple cities using the Urban Platform, extending the benefits of the trained ML models while preserving the privacy of the data collected on the original city.

The FL solution is not without its issues: imbalanced data collected from edge devices can imply more rounds of training to reach an acceptable level of prediction accuracy; likewise, non-Identically and Independently Distributed (IID) data can generate some bias for certain datasets collected by edge devices, that are not able to be balanced locally, but only by training the model multiple times along the network. Besides these issues regarding the data collected, the infrastructure itself can be a bottleneck, as connecting massive amounts of edge devices with different computational capabilities poses a challenge algorithmically: to tackle workload splitting and model training on these different devices, while similarly handling the limited amount of bandwidth can also be a bottleneck due to its impact on latency times in the communication between devices. Solutions to these issues range from optimizing computational costs of the federation network [SFU24], optimizing bandwidth allocation [LJZ+22], and enhancing communication efficiency and training time [JW24]. The concept of Smart City taps into IoT devices to collect relevant data that is then used by ML models in order to predict certain outcomes more accurately, in this case, on the city verticals that run the Smart City. These city verticals affect most sectors of the infrastructure, such as:

- transportation systems, with smart public transportation and smart vehicles allowing for Mobility as a Service (MaaS) options for the customer, as well as communication between vehicles and traffic data collection allowing for more accurate traffic flow prediction, which optimizes routes that vehicles take, minimizing travel time and travel costs;
- agriculture, with smart monitoring solutions of real time data like weather parameters and soil contents, which can be used to predict harvesting timings and provide guidelines for a better and more plentiful yield;
- energy, with smart energy management enabled by a correct prediction of energy expenditure from the data collected from smart homes and city structures;
- health, allowing for more detailed research, as privacy-preserving patient data enables the collaboration between infrastructures, which creates more models able to detect and prevent diseases and other health conditions;
- industry, using cobots that collect data and communicate with the production line structure in order to optimize efficiency while also minimizing worker risks and raising the fault tolerance of the systems in place.

A common benefit of FL amongst these sectors of the Smart City is to improve efficiency and reliability. To fulfil this requirement, the current FL toolkit must be improved in order to automate the design and maintenance processes of ML models that match up with the ever-changing needs of the city and its inhabitants. This is where MLOps begins to take shape: borrowing from the concept of DevOps, stemming from Development and Operations, it amounts to, according to Microsoft, "the union of people, process, and technology to continually provide value to customers" [Mic24], which translates to the supporting operations of application development, throughout the planning, developing, delivery and Lifecycle Management (LCM) of these applications. In this sense, MLOps encompasses the same philosophy but towards the development process of ML models and services, such as data preparation, monitoring, deployment and following LCM of the ML models on the network. This can be applied on the FL approach: the collaborative and decentralized nature of FL allows privacy-preserving considerations on top of the MLOps toolkit.

These LCM processes, such as the Continuous Integration and Continuous Delivery (CI/CD) of updates towards the ML models hosted and trained on our framework, is monitored and kept by our Urban Platform. Zero-touch mechanisms that automatically manage the lifecycle of model uploads, model aggregation and model results, as well as manual intervention from local authorities can provide the necessary management requirements in order to extend the MLOps practices towards our Federated Learning framework.

3.1.2.7 Failure prediction in 5G Core – experimental results

With the emergence of new use cases of 6G networks (such as from robots to cobots, massive twinning, immersive telepresence for enhanced interactions), reliability and high uptime remain paramount. Therefore, in this contribution the main focus is failure prediction in 5G/6G Core network. For this purpose, a series of experimental studies using the 5G Core Observability Platform (called 5GCOP here) was performed. Based on the obtained results, key takeaways for the implementation of AI/ML Specific View for 5G/6G Core were formulated, in particular for the implementation of the failure prediction use case in the 5G/6G Core.

Figure 3-24 shows the process of predicting failures using ML. First, the raw data is collected and transformed into a data set in the form of a csv file. Figure 3-24 on the right shows the operation of the ML algorithm integrated with AIaaS (see Section 3.1.4).

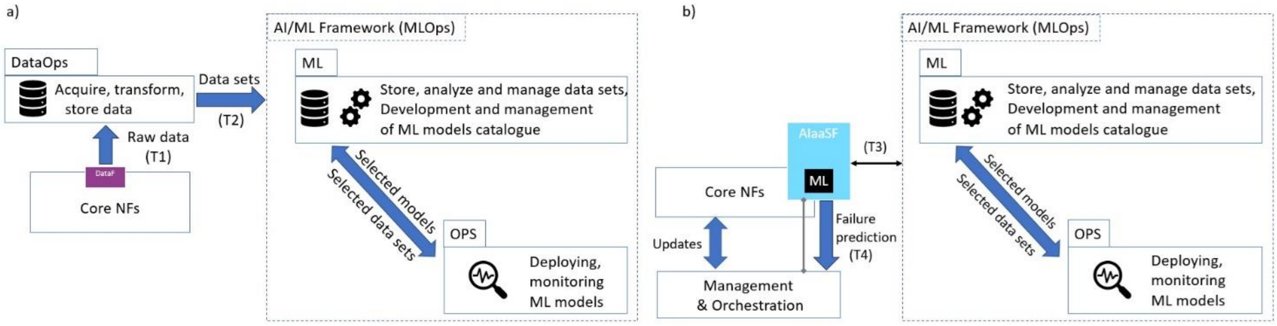

a)
AI/ML Framework (MLOps)
DataOps
Acquire, transform, store data
Data sets
(T2)
Raw data
(T1)
DataF
Core NFs
ML
Store, analyze and manage data sets, Development and management of ML models catalogue
Selected models
Selected data sets
OPS
Deploying, monitoring ML models
b)
AIaaSF
ML
(T3)
Core NFs
Updates
Failure prediction
(T4)
Management & Orchestration
AI/ML Framework (MLOps)
ML
Store, analyze and manage data sets, Development and management of ML models catalogue
Selected models
Selected data sets
OPS
Deploying, monitoring ML models

Figure 3-24: Specific view of process for failure prediction in 5G/6G Core.

The data sets focused on measuring delays between pairs of network functions of the 5G Core, collected by the Linked Service Mesh (see section 3.1.1.4), were leveraged. The time of data processing (T1 - collecting raw data, T2 creating data sets, T3 - ingesting data set by a Machine Learning model) and the ML algorithms execution with preprocessing (T4) were measured. The obtained results depending on the number of loaded data (from 1000 to 50000) are presented in Figure 3-25. As can be seen in the figure on the right, changes in T4 are at the level of milliseconds. On the other hand, T3 depends on the time required to load the csv file, in the considered case the loading is done via the pandas library function. The collection of raw data and the creation of data sets takes the longest time (T1+T2).

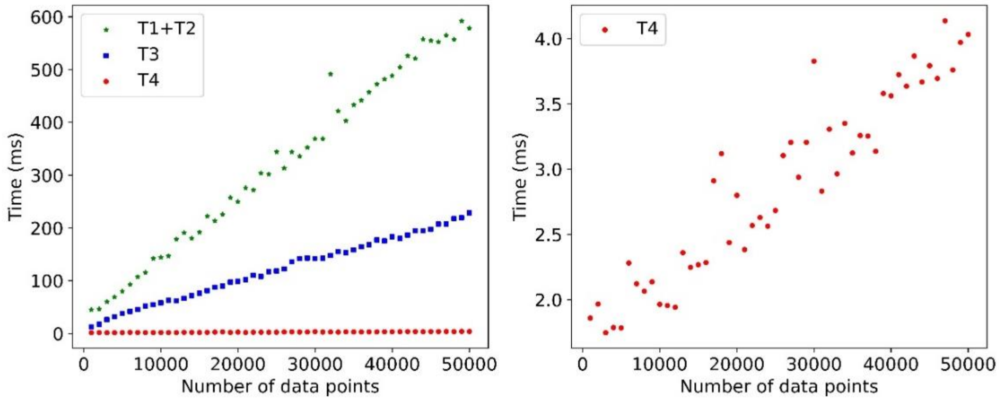


Figure 3-25: Times T1+T2, T3, T4 in the function of data points.

For failure prediction, the following the most well-known ML algorithms were analyzed: Regression Tree, Support Vector Regression (SVR), Multi-layer Perceptron regressor (MLP Regressor), and Long Short-Term Memory network (LSTM), often used for prediction in telecommunications.

As the next step, the data set containing 50 000 time series measurements of the delays between network functions was created. The 'amf-outbound-nrf-204' attribute was selected, which represents the bidirectional delay for the 204 HTTP code between “amf” and “nrf” network functions of 5G Core. This measurement represents the TTFB metric (Time to First Byte). Then, the data was randomly divided into training and test data in a ratio of 80/20.

The obtained data is summarized in Table 3-4. Figure 3-26 to Figure 3-29 present the visualization (50 000 data points) of the prediction of the 'amf-outbound-nrf-204' attribute for Regression Tree (Figure 3-26), SVR (Figure 3-27), MLP regressor (Figure 3-28) and LSTM (model under validation, Figure 3-29).

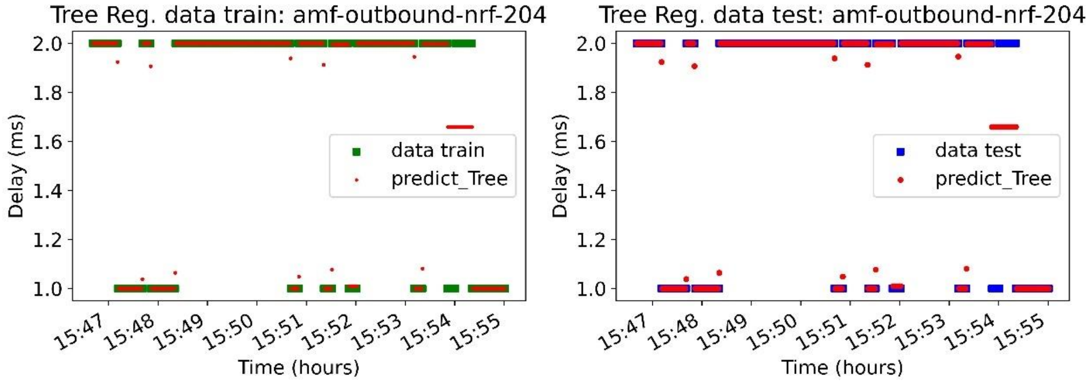


Figure 3-26: Regression Tree prediction for training and test data - 'amf-outbound-nrf-204'.

Figure 3-26 shows the prediction of the Regression Tree of training and test data for a model with the following parameters: max_depth=9, min_samples_split=2, min_samples_leaf=4, random_state=42. However, in further experiments, the regularization was increased by using the parameter max_depth=8.

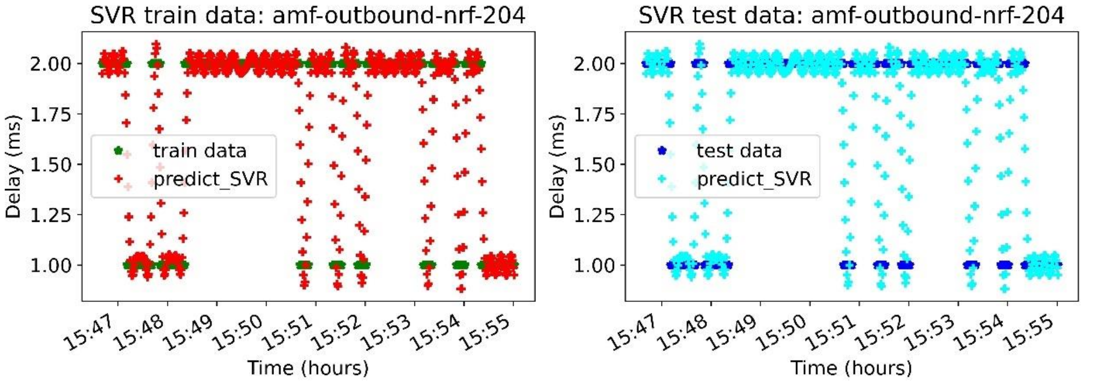


Figure 3-27: SVR prediction for training and test data – ‘amf-outbound-nrf-204’.

Figure 3-27 shows the prediction of the SVR of training and test data for the model parameters: 'C': 1000, 'epsilon': 0.05, 'gamma': 1000, 'kernel': 'rbf'.

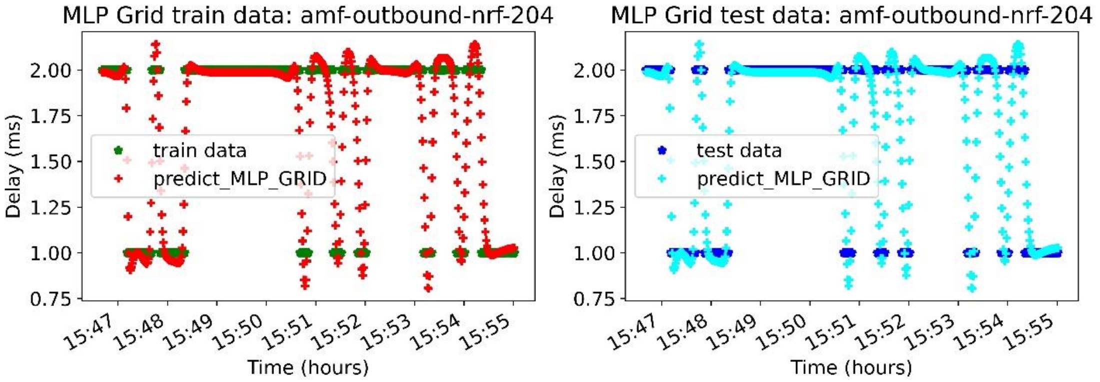


Figure 3-28: MLP Regressor prediction for training and test data - 'amf-outbound-nrf-204'.

Figure 3-28 shows the prediction of the MLP regressor of training and test data for the following model parameters: 'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (80, 40), 'learning_rate': 'constant', 'max_iter': 3000, 'solver': 'adam'.

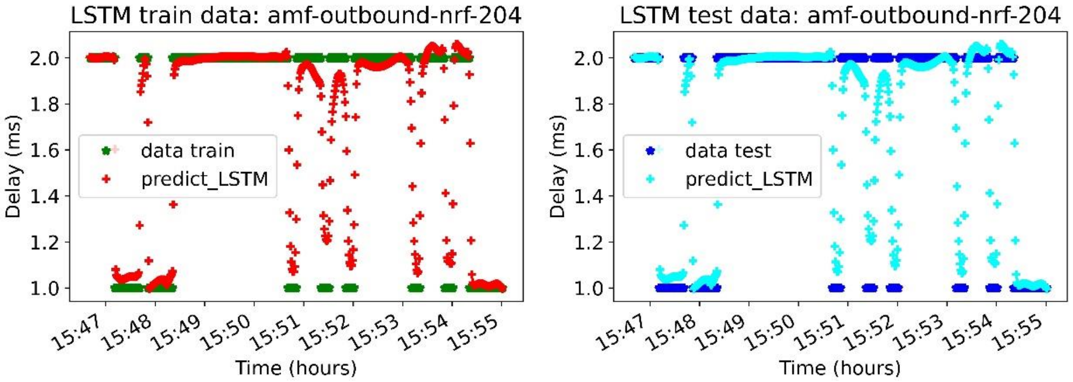


Figure 3-29: LSTM prediction for training and test data - 'amf-outbound-nrf-204' attribute.

Figure 3-29 shows the prediction of the LSTM of training and test data for the model parameters: LSTM(70)+ RepeatVector(2) + LSTM(45), Dropout(0.4), Dense(1), optimizer='Adam' (learning_rate=0.001), loss='mean_squared_error', epochs=500, batch_size=128.

Table 3-4: Prediction results for individual models.

Model	Tree Regressor	SVR	MLP Regressor	LSTM
Model fitting time (s)	0.029	validation 30252.90	validation 18543.025	validation 1715.177
Prediction R2 (R2_score) for training data (-)	0.922	0.924	0.913	0.929
Prediction MSE for training data (ms^2)	0.015	0.016	0.018	0.015
Time for prediction (Inference time) (s)	0.0019	13.57	0.033	5.266
Prediction R2 (R2_score) for testing data (-)	0.928	0.921	0.907	0.928
Prediction MSE for testing data (ms^2)	0.014	0.017	0.019	0.015
Time for prediction -T3 (s)	0.00050	3.391	0.013	1.625

Considering the shape of the waveform, the prediction time and the model fitting parameters (R2_score and MSE), decision trees achieved best results for predicting waveforms resembling squares. The model of choice for failure detection was the Tree Regressor based on the results provided in Table 3-4. Regression trees learn the fastest (fit) and process data the fastest (predict), without requiring significant hardware resources (amount of memory, number of processor cores), which allows their potential implementation in AIaaS for 5G/6G Core failure prediction. However, their excessive model fitting should be controlled by analyzing indicators such as: R2-score, MSE and F1-score. In this approach, selected regression trees require parameter tuning and data scaling. Random Forests and regression trees with gradient boosting (XGBoost) models are usually free of these inconveniences. However, these models require multiple regression trees, which is why they work slower.

As part of the experiments, a regression tree model was created with regularization allowing for the creation of a generalized model with the following parameters: Tree Regressor (max_depth=8, min_samples_split=2, min_samples_leaf=4, random_state=42). Training and prediction were performed on the same data set containing 50 000 data points of the 'amf-outbound-nrf-204' attribute. The visualization of predictions for training data is shown in Figure 3-30. The following results were obtained for the training data: Time fit Tree Regressor (s): 0.023; R2 tree train: 0.86; MSE tree train: 0.030; Time predict Tree Regressor for train data (s):0.0015.

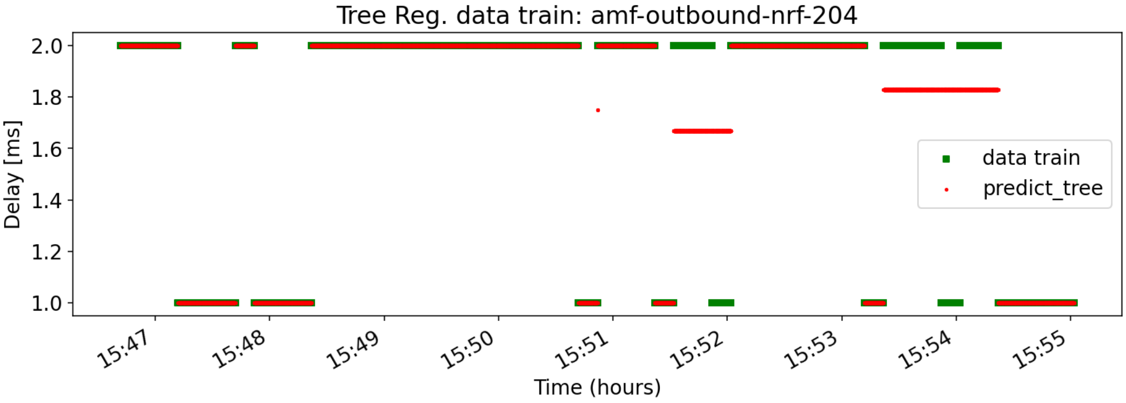


Figure 3-30: TreeRegressor model prediction for training data - 'amf-outbound-nrf-204'.

The pre-trained regression tree model was used to predict future data for a dataset containing a failure (an additional latency introduced half-way through the data set to model the failure), on which the model was not trained (Figure 3-31). It was noticed that due to the lack of prior training of the model on future data, the R2_score coefficient is always negative. For data without failures, the Mean Squared Error (MSE) value remained at the level of <1. On the other hand, for the data containing a single failure (defined as introducing additional 70 ms latency), the MSE value was over 1.20. With the increase in the number of points corresponding to delays above 70 ms, the MSE value increases rapidly, reaching the value of several thousand for 18 329 data points without failures and 9 671 recorded delay values >70 ms (failure). The failure occurs when the delay exceeds the threshold set by SLO - in our case 10 ms. For the test data (11 000) and future data 28 000 (18 329 + 9 671), the prediction which is shown in Figure 3-31, the following coefficient values were obtained: R2 tree test: 0.86; MSE tree test: 0.030; R2 tree future: -0.50; MSE tree future: 3482.09.

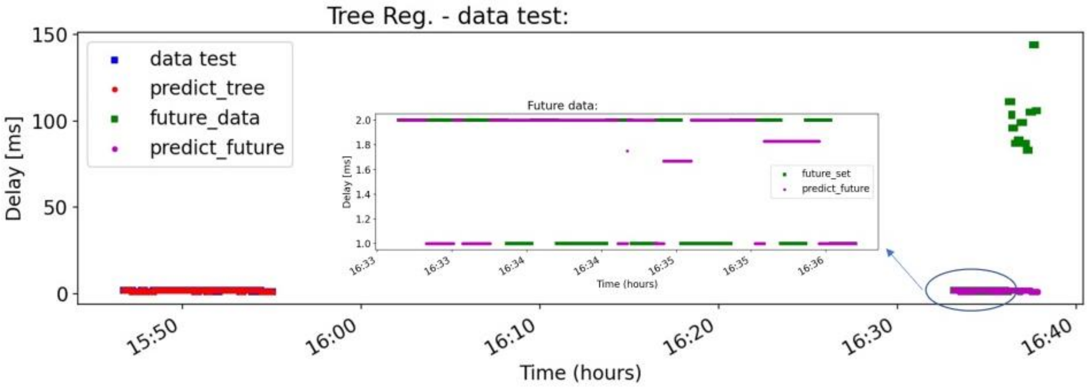


Figure 3-31: Tree Regressor model prediction for test and future data - 'amf-outbound-nrf-204'.

Fault classification was also performed using classification trees (DecisionTreeClassifier) with the same parameters as in the case of regression. In this case, the data was prepared for the data set 'amf-inbound-200-all-unauthenticated' where delays above 10ms were recorded as a failure. For 1200 delay values of which 859 are failures (≥10 ms), the confusion matrix shown in Figure 3-33 was obtained, as well as the parameters: F1_score_tree: 0.93, Precision score: 0.96, Recall_score (Fullness): 0.90. This model is very sensitive to overfitting (Figure 3-32) hence its performance should be monitored.

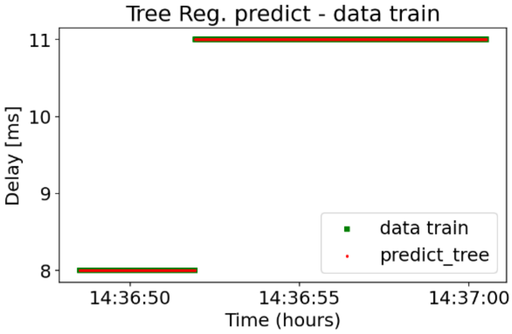


Figure 3-32: Data set 'amf-inbound-200-all-unauthenticated' undergoing classification.

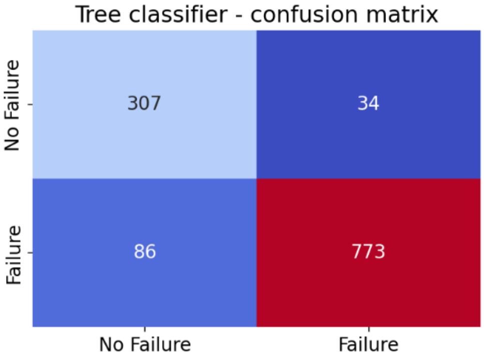


Figure 3-33: Confusion matrix.

Due to measured model fitting time and inference time for different ML algorithms, and the possibility of implementation of ML algorithms in AIaaS, regression trees were selected. Regression trees learn the fastest (fit) and process data the fastest (predict), without requiring significant hardware resources (amount of memory, number of processor cores), which allows their potential implementation in AIaaS for 5G/6G Core failure prediction. To minimize the times of data processing (T1 - collecting raw data, T2 creating data sets, T3 - ingesting data set by a Machine Learning model) and the ML algorithms execution with preprocessing (T4), the critical attributes for failure prediction should be selected. Then only for these attributes' failure prediction should be performed. Based on the analysis of experimental results we can determine the times T1, T2, T3 and T4 as a function of the number of processed data while maintaining R2_score>0.7 and MSE in the range (0, 1> for the selected ML model. It should allow for the effective operation of ML algorithms in AIaaS.

3.1.2.8 Summary

The MLOps enabler is summarized in Table 3-5.

Table 3-5 Summary of MLOps enabler.

Description	MLOps offers a set of tools for efficient ML model lifecycle management, with a particular focus on distributed AI/ML functions in 6G networks. This includes optimization techniques and protocols designed to minimize communication, computing, storage, and energy costs, while ensuring privacy preservation and scalability of distributed AI/ML applications.
Key take-aways	• Measurable KPIs: ML model accuracy, security, and training time, communication and computation energy consumption and time, computing resource utilization, service operator profit • Non-measurable KPIs: Privacy-preservation, robustness, flexibility • ML model layer offloading allows efficiently utilizing resources across the computing continuum while minimizing on-device energy consumption as well as minimizing overall (network and on-device) energy consumption with increasing number of devices and applications. • Privacy-preserving data collection and training protocols allow for trusted distributed AI/ML applications development. • Due to measured model fitting time and inference time for different ML algorithms, and the possibility of implementation of ML algorithms in AIaaS, regression trees for failure prediction in 5G/6G Core were selected.

Requirements	• Key data-related requirements for MLOps include high-quality data acquisition, robust processing, comprehensive versioning, and proper alignment with models, highlighting the interplay between MLOps and DataOps. • Optimization-related requirements for MLOps include low-complexity, real-time decision-making mechanisms to minimize costs and facilitate scalable and responsive AI/ML operations across distributed networks. • Establishing communication links between distributed computation nodes is also imperative for ML model parameter exchange, demanding careful design as the network scales. • Continuous monitoring and feedback loops are needed to guarantee freshness of datasets and trained models across the network. • The introduction of privacy-preserving architectural components is essential for communicating data, ML model parameters, and for the collaboration between distributed nodes, while carefully considering the added complexity to the network architecture. • To minimize the times of data processing and the ML algorithms execution with preprocessing, the critical attributes for failure prediction should be selected. Then only for these attributes' failure prediction should be performed.
Standard relations & regulations	• 3GPP TR 23.700-82 [23.700-82] • TR 38.817 [38.817] for RAN intelligence • 23.288 Vertical FL

3.1.3 AIaaS

3.1.3.1 Introduction

With 6G, the realization of a data-driven architecture with native and in-network AI capabilities is fundamental for enabling full automation in application, network and resource management and operation, while helping to satisfy the mobility, performance, pervasiveness and sustainability requirements imposed by 6G use cases and services. In this context, AIaaS becomes a key enabler for the implementation of AI-native 6G networks through advanced AI capabilities and services exposed via APIs. Specifically, AIaaS builds on top MLOps capabilities and functionalities to offer a broader range of managed AI/ML services, including pre-built models, datasets, algorithms, on-demand training and inference services. This facilitates applications in accessing and consuming AI functionalities in the form of services through common exposure APIs (that can leverage on 3GPP and CAMARA specifications and solutions), thus without the need of owning and maintaining the required AI and computing infrastructures.

The realization of such an AIaaS solution requires adequate AI/ML model lifecycle management capabilities to be supported, which on the one hand enhance and expand the basic MLOps functionalities, and on the other guarantee to satisfy a wide range of applications and use cases, such as Digital Twins and Transfer Learning that pose specific constraints in terms of AI/ML model interdependencies and relationships in network and service management and orchestration.

In the context of AIaaS, it is also critical to clearly define AI/ML services and capabilities to be offered to different consumers (e.g., application layer or management and orchestration layer), following a unified approach for exposure through APIs to query, create, access and consume such services. For this, an AIaaS software prototype has been also implemented to support and validate distributed AI/ML services composed by interconnected AI/ML functions deployed and operated on top of heterogeneous edge/cloud continuum environments

3.1.3.2 Exposing AIaaS Functionalities for AI-Native 6G Networks

As 6G networks emerge, a shift towards "AI-native" design is set to revolutionize telecommunications. Unlike previous generations, 6G will integrate AI and ML across its architecture, enhancing service delivery and functionality for Communication Service Providers (CSPs). In this evolving environment, AIaaS takes center stage, enabling 6G networks to support AI-driven applications across diverse use cases. This section explores AIaaS as a means for CSPs to deliver advanced AI capabilities via application programming interfaces (APIs), examining a specific AIaaS use case, potential architectural configurations for AIaaS, and implications for standardization.

Defining an AI-native architecture for 6G is challenging, as it varies based on the network structure. For example, within a functional architecture, AI-native traits might appear in different ways, such as integrating MLOps frameworks, which may become standard across CSP domains. MLOps frameworks simplify lifecycle management of AI/ML models, allowing for easy deployment, operation, and monitoring across various network domains. As 6G evolves, network capabilities, including network slicing, edge computing, QoS, location-based services, AI-based optimization, and security and privacy mechanisms, are expected to be accessible via APIs, providing developers with tools to innovate. A critical element in this evolution is MLOps-as-a-Service (MLOps-aaS), where CSPs provide MLOps frameworks through APIs to applications. This model offers CSPs a competitive edge, particularly in delivering services with specific QoS guarantees and real-time network insights. Building on this, AIaaS expands MLOps-aaS by offering a broader range of AI capabilities, including pre-built models, datasets, and algorithms, accessible through APIs, enabling applications to use AI functionalities without developing or maintaining their infrastructure. This positions 6G networks as a powerful platform for supporting innovative applications and services.

Towards an AI-Native 6G Architecture through AIaaS

An AI-native network design incorporates intrinsic, trustworthy AI capabilities across design, deployment, operation, and maintenance phases. In telecommunications, this architecture is defined by four key aspects: 1) intelligence everywhere, 2) a distributed data infrastructure, 3) zero-touch management, and 4) AIaaS. This paper focuses on AIaaS, which requires new AI and data-handling functions that may be exposed as services to external parties, such as life cycle management of AI models, execution environments, and data exposure.

Exposure of AIaaS APIs

AIaaS enables applications to access AI functionalities without requiring dedicated AI infrastructure, thus supporting a variety of innovative use cases. This approach aligns with 3GPP’s functional architecture by placing AIaaS within specific architectural elements as NEF, Service Exposure Application Layer (SEAL) and CAMARA. One approach is to leverage the NEF when exposing AIaaS functionalities to entities external to the CN. Through NEF, the AIaaS framework can operate similarly to an external Application Function (AF) by accessing real-time network data, including traffic patterns and user behavior, to train or execute AI/ML models. NEF’s service exposure capabilities—such as location, QoS, and analytics—serve as a standardized and secure interface, enabling AIaaS to leverage core network functionalities for use cases requiring advanced data processing and AI/ML-driven insights. Additionally, NEF enhances security by managing access to sensitive data, ensuring the AI models developed within the AIaaS framework are both trustworthy and secure. Another key function in this ecosystem is the NWDAF, a 3GPP-standardized component in the CN that aggregates and analyzes vast amounts of network data from various functions, empowering CSPs with data-driven insights. NWDAF can employ AI/ML models to process data and generate analytics, aligning with MLOps frameworks. In the future, NWDAF may integrate with MLOps frameworks to expand its capabilities. When exposed through NEF, NWDAF analytics services and MLOps frameworks could serve broader network functions. Alternatively, the MLOps frameworks might not be embedded within NWDAF, instead acting as a shared function for multiple network domains, as shown in Figure 3-34, with NEF providing exposure from this central function.

Another option for exposing AIaaS APIs is through the 3GPP SEAL framework, defined by the 3GPP Technical Specification Group (SA6) which offers standardized interfaces for making network services accessible to external applications. SEAL facilitates interoperability between network elements and external applications, allowing CSPs to securely expose services to third-party users while protecting against threats. The SEAL-based framework in the AI/ML Enablement (AIMLE) architecture enables vertical industries to consume AI/ML capabilities via APIs, offering functions like AI/ML client registration, discovery, and selection; Transfer Learning; Vertical and Horizontal Federated Learning (FL); and endpoint configuration and discovery for distributed AI/ML operations. This framework also supports AI/ML model and service life cycle management and interacts with underlying 3GPP systems, such as NEF and SEAL, to enable effective AI/ML service delivery. The architecture can also use application-specific data from VAL applications and integrate with the Application Data Analytics Enablement Server (ADAES) for advanced performance analytics, creating a robust AIaaS framework.

The Common API Framework (CAPIF) from 3GPP also plays a significant role by providing a standardized interface to integrate northbound APIs, particularly for API providers. CAPIF supports seamless integration between AIaaS solutions and network functions or external applications (Figure 3-34Figure), offering API security features like secure authentication, authorization, and usage monitoring. This enables scalable and efficient interactions between AI-driven services and network components while maintaining high security standards. CSPs can leverage CAPIF to manage APIs (such as NEF and SEAL) and ensure they comply with CAPIF, thereby enhancing API management and interoperability.

A third option is using CAMARA, an open-source project from the Linux Foundation [Cam24], which aims to harmonize common network service exposure and enable secure, flexible, and scalable AIaaS deployments. CAMARA uses northbound APIs, including NEF and SEAL, to facilitate AIaaS, with a focus on integrating network analytics across various services. Although there is no specific working group on AIaaS within CAMARA, it benefits from the collaboration of diverse industry stakeholders, providing access to a range of network capabilities that hold potential for AI application development.

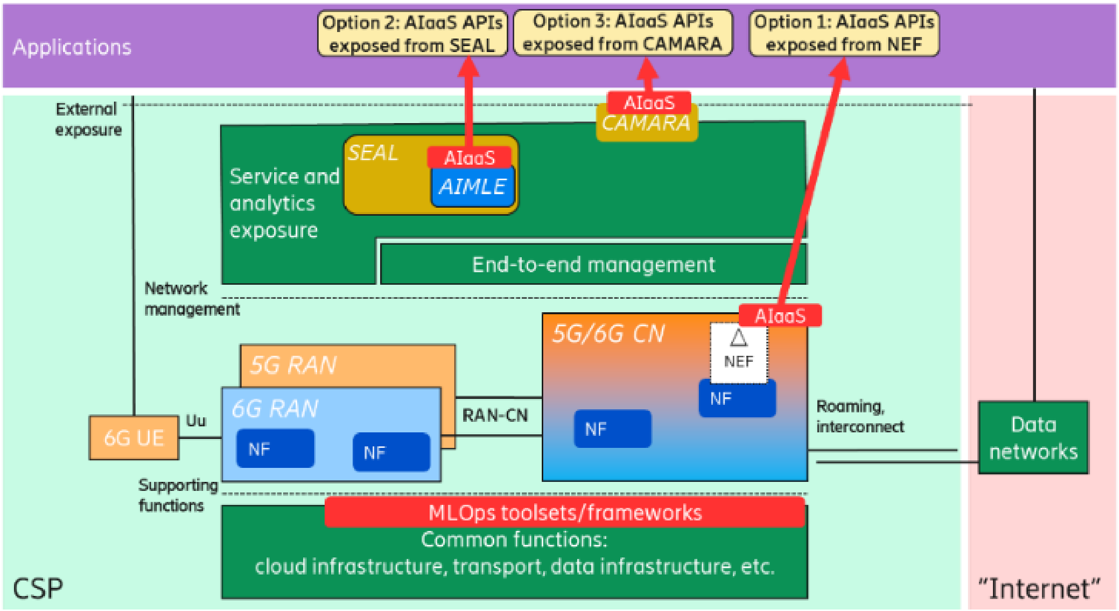

Applications
Option 2: AIaaS APIs exposed from SEAL
Option 3: AIaaS APIs exposed from CAMARA
Option 1: AIaaS APIs exposed from NEF
External exposure
AIaaS
CAMARA
SEAL
AIaaS
AIMLE
Service and analytics exposure
End-to-end management
Network management
AIaaS
5G/6G CN
NEF
NF
NF
5G RAN
6G RAN
NF
NF
RAN-CN
6G UE
Uu
Roaming, interconnect
Data networks
Supporting functions
MLOps toolsets/frameworks
Common functions:
cloud infrastructure, transport, data infrastructure, etc.
CSP
"Internet"

Figure 3-34: Potential locations for AIaaS within the standardized functional architecture.

3.1.3.3 AI-native architecture for AI/ML lifecycle management in AIaaS

Next generation networks and services increasingly rely on AI/ML methods to integrate intelligence in the decision-making process with the introduction of MLOps pipelines and Closed Control Loops for automating orchestration and lifecycle management. Further, Digital Twinning is viewed as a key enabling technology for 6G networks due to its foreseen benefits in network management optimization by providing a digital replica of the network. However, further work is required for an AI-native 6G architecture that fully integrates AI/ML into network and service management. Indeed, the use of Digital Twins and Transfer Learning leads to dependencies between models, which ought to be taken into account during their lifecycle management.

To this end, a framework and set of procedures have been developed for enabling an AI-native network architecture which supports the handling of AI/ML model interdependencies and relationships in network and service management and orchestration [HEX224-D33] and support proactive and automated AI/ML model and digital twin lifecycle management, as well as a unified AI/ML model exposure interface to facilitate model selection and management.

To support AI/ML model lifecycle management and AIaaS exposure, the developed framework was aligned with the AI/ML application enablement layer (AIMLApp) work in 3GPP SA WG6 [23.700-82, 23.482], and solutions were contributed for AI/ML model lifecycle management considering model dependencies.

The AIMLApp work in 3GPP defines the architecture, APIs and protocols for AIML application Enablement (AIMLE). The architecture includes the AIMLE server which manages the AI/ML operations, AIMLE clients deployed on e.g. UEs to perform training, and AI/ML enablement consumers. APIs and protocols have been defined for AI/ML model training, storage, retrieval, or AI/ML client registration and discovery. Additional procedures are also defined to support Federated Learning, Transfer Learning and Split Learning.

AI/ML model lifecycle management:

To support AI/ML model lifecycle management, the workflow illustrated in Figure 3-35 has been defined. The workflow is initiated when either the server or the consumer detects a performance degradation of the trained model. The enablement consumer then triggers the AI/ML model update in step 1 by sending the request to the server. The latter retrieves the ML model information from the ML Model Repository in step 3 and may also perform ML model discovery to determine whether an existing ML model stored by the ML Model and Data Repository can be used to train the new model (e.g., using Transfer Learning).

Additionally, to take into account the dependencies between the models and data, the AI/ML Enablement Server can also discover models that are related due to Transfer Learning or the use of the same training data, to identify additional models that may require an update. The update process can then be triggered for those related models as well.

Finally, in steps 4 and 5, the model is re-trained and either provided to the AI/ML enablement consumer directly, or by providing an endpoint information to retrieve the model from the repository. This procedure for AI/ML model update has been included in clause 8.21 in [23.482].

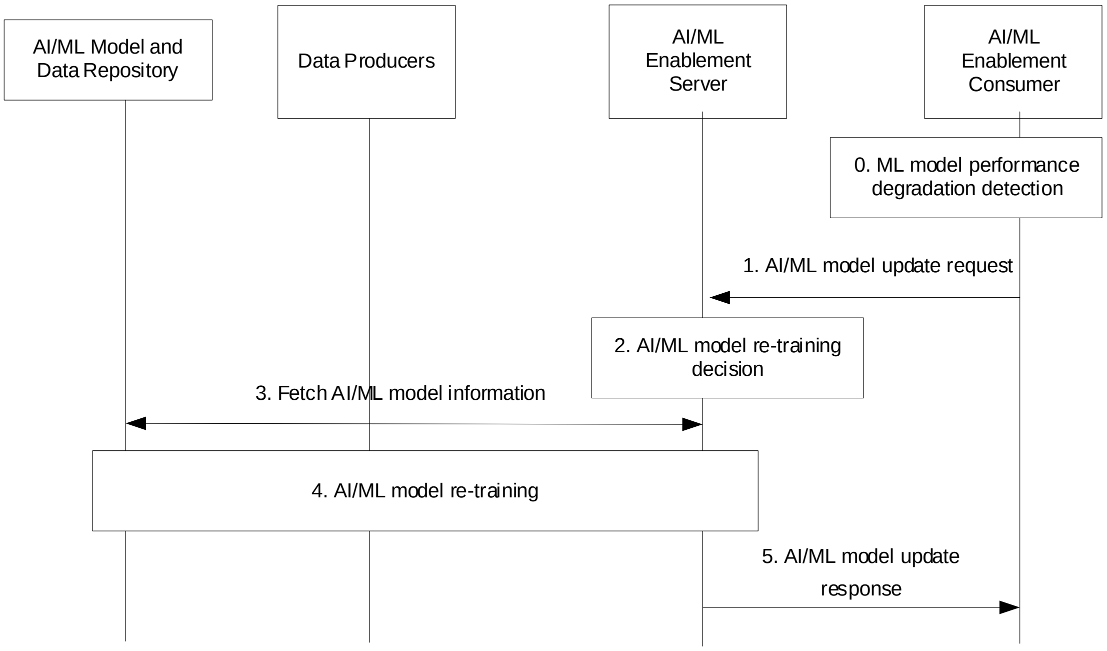

AI/ML Model and Data Repository
Data Producers
AI/ML Enablement Server
AI/ML Enablement Consumer
0. ML model performance degradation detection
1. AI/ML model update request
2. AI/ML model re-training decision
3. Fetch AI/ML model information
4. AI/ML model re-training
5. AI/ML model update response

Figure 3-35: Support for AI/ML model lifecycle management in 3GPP AIMLApp

AI/ML model storage and exposure:

To facilitate AI/ML model discovery and exposure to consumers, the AI/ML models are stored together with metadata in the AI/ML model and data repository. The ML model information specifies the domain of use of the model (e.g., for speech recognition, image recognition, video processing, location prediction, etc.), observed performance information, or specific interoperability information needed to deploy, use and share the model. This standardized model information format allows a simplified model exposure and discovery via the specified APIs and allows pre-trained model re-use instead of training new models from scratch, which facilitates and accelerates the model deployment process.

Additionally, to support the proposed lifecycle management with dependencies, the ML model metadata is augmented to include training information that indicate details on the data that has been used to train the model (e.g. data sources, volume, freshness), and the base model ID in case of Transfer Learning. This helps automate model management and enables proactive updates for related models that are expected to experience performance degradations in the future.

Currently, the features developed in 3GPP Release 19 for 5G Advanced only support AI/ML application enablement. In future releases for 6G, it is expected that support for digital twins will also be added. Then, the training information of AI/ML models that have been trained using digital twins can also include the ID of those DTs to track those dependencies as well.

3.1.3.4 AI/ML services offered through AIaaS and software prototype

The clear identification and definition of a set AI/ML services is required to make AIaaS a critical asset in the 6G network and service architecture. The management and API exposure aspects related to AI/ML services are already under definition in standards or de-facto standards, which are worth to take as reference baseline. Specifically, two main sources have been considered: O-RAN and 3GPP SA5. In the context of the specifications of the O-RAN Non-RT RIC framework, AI/ML workflow functions and services, as well as detailed operation workflows are defined [ORAN003]. Even if O-RAN focuses specifically on RAN aspects, it provides relevant technical reference for the identification of common AI/ML services, such as to cover capabilities for training, registration, storage, discovery, location retrieval and deployment of AI/ML models, as well as subscriptions to changes/updates to AI/ML models and performance monitoring. On the other hand, the 3GPP SA5 defines AI/ML management functionalities and service frameworks, introducing the concept of ML entity related AI/ML operational workflow [28.105][YAM23]. In particular, 3GPP SA5 identifies the ML entity as either an ML model or an entity containing an ML model and the ML model-related metadata, considering it in practice manageable as a single composite entity. On top of this, different phases of its lifecycle are identified (training, testing, emulation, loading, inference). Starting from these O-RAN and 3GPP references, Table 3-6 lists the AI/ML services offered through AIaaS.

Table 3-6 AI/ML services offered through AIaaS

AI/ML Service	Service Scope
AI/ML model management	Query of available AI/ML models (catalogue) Query of AI/ML model details (training information, metadata, inference constraints, etc.) Query of available AI/ML inference services (deployed AI/ML models)
AI/ML training	AI/ML training invocation (consumer and producer-initiated training) AI/ML training policy management • training/re-training process • automated/manual triggers AI/ML training performance management AI/ML training data management (load data, reference to data, etc.)
AI/ML validation	Evaluation of AI/ML training performance (on the validation data)
AI/ML testing	AI/ML testing invocation • Test AI/ML model inference on specific dataset

	• Selection of specific performance metrics AI/ML testing management • producer-initiated testing (after training and validation) vs consumer-initiated testing or automated testing
AI/ML emulation	AI/ML inference emulation • evaluate AI/ML model inference performance in an emulation environment AI/ML inference emulation management • control and monitor AI/ML inference emulation processes, e.g., to start, suspend or resume the inference emulation
AI/ML deployment	AI/ML model deployment control and monitoring • Automated loading/deployment of AI/M model in the target AI inference function • Notification to consumers of AI/ML inference ready to be consumed AI/ML model deployment update
AI/ML inference	AI/ML inference invocation • Direct mode: the service consumer directly accesses the AI function which executes the AI/ML model to perform the inference • Indirect mode: the service consumer invokes the inference through the AIaaS AI/ML inference control • Control inference, activate/deactivate inference function and AI/ML model (e.g., schedule-based/policy-based) • Configure context and metadata for performing inference AI/ML inference performance evaluation • Monitor and evaluate inference performance
AI/ML monitoring	AI/ML inference performance evaluation • Monitor and evaluate inference performance

API Exposure Mechanisms

The exposure of the AI/ML services offered by AIaaS is another key aspect to address to enable the consumption of such services. This means AI/ML services need to be accessible through well-defined APIs that consumers in the application layer or in the 6G management and orchestration framework can programmatically access to consume the required training, inference, validation, monitoring capabilities. Therefore, adopting a standard method and approach for API exposure towards external entities become fundamental to enable and facilitate interoperability and AI/ML service consumption. In this context, the 3GPP CAPIF provides a standard approach for API registration and discovery [29.222], which can be applied and used by any system exposing APIs. It provides native API discovery capabilities, and while it does not act as an API gateway or proxy (thus not introducing an additional layer for API invocation), it can be also used to dynamically register (and therefore discover on the consumer side) AI/ML training, deployment or inference service and APIs that may be made available by the AIaaS at different points in time (e.g. due to new AI/ML models available, new AI/ML inference services deployed, etc.).

AIaaS software prototype for distributed AI/ML services

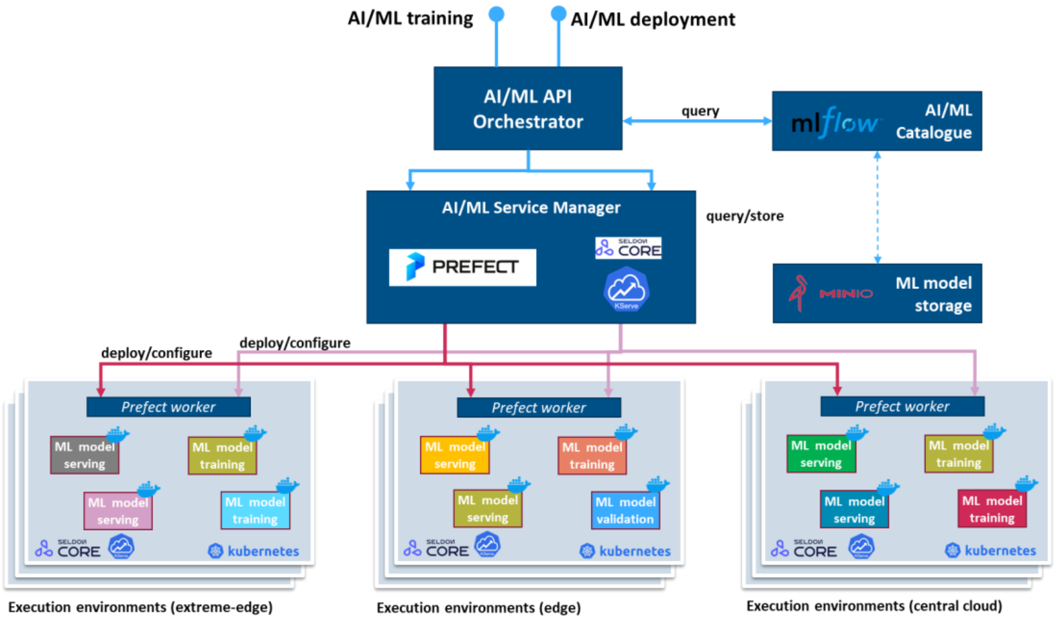


Figure 3-36 AIaaS platform proof-of-concept (mapped functional architecture reported in D3.3).

A software prototype for the AIaaS functional architecture presented in deliverable D3.3 has been implemented, with the aim of supporting distributed AI/ML services to be deployed and operated on top of heterogeneous edge/cloud continuum environments. As depicted in Figure 3-36, this AIaaS platform prototype integrates existing open-source tools to implement the functionalities of several functional blocks of the AIaaS functional architecture. The figure shows the mapping of the AIaaS functional blocks with the open-source tools used to implement their features and logics. The training management functionalities within the AI/ML service manager are implemented through the integration of Prefect [Pre24], a workflow orchestration tool that allows to automate the execution, scheduling and observability of distributed pipelines with a unified approach. This allows to execute training pipelines in distributed edge/cloud continuum environments (e.g. based on Kubernetes), following a declarative approach, through dedicated AI/ML training descriptors which contains the training steps and their requirements. The AI/ML model deployment functionalities in the AI/ML service manager are implemented through tailored Prefect pipelines, following a similar declarative approach through AI/ML deployment descriptors, that when executed allow to dynamically deploy and expose AI/ML models as REST services (through the integration with Kserve [KSER] and Seldon [SEL24], according to the availability and constraints in the edge/cloud continuum Kubernetes infrastructures). Dedicated Prefect workers, acting as hooks and drivers in the edge/cloud continuum environments, are deployed and integrated in Kubernetes services (or Virtual Machine, or bare metal) infrastructures to execute training steps or AI/ML deployments. The AI/ML Catalogue functionalities are supported through the use of MLFlow [MLF24], a tool that supports extensive training tracking and model accessibility through customizable metadata definition, which has been integrated with Prefect to publish the model resulting from the training services directly into the catalogue. It allows to maintain and compare multiple versions of the same model. Minio [Min24b], the high-performance Kubernetes native object storage is used as ML model storage, and it is also integrated with Prefect and MLFlow to store and expose the various model artifacts and have them ready for deployment.

With the aim of supporting highly distributed AI/ML services, like in the case of exposing federated learning capabilities and services, this implementation of the AIaaS platform has been enhanced with additional features. In particular, FEDaaS (Federation as a Service) and FLaaS (Federated Learning as a Service) solutions have been introduced and integrated in the AIaaS platform by integrating the open-source framework OpenFL [FSE+22][OFL24], which respect to other solutions (like Flower [FLO]) allows to natively create federations over a particular kind of data and use them to perform several federated training experiments. Along with a Federation Catalogue, OpenFL provides two abstractions that are well suited to implement FEDaaS and FLaaS in the AIaaS platform prototype:

- **Federation (FEDaaS):** distributed set of machines composed of a director and several envoys in which the director represents the central point of the federation and coordinates the envoys in the distributed training; these components are long-life units that can be used to launch federated training experiments. A common data format is defined and shared across the federation using the DataShard abstraction: a data class that defines the data format shared across the federation and provides information about the data location in each of the envoys.
- **Experiment (FLaaS):** represents a federated training experiment that is run over an existing federation. During an experiment the director creates an aggregator and each envoys create a collaborator; these are short-life components, with TTL equal to the experiment, that are actually participate in the distribute training. The role of each collaborator is performing local training using the data stored in the envoy while the role of the aggregator is to create global updates of the model aggregating the collaborators model updates.

More details about these FEDaaS and FLaaS implementations can be found in Annex A.3.

3.1.3.5 Summary of the AIaaS enabler

The AIaaS enabler is summarized in Table 3-7.

Table 3-7 Summary of AIaaS enabler.

Description	The AIaaS enabler implements an AI-native platform integrated within the 6G network architecture to provide AI capabilities as services in support of various applications. It provides a superset of MLOps features, incorporating MLOps APIs and enhancing them with additional capabilities (e.g., data exposure, on-demand training and inference, performance reporting). It offers tailored AI services through exposure of common APIs to meet specific consumer needs.
Key take-aways	• Measurable KPIs: AI/ML model performance (e.g. latency/response time, accuracy), compute resource utilization, operational efficiency • Non-measurable KPIs: AI model accessibility, automated execution, User satisfaction • AIaaS as a one-stop shop solution to ease the deployment, operation and execution of AI/ML services, pre-built AI models, datasets, facilitating their consumption through a set of open APIs • AIaaS allows to operate AI/ML services on top of distributed cloud-native infrastructures, integrating with the extreme-edge/edge/cloud continuum • AIaaS helps in simplifying the transition to AI-enabled services, which is key for those verticals and application providers with limited AI skills • AIaaS is a key enabler for AI and data monetization and new business propositions, and for the realization of new AI-driven vertical services and applications • Contribution to 3GPP and alignment with architecture and API specifications ensures wider adoption of the developed concepts
Requirements	• Complexity of internal “aaS” logics to satisfy heterogeneous use case and AI task requirements (e.g. for deployment constraints, performance needs, etc.) • Need to regulate ownership and interactions in case of multi-stakeholder scenarios (for AI/ML models, data for training and inference, access to services) • Need to define new APIs, protocols, data models for AI services (and their exposure) management with lack of standard specifications
Standard relations & regulations	• ETSI GS ZSM 012 • 3GPP TR 23.700-82 • 3GPP TS 23.482 • 3GPP TS 29.520

3.1.4 AI/ML-Specific 6G Architecture

The 6G architecture integrates AI/ML functionalities through distinct layers, enabling intelligent and adaptive network operations. The process begins at the **Application Layer**, where end-user or enterprise services request resources or capabilities, extending beyond traditional network resources to services like Compute-as-a-Service (CaaS) and AIaaS. Standardized APIs provided by the **Service Exposure Interface** simplify access to AI-driven capabilities, such as model training, inference, and data processing, ensuring seamless integration without deep core network modifications.

The **Application Enablement Platform Layer** mediates between applications and network functions, enriching service requests with AI-driven context, authentication, and routing. It translates high-level application needs into specific network commands, ensuring efficient orchestration and optimization. The **Network Functions Layer**, comprising RAN, transport, and core functions, processes these requests using AI to enhance connectivity, optimize resource allocation, and deliver adaptive services. AI-driven air interfaces further improve the physical and MAC layers in radio access networks.

At the core of the architecture lies the **AIaaS Framework**, encompassing tools like the AI Orchestrator, AI/ML Catalogue, and MLOps Orchestrator, see Figure 3-37. The **AI Orchestrator** manages AI service deployment, execution, and lifecycle, optimizing network traffic, predictive maintenance, and resource allocation. The **AI/ML Catalogue** serves as a repository of pre-trained models, algorithms, and metadata, supporting model versioning, interoperability, and seamless integration into network environments. The **MLOps Orchestrator** oversees model lifecycle management, handling deployment, retraining, and adaptation to ensure sustained relevance and accuracy.

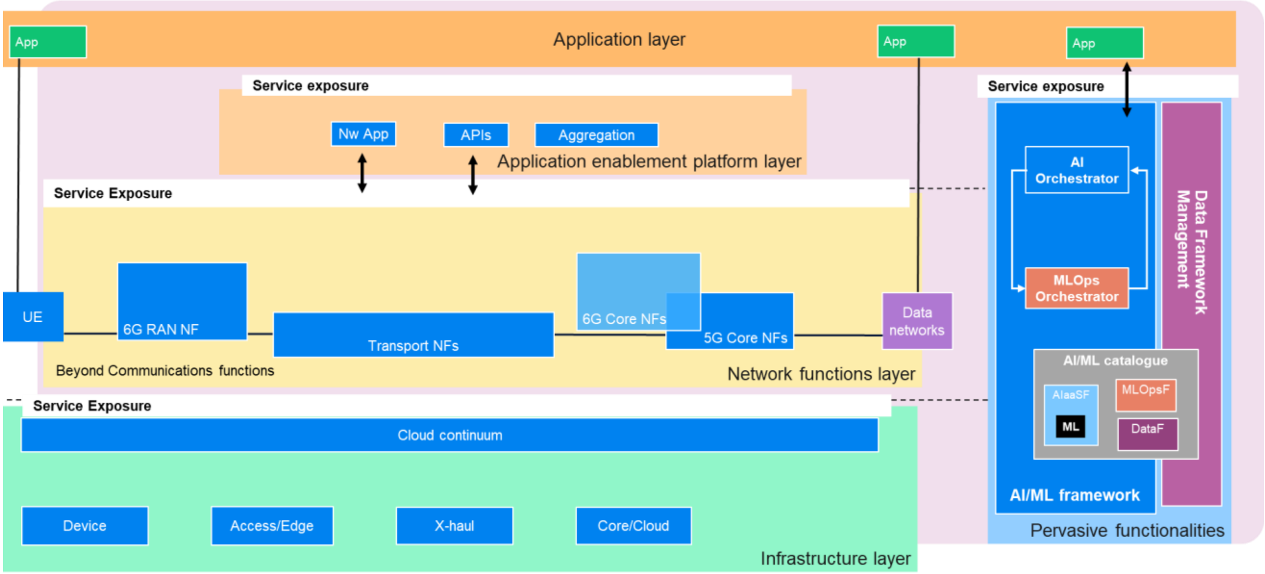


Figure 3-37: AI/ML specific view of 6G Blueprint Architecture

The architecture emphasizes cross-layer optimization by leveraging data from multiple sources—RAN, transport, and core networks—to predict congestion, enhance resource allocation, and enable predictive maintenance. The **Data Framework Management** supports these processes by ensuring secure data collection, processing, and governance, adhering to regulatory standards. Techniques like encryption, anonymization, and differential privacy protect sensitive information, while centralized and distributed data lakes provide high-quality data for AI/ML operations. This approach establishes a robust foundation for enabling advanced AI capabilities within 6G networks.

DataF: Data Framework Management in AI/ML Operations

The architecture in Figure 3-38 is a DataOps-centric framework that integrates the Data Function (DataF) as the central enabler for managing data flows and enabling AI-driven functionalities across multiple network layers. This architecture is designed to support dynamic, data-driven services and efficient integration of AI/ML workflows in next-generation networks like 6G.

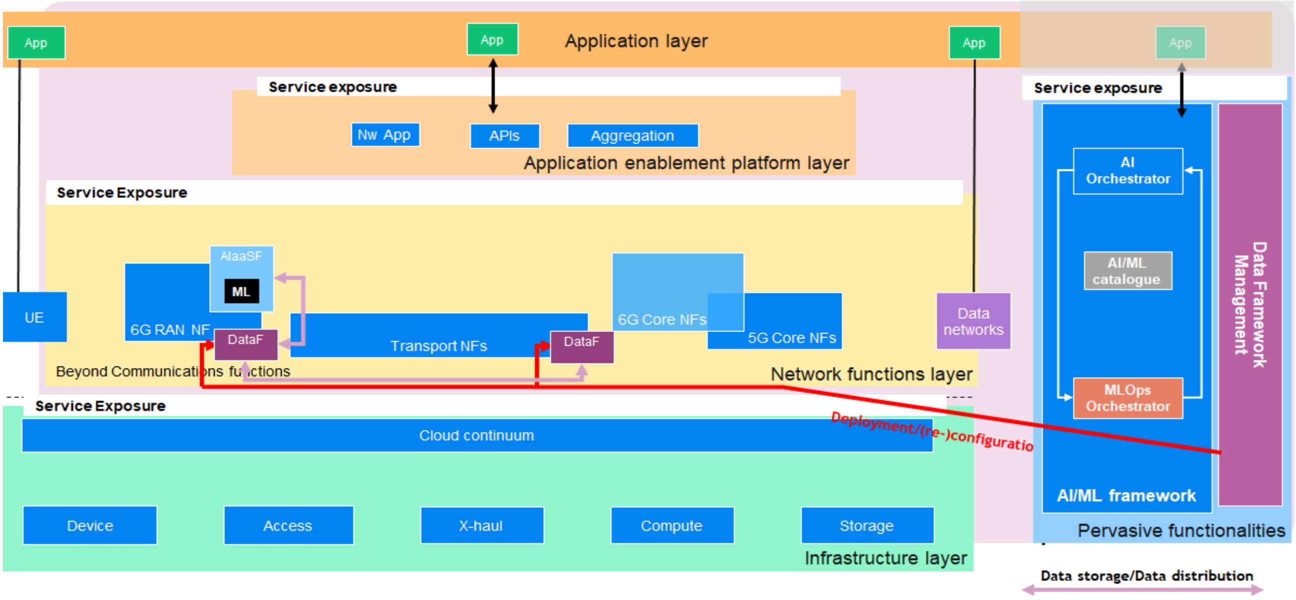


Figure 3-38: DataF in AI/ML Blueprint Architecture

The Infrastructure Layer forms the foundation, comprising physical and virtual resources for data generation, storage, and processing. This includes UE for data generation, access points for connectivity, X-haul for data transport, and distributed compute and storage resources that span the cloud continuum. These components ensure seamless data flow from the edge to the core of the network. The Cloud Continuum extends compute and storage capabilities, enabling efficient data handling. At the edge, low-latency data processing supports real-time applications, while centralized processing facilitates large-scale data analytics and AI/ML model training. This ensures optimal resource utilization and balanced data flow between local and centralized systems. At the heart of the architecture, the Network Functions Layer integrates DataF with critical network components, such as 6G RAN, transport, and core network functions. DataF aggregates, distributes, and dynamically manages data streams, enabling efficient routing, optimization, and provisioning of AI/ML services. Additionally, it supports beyond-communication functions like AIaaS, continuously supplying data to AI models for advanced use cases. The Application Enablement Platform Layer exposes network data and capabilities to developers through APIs powered by DataF. This layer allows network applications to leverage data for analytics, optimization, and innovative services. Data aggregation at this level prepares data for use by AI/ML workflows and end-user applications. The Application Layer represents end-user applications that rely on DataOps-enabled APIs to access real-time network insights and actionable data streams. These applications dynamically interact with DataF, enabling adaptive and innovative user experiences driven by data. The AIaaS Framework integrates DataOps principles for lifecycle automation of AI/ML models. Components like the AI Orchestrator and MLOps Orchestrator utilize DataF for continuous data provisioning, enabling tasks such as model deployment, scaling, re-training, and inference. DataF ensures secure, efficient, and compliant data handling across the framework. Finally, DataF acts as the core of the architecture, enabling workflows such as data-driven deployment, continuous data flow, and service exposure via DataOps. By managing data storage, distribution, optimization, and privacy compliance, DataF ensures seamless integration of AI/ML into network operations. This architecture positions DataF as essential for managing data lifecycles and delivering dynamic, AI-powered services in networks like 6G.

MLOpsF: Model Lifecycle Management Orchestrator

In the 6G data driven architecture, AI/ML plays a key role in enabling intelligent and autonomous network operations by managing the entire ML development lifecycle that consists of model initialization, training, deployment, monitoring, and adaptation for instance in the case of concept drift, see Figure 3-39. MLOps functionalities are required in the data driven architecture to host ML models wherever needed, and these ML models are expected to assist functionalities. ML models can be exposed and offered as a service to improve applications via the AIaaS Function (AIaaSF), cf. Figure 3-40. MLOps orchestrator is a pervasive functionality and is placed within the AI/ML framework. The MLOps orchestration takes into account KPIs such as communication, computation, storage, and energy costs for training and maintaining the lifecycle of the ML models and optimizing the operations such a way to satisfy the demanded KPI constraints as in most cases there exists trade-off in between them as presented in this deliverable.

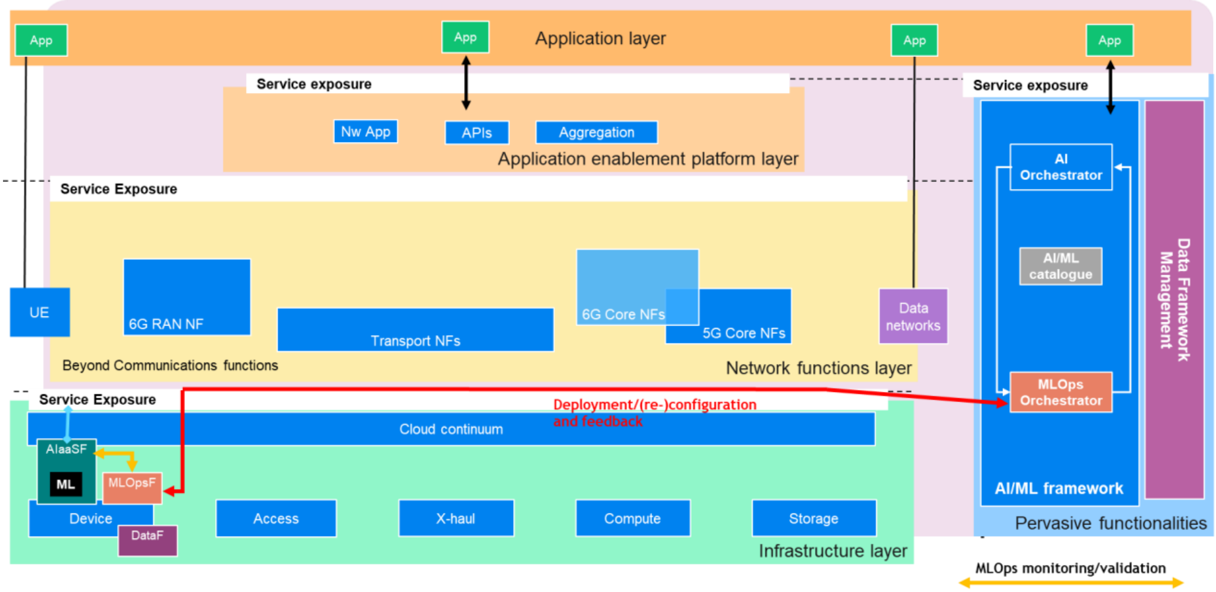


Figure 3-39: MLOpsF in the AI/ML Blueprint architecture

Standardization of AI/ML in RAN L2 and L3 via 3GPP is introduced in Rel 17 with SON functions including energy saving, optimization of mobility, and load balancing. Later in Rel 18, the focus was on the the training and deployment of AI/ML. Distributed learning techniques such as Federated Learning between NWDAFs were also considered in Rel 18. In standardization in 3GPP SA2 [23.288], vertical federated learning between ML models in the network and application functions is discussed and included in Rel 19. Splitting model between different entities based on the delay and privacy sensitiveness between the network and device is included in [22.261].

There has been significant efforts in AI/ML in the O-RAN, where AI/ML can be deployed in non-RT RIC and near-RT RIC depending on the use case requirements. For fast control loops, e.g., for the mobility type of use cases where handover decisions need to be given fast and frequently, deployments in the near-RT RIC is preferred. Non-RT RIC is designed to host rather slower control loop use cases such as network optimization. In that case, AI/ML are deployed within the rApps or at the Non-RT RIC framework. AI/ML workflow functions support model training, monitoring.

In the Hexa-X-II architecture, standardized AI/ML functions such as NWDAF (as well as sub-functions such as Model Training logical function (MTLF) and Analytical Logical Function (AnLF)) can be hosted within the Network functions layer.

AI-as-a-Service Framework

As illustrated in Figure 3-40, the architecture includes the AIaaSF, which is in charge of providing AI/ML training, inference, and testing services. The functions are managed (i.e. deployed and operated) by the AI orchestrator in the AI/ML framework, and the trained AI/ML models are stored in the AI/ML catalogue. Multiple AIaaSF instances can also operate in a collaborative manner (e.g. for distributed ML), with the coordination of the AI orchestrator.

The AIaaSF embedding the ML models can be integrated into NFs or deployed separately in the infrastructure cloud continuum. The AI capabilities can be exposed directly by the AIaaSF towards the application layer (e.g. through standardized APIs such as CAPIF [29.222]), or indirectly, via the AI orchestrator which is similar to the AI/ML application enablement layer (AIMLApp) in 3GPP [23.700-82, 23.482], where the AI/ML enablement is provided to the application layer via the AI/ML enablement server, which is responsible for managing and exposing AI/ML models.

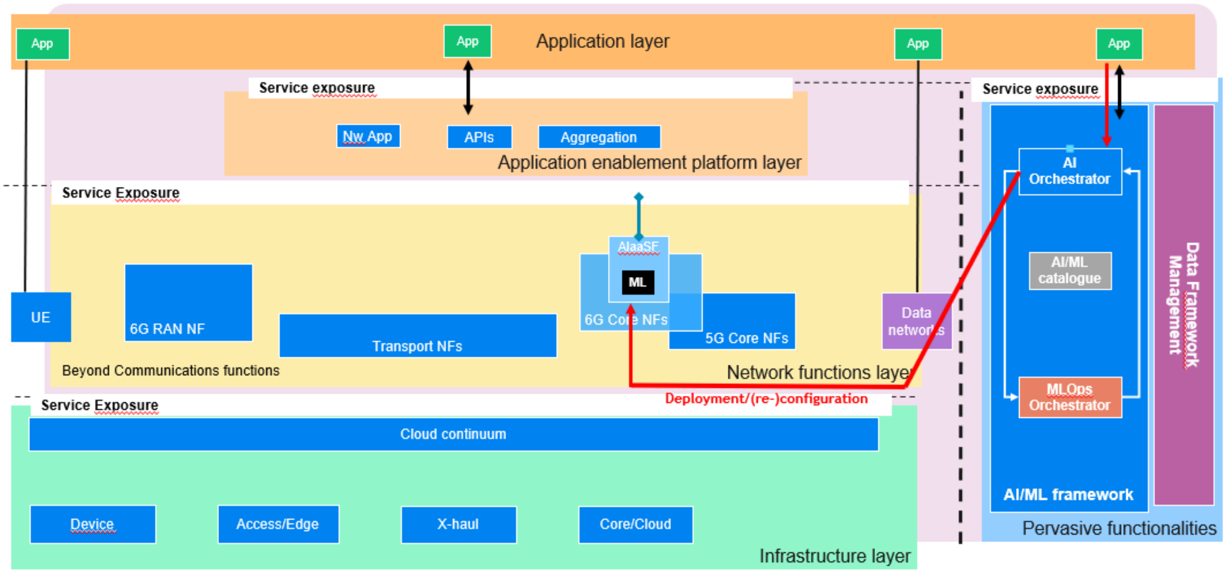


Figure 3-40: AIaaSF in the AI/ML Blueprint architecture

3.2 Integrated Sensing and Communication

This enabler explores the fundamental concepts and technologies that drive the convergence of sensing and communication, highlighting the optimization of resource utilization, the implementation of robust protocols, and the adoption of innovative approaches for autonomous environment mapping and navigation.

3.2.1 Sensing Management Function (SeMF)

The provision of sensing services by next generation communication systems necessitates the introduction of a Sensing Management Function (SeMF) that will be responsible to facilitate the efficient coordination of sensing procedures, considering various aspects such as sensing requirements, sensing capabilities, sensing constraints, etc., as introduced in [HEX224-D33]. The SeMF can also include sensing control, sensing processing and the sensing requests.

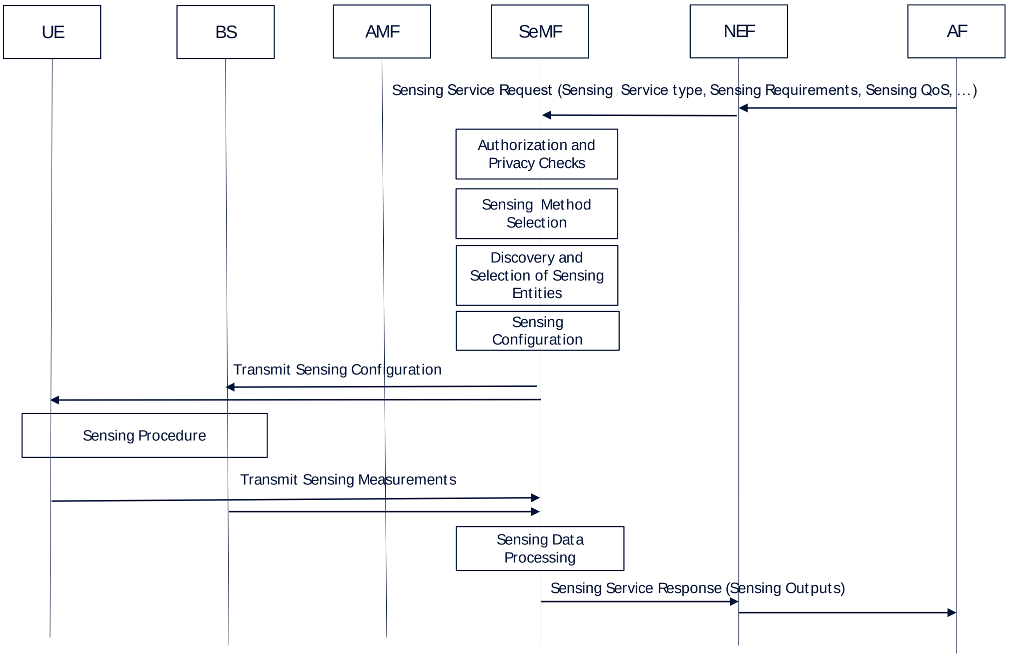

UE
BS
AMF
SeMF
NEF
AF
Sensing Service Request (Sensing Service type, Sensing Requirements, Sensing QoS, ...)
Authorization and Privacy Checks
Sensing Method Selection
Discovery and Selection of Sensing Entities
Sensing Configuration
Transmit Sensing Configuration
Sensing Procedure
Transmit Sensing Measurements
Sensing Data Processing
Sensing Service Response (Sensing Outputs)

Figure 3-40 Example Flow Chart for Sensing Service Request from an Application Function

A sensing client can be a UE or an AF or even an in-network function (NF) triggering a sensing request with respect to one or more sensing target objects within a sensing target area, where the sensing target object may be a passive or non-connected object, which distinguishes sensing from the classical positioning service. Figure 3-40 describes an example of how a sensing service request by an AF could be supported by a communication system. The AF can send a sensing service request via the NEF or directly to the SeMF (in case of a trusted AF). The AF can send a one-time request or a subscription of the sensing service for event based periodic notification. In the case that the sensing client is a UE, then NAS protocol could also be used for the sensing service requests, as shown in the Figure 3-41.

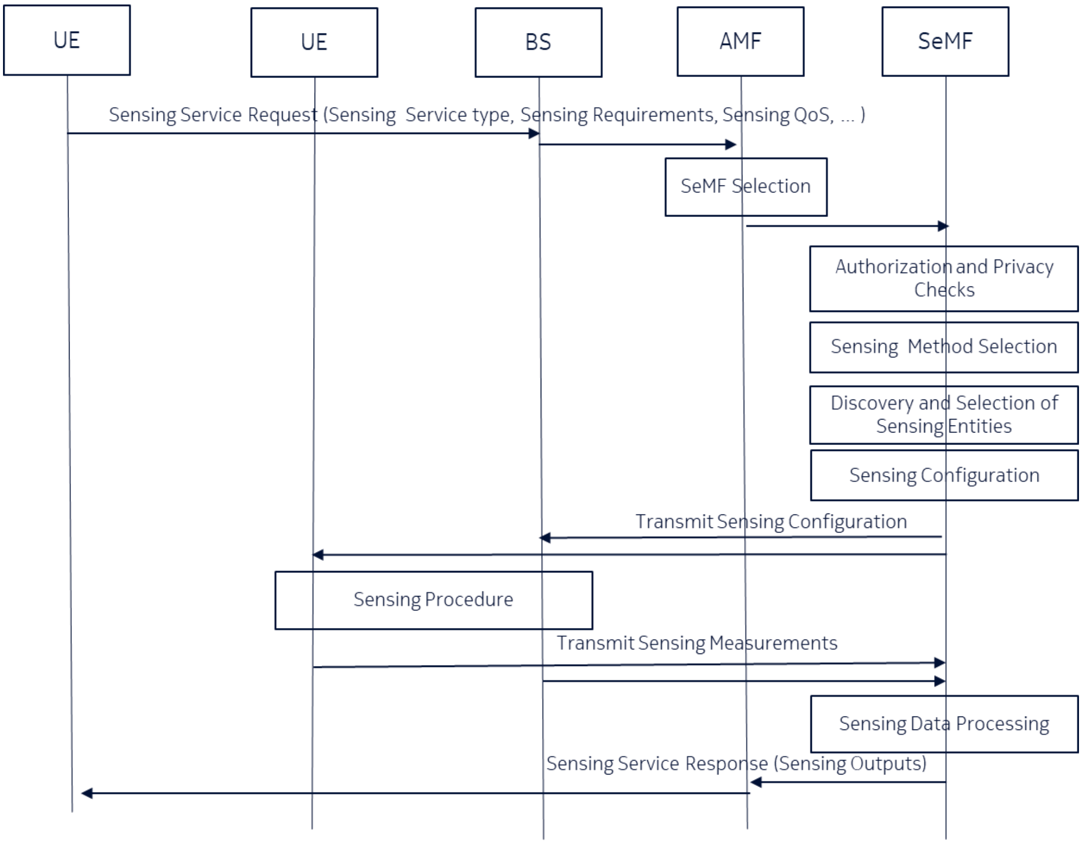

UE
UE
BS
AMF
SeMF
Sensing Service Request (Sensing Service type, Sensing Requirements, Sensing QoS, ...)
SeMF Selection
Authorization and Privacy Checks
Sensing Method Selection
Discovery and Selection of Sensing Entities
Sensing Configuration
Transmit Sensing Configuration
Sensing Procedure
Transmit Sensing Measurements
Sensing Data Processing
Sensing Service Response (Sensing Outputs)

Figure 3-41 Example Flow Chart for Sensing Service Request from a UE

The information elements that are provided in the **sensing service request** depend on the target scenario and application, and can include the following information:

- Sensing service e.g., tracking of an object, detection of objects, environment monitoring etc.
- Target sensing area described, for instance, using GPS coordinates or geographically scoped or relative to a known object or known location such as a requesting UE indicates the target sensing areas as a circle around itself.
- Time and/or duration of the requested sensing
- Type of sensing output e.g., detected presence or absence indication, object map or unprocessed data
- Sensing configuration parameters including
    - type of the Object(s) e.g., car, drone, human etc.
    - mobility profile of the Object(s) e.g., static, low mobility, speed information indication
    - object size
    - environment of sensing e.g., indoor
    - altitude
- Sensing QoS requirements
    - Sensing resolution in terms of range and/or velocity
    - Sensing Accuracy in terms of detection probability, position and/or velocity
    - Maximum sensing service latency
    - Update rate of the sensing outputs

The sensing client can receive the **sensing service response** e.g., via a NEF in case of an AF or via a NAS message in case of a UE request, and the response can include the following information:

- Sensing Method Used and sensing configuration information
- List of Sensing time and/or location instances
    - (List of) Objects
        - Object ID
        - Object Location
        - Time of measurement information
        - Object Features
        - Mobility information of objects (e.g., static or low mobility profile or speed threshold)
- Environment information (e.g., weather conditions)
- Achieved Sensing QoS and respective confidence information on sensing QoS parameters.

The protocol for the interaction between the SeMF and the BSs and/or the UEs is another important interface of the SeMF. The SeMF selects the appropriate sensing method and configuration information, according to the requirements received from Sensing client as part of the Sensing Service Request, and the available sensing capabilities in the area that the request targets, which will help to determine sensing configuration actions. For instance, in case of BS-based sensing, the SeMF can determine the required resource allocation characteristics and signal them to all involved BSs, which are taken into consideration by their schedulers during resource allocation. Together with the sensing resources configuration, the SeMF includes also in the transmitted configuration message the sensing-related information and measurements that each BS should report (e.g., Objects, periodograms, point clouds). Of course, in case of bi-static or multi-static sensing the required measurements are provided only to the BS(s) that have the role to receive and process sensing signals. In addition, Sensing Requirements (e.g., the target sensing area, duration, update rate) together with Sensing QoS requirements (e.g., sensing accuracy, sensing resolution, sensing delay) can be also included in the transmitted configuration from the SeMF to the BS.

The SeMF can enable the utilization of the communication network to support sensing services, utilizing communication sites and the large coverage that they can provide, trying in parallel to avoid impacting the communication services and involve with the assistance of UEs, whenever it is needed. The exposure of sensing services can provide new monetization opportunities for the Mobile Network operations. The potentially high volume of sensing data collection and the privacy-aware collection and processing of sensing data are aspects that the SeMF can address and manage.

3.2.2 Sensing procedures

In [HEX224-D33], the relation of some identified sensing functionality has been described in a functional architecture. This section provides more details on various sensing functions and their procedures, in particular the sensing control function, see Figure 3-42. The application is sending a sensing request, which includes what and where to sense, authorization, privacy, etc. The application can be located, e.g. in a UE, to detect an object, or in the RAN, to make sure that the transmission path is free from obstacles. Once the request passed checks for authorization, etc., it is forwarded to the sensing control, which configures measurements and processing based on the request and available resources in the area (given in the request). Based on this configuration the sensing measurements are executed, involving a transmitter transmitting some signal in the requested area and a receiver receiving the resulting waveform. The transmitter and receiver can be collocated but they can also be in separate locations. The resulting waveform or measurement based on it is forwarded to the sensing processing function, which extracts useful features from the measurement data and provides sensing results, useful for the application that requested the sensing.

If UEs are included in the sensing process some additional steps or actions are required. One such action is to identify and locate UEs capable of sensing and willing to take part in the current sensing process. A UE involved in the sensing process needs to be connected to a serving base station, which may be involved in the same sensing process.

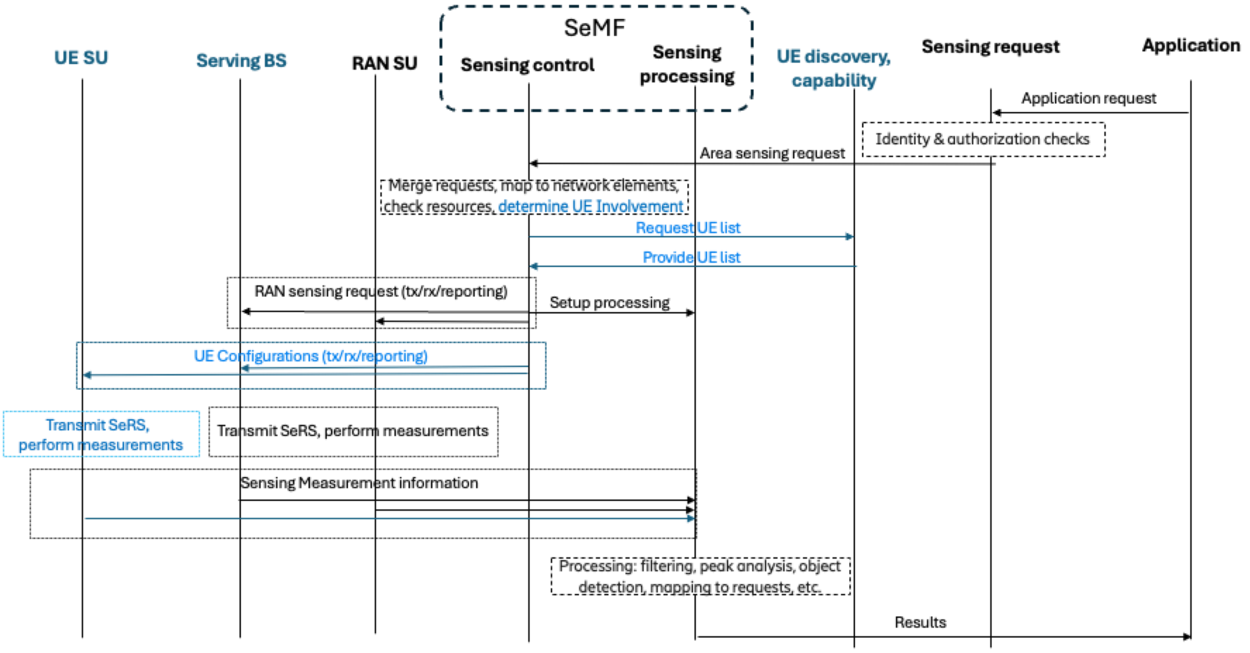

SeMF
UE SU
Serving BS
RAN SU
Sensing control
Sensing processing
UE discovery, capability
Sensing request
Application
Application request
Identity & authorization checks
Area sensing request
Merge requests, map to network elements, check resources, determine UE Involvement
Request UE list
Provide UE list
RAN sensing request (tx/rx/reporting)
Setup processing
UE Configurations (tx/rx/reporting)
Transmit SeRS, perform measurements
Transmit SeRS, perform measurements
Sensing Measurement information
Processing: filtering, peak analysis, object detection, mapping to requests, etc.
Results

Figure 3-42 Setup and execution of sensing in the functional architecture. The actions in blue are needed when UEs are involved in sensing.

The main control plane functions are: the application that requests sensing, a function that handles requests, and a sensing control function or management function the distributes the request to the RAN. The sensing processing function that interprets the resulting measurements is not really control plane or user plane and therefore there is a need to introduce a data plane (see [HEX224-D33]). In this deliverable, the focus is on the sensing control and possible overhead (as in radio resources) needed for sensing. As mentioned in section 3.2.1, the sensing control can be part of the SeMF. In Figure 3-42 the important parts of the sensing control function are depicted. The sensing unit (SU) selection function processes the information provided in the request and a priori information to understand which resources are needed. In more detail, the sensing request will include information on the locations to sense and what the application expects to learn about the location. The SU selection function knows, based on topographic maps of the area together with knowledge about the deployed network, which SUs are best suited for the task. For example, one SU is close to the requested area, but the signal would need to intersect a building, thus another SU needs to be selected. Once the SU selection function has determined which SUs will provide the best response to the request, the scheduler is contacted. The scheduler "owns" all the radio resources and schedules resources to users based on a number of different parameters [38.321] and distributes resources to all activities in a cell. Based on the request and the processing performed by the SU selection function, the scheduler provides resources needed to fulfil the request.

With the resources provided by the scheduler, the sensing units are configured to perform the sensing measurements. Measurement data are normally forwarded to the sensing processing function that extracts information to answer questions in the request. One could imagine a scenario where raw measurement data are sent to an external processing function, however, that option is not discussed further in this text.

In the scenario where one or more UEs are the appointed SUs that receive measurements, data also needs to be forwarded to the SPF and this forwarding needs to be done over the air, on the UL. Thus, in addition to resources obtained from the scheduler for actual measurements, additional radio resources are also needed for the forwarding of measurement data. The amount of radio resources needed to forward sensing measurements in the UE UL can be significant, perhaps even requiring some priority function, so that this process does not affect communication. Regarding priority, if the sensing function has high priority, e.g. the request comes from a first responder, this priority is used by the scheduler, so the sensing process receives resources first, while remaining resources will be scheduled among other devices with lower priority. Further, one could design some method in which the UE (working as a SU) has the capability to process the measurement data, at least by, e.g. removing unnecessary data, compressing data, identifying important features (and possibly being able to discard parts of the measurements), etc. The possibilities for the UE to process data is connected to UE capability, the situation in the network, etc.

In Figure 3-43, it is further assumed that the sensing unit selection function (SUSF) is collocated with the scheduler so that some fairly quick and simple interface is used for the signaling; i.e., when the sensing control request radio resources for a sensing process. One area of research that remains is to determine if colocation is better than some other deployment, e.g. is it more efficient if one SUSF handles several gNBs (and also schedulers), for example in a case where an object is tracked over the coverage of many gNBs. For this to work an interface is needed for signaling between SUSF and scheduler.

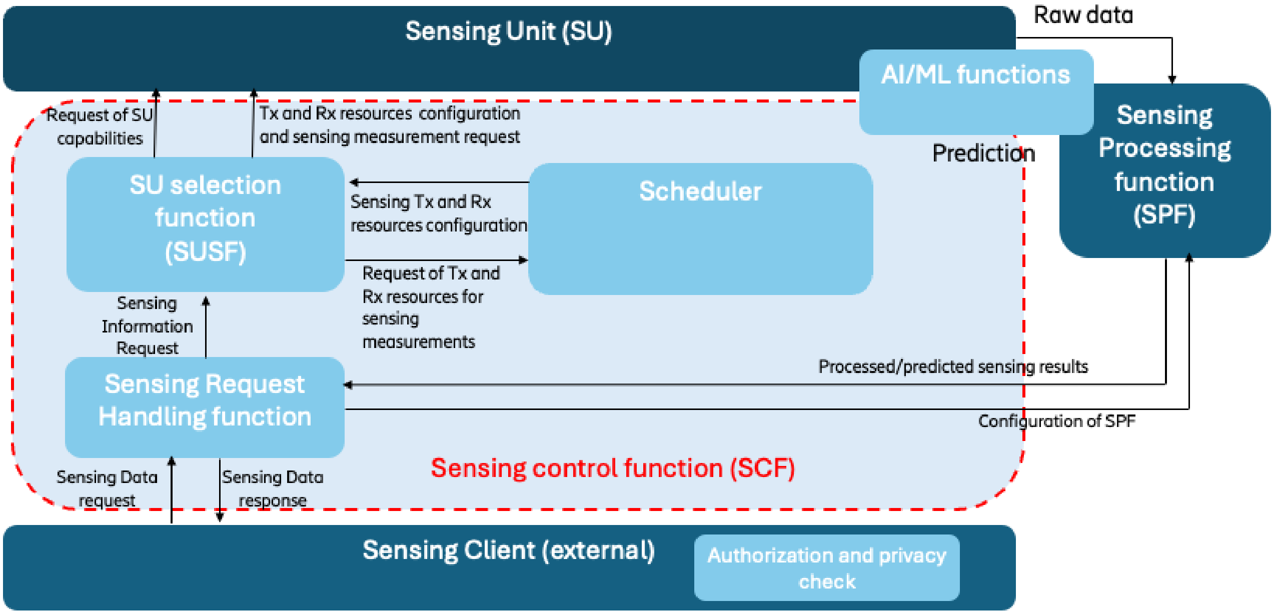

Sensing Unit (SU)
Raw data
AI/ML functions
Sensing Processing function (SPF)
Prediction
Request of SU capabilities
Tx and Rx resources configuration and sensing measurement request
SU selection function (SUSF)
Scheduler
Sensing Tx and Rx resources configuration
Request of Tx and Rx resources for sensing measurements
Sensing Information Request
Processed/predicted sensing results
Sensing Request Handling function
Configuration of SPF
Sensing control function (SCF)
Sensing Data request
Sensing Data response
Sensing Client (external)
Authorization and privacy check

Figure 3-43 The sensing control functionality

Overhead estimation for transmission

Overhead is here defined as the amount of resources, otherwise available for communication, used for sensing. Sensing needs resources during the actual measurements but may also need resources to transfer measurements to a sensing processing function. Depending on how a sensing signal is designed, the amount of overhead varies. There are at least 3 different methods to generate sensing signals:

1. Using communication transmissions also for sensing
2. Using some resource similar to PRS/SRS for sensing
3. Using some dedicated (scheduled) resource for sensing.

The first method relies on using communication signals also for sensing. By using communication signals there may be almost no overhead (i.e. even communication signals may need some configuration if used for sensing) at all during the measurement phase. An important drawback with this setup is that in case there is no ongoing communication, sensing measurements are not possible. In other words, sensing measurements can only be performed during communication sessions. There may be other drawbacks, e.g. the signals not being optimized for sensing, however, that does not affect overhead while affects the sensing quality.

In existing cellular networks there are specialized signals introduced to provide accuracy of positioning and location services, called positioning reference signals (PRS). There are several different configuration possibilities for PRSs, e.g. bandwidth, periodicity, frequency band, comb size etc. PRSs are not used for communication and could therefore be called overhead.

A third possibility could be to schedule dedicated resources to serve a sensing request. Assuming that there is a procedure similar to scheduling (for communication), a sensing request (via the SU control function) would trigger a signal for sensing measurements. In an example of bistatic setup (gNB and UE), if the signal is sent from the base station, on DL, and received by a UE; it is likely that the received signal measurement is reported on the radio interface (from the UE via a base station) to an SPF. If the transmitting sensing unit is a UE, the gNB would grant the UE resources to transmit the measurement signal, which would then be received by a base station. Resources granted for sensing measurements and for reporting of sensing measurements to an SPF need to coexist with resources granted for communication.

To understand overhead from sensing transmissions, we can look at an example. Similar to PRS configuration, the overhead due to sensing transmissions will be a function of bandwidth of the sensing signal, the number of pulses (repeated for this particular sensing measurement) and the refresh rate. These parameters also affect the accuracy of the measurement. Figure 3-44 shows the range and angle error “rule of thumb” equations for radar communication, as a function of SNR, bandwidth and number of beams (for the angle error) [Cur04] [YER24].

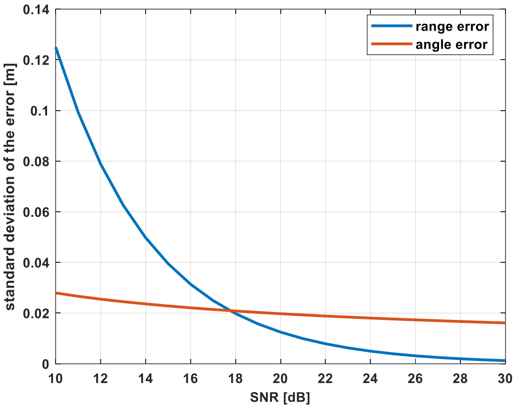


Figure 3-44 Relation between SNR and range error [Cur04].

For example, a larger bandwidth generally improves the detection rate of a sensing target. A drawback with large bandwidth is that it susceptible to background noise that may obscure the signal. The number of pulses directly affects the signal to noise ratio of the sensing signal, thus affecting both detection rate and (positioning) error.

Using system parameters such as bandwidth (300MHz), number of pulses (41), refresh rate (0.1s) and frequency (26GHz) the resource utilisation can be estimated [YER24] for a single sensing target the resource utilization can be expressed as being 0.4%. This is for a case where the number of subcarriers in the dedicated channel equals the number of subcarriers used for sensing, i.e. the entire bandwidth is utilized. In a multi-user scenario where the channel is shared between many UEs transmitting and receiving data and control signals, the overhead from sensing could be relatively larger. Also, this simple calculation only assumes measurements for one sensing target.

3.2.3 Sensing exposure

Section 3.1.1 discusses the need for exposure and APIs for AI and DataOps, but there is also a need for standardized protocols for sensing data collection and exposure, as well as roll out networks and devices with sensing capability. The interaction of networks with applications using application programming interfaces (APIs) has traditionally been termed exposure. The role of an exposure layer is to enable the services of the network to be consumed by other parties. The exposure functionality in 6G should be flexible and easy to use to be able to quickly open network services and adapting them to applications needs. The type of exposure is influenced by the desired outcome, i.e. depends on the use case. For example, one application may require an answer to a question, e.g., whether there is any object present in an area. Another case involves tracking of an object over a large area or observing movement within confined spaces resulting in a stream of measurements. Put into perspective, i.e. with the proposed architecture, the Data Plane (DP) can be suitable for continuous streaming output or when the result size is large, while the Control Plane (CP) may rely upon the existing network exposure function (NEF) framework and is suitable when sensing result size is not large.

The application may be connected to the sensing system via an API running on top of an existing exposure framework. APIs can exist at multiple levels for exposure directly from the 3GPP NEF interfaces [29.522], CAMARA [Cam24] or service enabler layer-like intermediate APIs (e.g. Service Enabler Architecture Layer for Verticals (SEAL)) or global aggregator exposure from API providers, see Figure 3-45. Aggregator APIs are often provided as easily usable libraries with services that application developers can apply. It is even possible that sensing could happen via ORAN rAPPS. It is likely that several exposure interfaces are needed, the pure minimum being an exposure interface from an individual operator's NW to applications and global aggregators and an interface from aggregator to applications.

A typical application of an API for sensing involves the following steps:

- Sensing service request that initiates the service, including authorization and authentication of the sensing request
- An admission control that ensures services can be performed
- Result delivery
- Sensing service termination.

A requesting application needs to understand what sensing services are available in the requested area and when. The API may include additional information regarding likelihood of useful results and availability of resources. Further, the API guarantees coverage for the application (and lets the application know if coverage cannot be guaranteed).

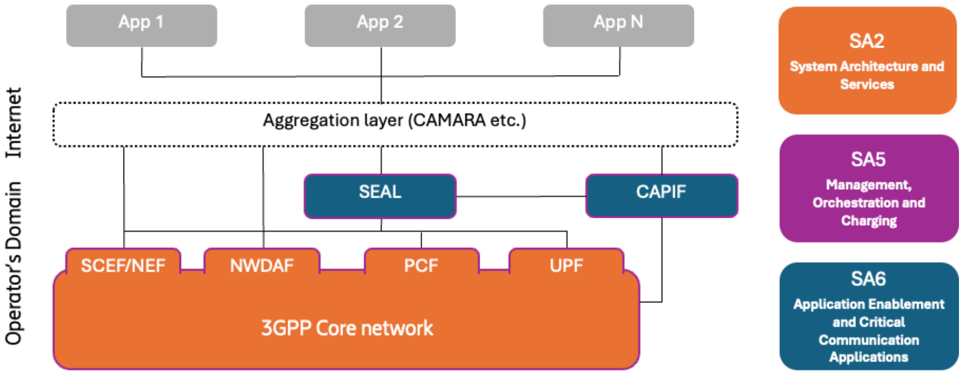

App 1
App 2
App N
Internet
Operator's Domain
Aggregation layer (CAMARA etc.)
SEAL
CAPIF
SCEF/NEF
NWDAF
PCF
UPF
3GPP Core network
SA2
System Architecture and Services
SA5
Management, Orchestration and Charging
SA6
Application Enablement and Critical Communication Applications

Figure 3-45 Network and service exposure in 3GPP-defined 5G networks.

3.2.4 Self-sensing approaches

This work highlights the experimental results derived from implementing the self-sensing solution outlined in D3.3 [HEX224-D33]. It provides an in-depth evaluation of the proposed approach, demonstrating its effectiveness and validating its potential for practical applications. Simultaneous Localization and Mapping (SLAM) [DNC+01] allows mobile robots to map environments and estimate their position in real time. The demand for SLAM is growing rapidly, driven by industries seeking to enhance productivity and reduce costs with autonomous robots. Traditional indoor mapping methods, such as LiDARs and cameras, deliver high accuracy but struggle in conditions like low light, fog, dust, or environments with glass surfaces [AZL+23], [FA18]. To address these limitations, mmWave sensing has emerged as a promising alternative. However, existing solutions often require multiple access points, rely on costly hardware, or involve high complexity and energy use. *waveSLAM* [PGB+23] is an indoor mapping system that uses sensing capabilities from off-the-shelf (COTS) 60 GHz mmWave radios integrated into mobile robots, operating on 2.16GHz bandwidth. It enhances mapping accuracy by combining mmWave radio self-sensing with LiDAR data, achieving centimeter-level precision without the need for external devices (Figure 3-46: Key intuition behind). This section explains the mmWave sensing techniques used for indoor mapping, focusing on Time of Flight (ToF) calculations through Fine Time Measurement (FTM) and Angle of Arrival (AoA) estimation using a Uniform Rectangular Array (URA).

ToF Calculation via FTM. The IEEE P802.11-2016 standard [802.11-2016] describes FTM, enabling two devices to estimate their distance without clock synchronization. The Initiator begins by sending an FTM request, which the Responder acknowledges. The procedure involves multiple bursts of measurements, each consisting of time-stamped packets exchanged between the devices. Using these timestamps, ToF is calculated by compensating for clock offsets and averaging measurements, where n is the number of measurements (eq.3-1) (see Figure 3-47 left):

$$\mathrm{ToF} = \frac{1}{2n}\sum_{x=1}^{n}(((t_4(x) - t_1(x)) - (t_3(x) - t_2(x))) \tag{3-1}$$

AoA Estimation with URA. The URA setup consists of K×J antenna elements (see Figure 3-47 right). Signals from the transmitter propagate through multiple paths, defined by parameters such as signal attenuation γ, elevation angle α and azimuth angle β, and W is gaussian noise. The multipath wireless channel is represented as (eq.3-2):

$$H = \sum_{p=1}^{P} H_p(\gamma_p, \alpha_p, \beta_p) + W \tag{3-2}$$

To extract these path parameters, the mD-track algorithm iteratively reconstructs the strongest signal path, subtracts it, and identifies weaker paths. This method, initially for linear arrays, was adapted for mmWave URAs. Beam training, involving beam-sweeping, is used to estimate the channel between devices [BMG+22].

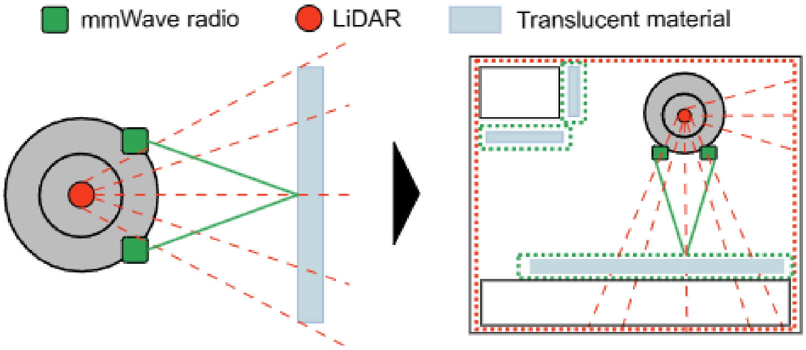


Figure 3-46: Key intuition behind WaveSLAM

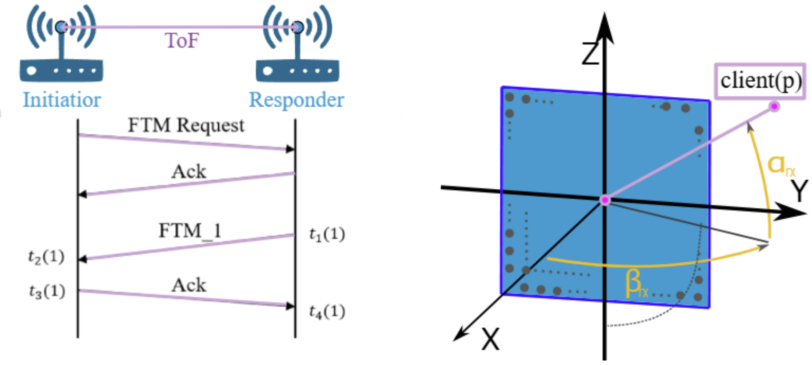


Figure 3-47: ToF exchange between two STAs (left) and AoA (α,β) estimations using an URA (right)

System design. Modern industrial robots rely on multiple sensors, such as cameras, LiDAR, and radio-frequency systems, for environment sensing and communication. Adding mmWave antennas enhances reliability, ensuring consistent data transmission even with obstructions. This solution integrates mmWave sensing with LiDAR-based SLAM systems to improve mapping accuracy.

In the data collection stage, mmWave devices mounted on the robot operate as Initiator and Responder, using self-sensing to provide ToF and CSI data. LiDAR measures distances by emitting laser pulses, while odometry tracks the robot's position over time. For mmWave sensing, the strongest signal path is identified through the mD-track algorithm, which extracts azimuth and elevation angles and computes the distance between devices. While mmWave is robust in translucent or high-light environments, it may face inaccuracies in narrow corridors or dense obstacle areas due to multipath effects. LiDAR, on the other hand, is highly effective in opaque environments but struggles with translucent materials or fog.

The system filters data from both sources to ensure reliable point clouds. When discrepancies arise, historical context helps determine the more accurate data source, with mmWave often compensating for LiDAR limitations in challenging scenarios. The SLAM algorithm then combines these filtered points with odometry to generate high-resolution indoor maps, improving mapping precision without altering existing SLAM methods.

Prototype design. To evaluate the feasibility of the system, a prototype was built using a Kobuki Turtlebot 2 as the mobile base. This robot carried an RPLiDAR A2 for LiDAR sensing and two mmWave devices separated by absorbing material to reduce interference and improve antenna directivity. These mmWave devices, minimized the risk of direct communication while maintaining effective sensing capabilities. A Raspberry Pi acted as the system controller, managing data collection, navigation, and storage while transmitting the raw data via Wi-Fi to a remote server for offline processing. The LiDAR and robot base connected to the Raspberry Pi through USB, while the mmWave devices used Ethernet. The system leveraged the Robot Operating System framework for seamless integration and navigation (see Figure 3-48 left).

The prototype used MikroTik wAP 60G as the mmWave devices, with OpenWRT firmware exposing ToF and CSI measurements. These measurements, alongside LiDAR and odometry data, were processed on a remote server. LiDAR point estimation relied on its SDK and ROS, while MATLAB scripts on the server calculated mmWave points by combining odometry data, ToF-derived distances, and CSI-derived angles. A filtering algorithm identified and corrected erroneous points by comparing LiDAR and mmWave results, ensuring reliable data input. For mapping, the prototype used the Hectormap SLAM algorithm implemented in ROS. It reconstructed 2D occupancy grid maps from the filtered individual points, combining data from LiDAR and mmWave to create precise indoor maps. This approach demonstrated the system's ability to integrate different sensing modalities for enhanced mapping accuracy (see Figure 3-48 right).

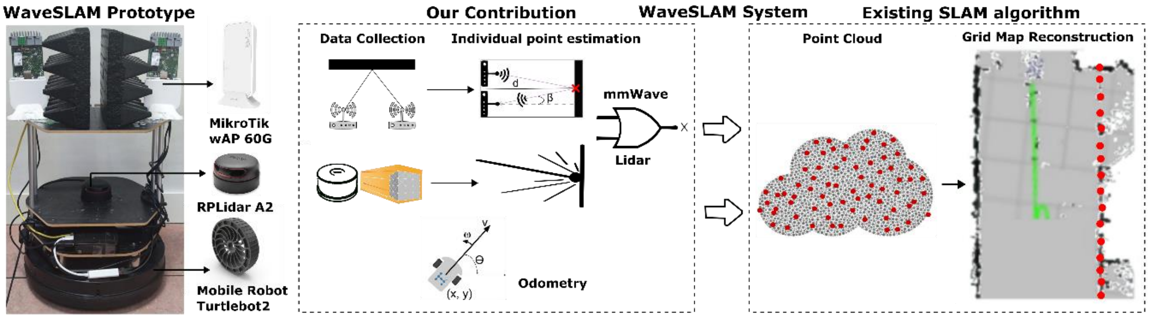


Figure 3-48: WaveSLAM prototype–(left) and system building blocks–(right)

Experimental results

To assess the capabilities, distance and angle estimations were evaluated. Distances between the robot and a wall were increased from 1 to 7 meters, and angle estimations were performed at different robot distances from the wall. The FTM distance error remained below 10 cm for 80% of the cases at all tested distances, though errors grew with longer distances due to increased attenuation and reflection sensitivity. The azimuth angle error also increased with larger rotation angles due to higher complexity and decreased antenna power. At distances up to 7 meters and angles up to 40 degrees, *WaveSLAM* provides accurate estimations, demonstrating the feasibility of using COTS mmWave devices for SLAM (see Figure 3-49).

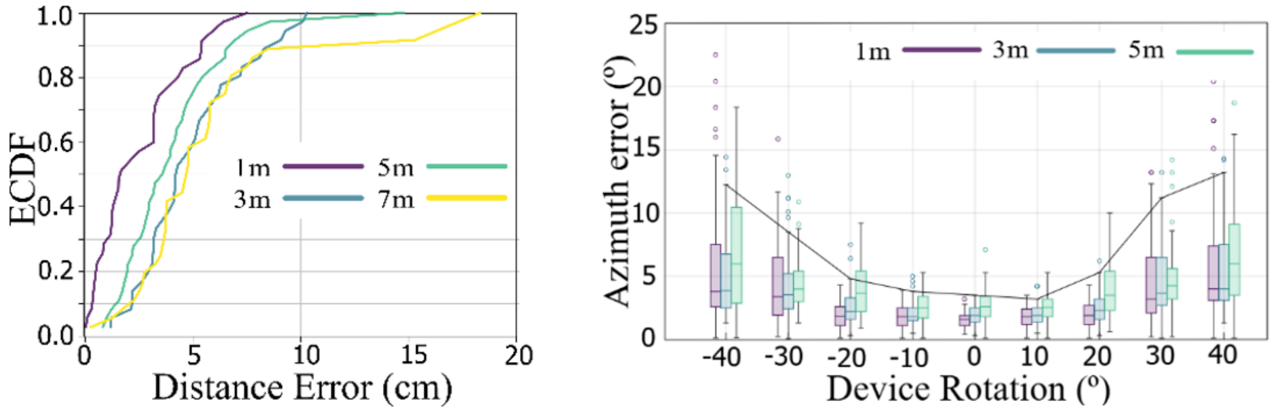


Figure 3-49: WaveSLAM capabilities: distance error (left) and angle error (right)

The map reconstruction accuracy of *WaveSLAM* was compared to LiDAR-based indoor mapping. The robot's path and position were adjusted so that mmWave devices aligned with target surfaces. The accuracy was assessed by comparing the mmWave points against a LiDAR. In a lab (Figure 3-50a), mmWave offered higher spatial resolution by detecting obstacles at different heights than LiDAR. In a dark corridor (Figure 3-50b), mmWave points complemented LiDAR without issues. In a light corridor with glass walls (Figure 3-50c), LiDAR failed due to sunlight interference and inability to detect transparent surfaces. However, mmWave detected the glass walls as opaque, improving map accuracy.

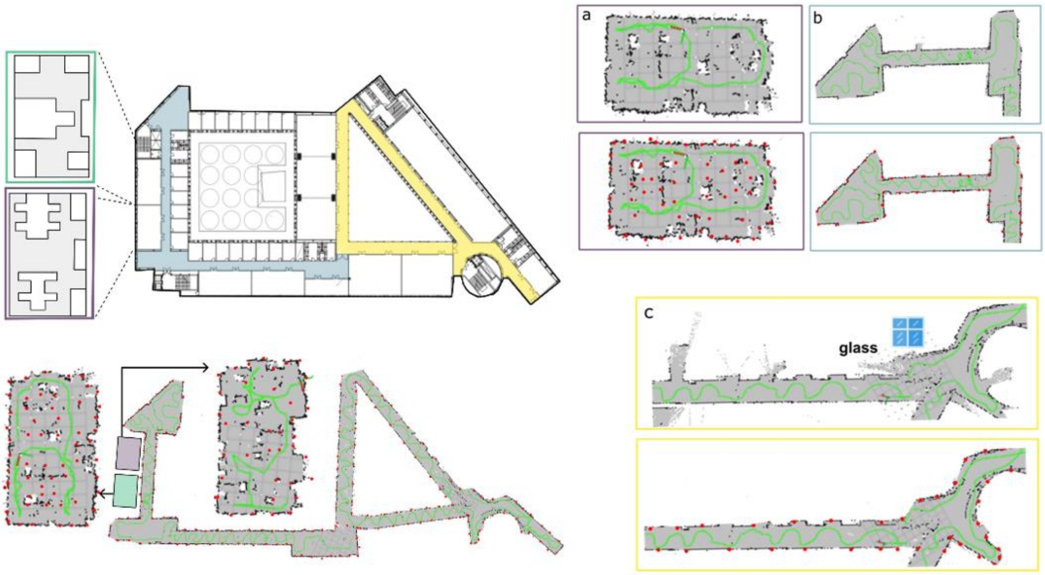


Figure 3-50: Map reconstruction (left), with zoom on closed lab (a), dark corridor (b) and light corridor (c).

Building indoor maps in low-visibility environments full of airborne obscurants is still a challenging problem today. To address this challenge, an efficient indoor mapping system called *waveSLAM* is developed, that improves the performance of existing state-of-the-art optical SLAM solutions. In particular, *waveSLAM* uses a pair of COTS mmWave radios to perform self-sensing and successfully map the indoor environment when the optical sensor fails. Table 3.8 compares distance measurements using mmWave self-sensing and LiDAR in scenarios involving walls and glass surfaces. **mmWave self-sensing** demonstrates its complementary nature to LiDAR by reliably measuring distances to glass walls, where LiDAR typically fails, producing infinite or highly erroneous values. In scenarios where LiDAR functions correctly (e.g., brick walls), mmWave self-sensing exhibits slightly lower accuracy, with a small range of variability. This highlights the advantage of mmWave self-sensing in overcoming LiDAR limitations, particularly in challenging environments with transparent surfaces.

Table 3.8: Self-sensing vs Lidar accuracy comparison

Real distance (m)	Measured distance (Self-Sensing) wall/glass (m)	Measured distance Lidar on wall (m)	Mesured distance Lidar on glass(m)
1	0.981-1.042	0.993-1.013	2.074-Inf (not detected)
3	2.978-3.054	2.984-3.015	4.014-Inf (not detected)
5	4.978-5.066	4.996-5.027	Inf (not detected)
7	6.922-7.081	6.990-7.027	Inf (not detected)

3.2.5 Summary of the ISAC enabler

Table 3-8 describes the summary and key take-aways for the sensing enabler.

Table 3-8 Summary of the ISAC enabler

Description	The ISAC enabler encompasses architecture enhancements and protocols to integrate sensing services in a communication system
Key take-aways	New functionality is needed: • to handle sensing requests, both in the form of an application that sends a sensing request and in the form of a function that receives and processes the request, • to further make use of the information in the request (the SeMF) and transform the request information to control and configure usable sensing units (e.g. units located in the correct places), suitable forms of exposure, (radio) resources needed for the actual sensing measurements. • to process the measurement data and provide output in a format useful for the requester. Existing means for data exposure may need to be enhanced to support the sensing measurement data in different scenarios. Overhead in terms of radio resources from ISAC appears to be small.
Requirements	The system shall be able to provide means to authorize and configure a UE for sensing operation (e.g., based on location, time, etc.) and for establishing the communication connection needed to assist the sensing service. The system shall be able to support means to enable RAN entities and UEs to transfer sensing measurement data to sensing processing entities in the 5G system responsible for processing and aggregation of the sensing measurement data.
Standard relations & regulations	TS 23.501 System architecture for the 5G System (5GS), TS 23.502 Procedures for the 5G System (5GS), 3GPP TSG SA WG 1, TS 22.137 Integrated Sensing and Communication

3.3 Compute offloading

3.3.1 Introduction

Compute offloading is a mechanism to move computation from one device to another with more suitable capabilities. It can offer offloading of critical tasks or functions to app developers, exposed as a network service and supports offloading of customized application modules. The offload solution aims to be dynamic and highly granular. This means that the application can dynamically offload certain of its functions (i.e., tasks) based on changes in the device (e.g., battery levels), network (e.g. radio quality) or application needs, see Figure 3-51. The offloading is initiated dynamically based on contextual and situational changes and takes advantage of network resources, processes, and information. One aim is to reduce computation times but also allow low-cost devices to perform advanced and computational heavy applications. Furthermore, offloading also allows to balance the compute and energy trade-offs of mobile devices, which prolongs battery lifetime. Finally, computational offloading can reduce device heat (which is specifically important for head-mounted devices) and facilitate synchronization and coordination among collaborating devices with a certain proximity [YWS+24].

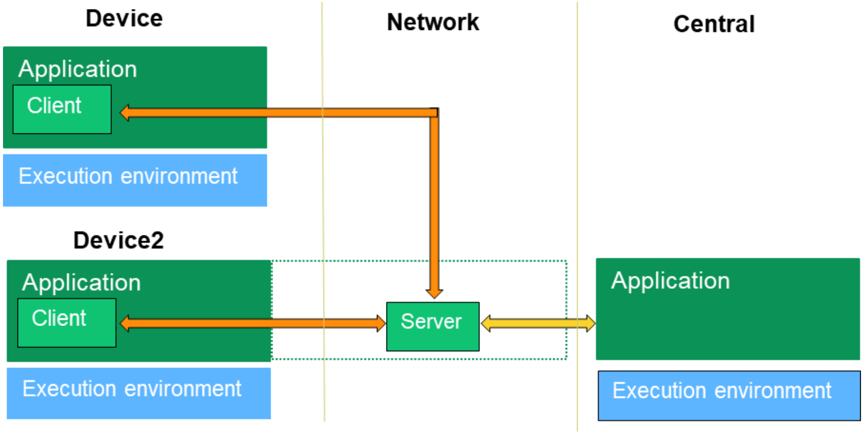


Figure 3-51 Compute device offloading concept from [HEX224-D33]

In the previous deliverables [HEX223-D32] [HEX224-D33], general architecture and novel functional entities for computational offloading are introduced. Moreover, common classification of computing and communication resources of each novel component, as well as a common characterization of offloaded compute workload based on predetermined requirements was introduced to satisfy the strict requirements on the computation and communication latency, trustworthiness, power consumption and data accuracy.

As stated in [HEX223-D32], the introduction of compute offloading in 6G should not increase the complexity of the communication protocols. In this deliverable, an overview of the offload framework concept is introduced in Section 3.3.1.1. It allows for interacting with the network without significantly disrupting RAN and NAS protocols, thus enabling faster time to market adoption, see section 3.3.1.1.

Moreover, in Section 3.3.2, the previously introduced staged procedure for computational offloading is further detailed with the introduction of detailed stages and service layers, which provide functions to maintain synchronization amongst nodes. Furthermore, the high-level message exchange for computation offload stage and its variants enabling the energy-efficient computation offloading by exploiting on the knowledge of the size of compute results are introduced and further detailed in 3.3.2.2 and Annex.

In Section 3.3.3, the scope of compute offloading is further expanded to non-terrestrial networks, specifically focusing on UAVs and Low Earth Orbit (LEO) satellites as compute nodes, examining their integration to support computation offloading for IoT devices deployed in remote or challenging environments. The game-theoretic solution for minimizing the computational offloading induced aggregate time and energy overhead of IoT devices is introduced.

3.3.1.1 Compute offloading framework

The focus of this contribution is on the architecture framework for compute offloading. The compute offloading architecture should allow mobile devices to offload computational tasks via the network to remote execution environments, while the actual offloading overhead should be minimized.

Figure 3-52 shows a draft of a possible compute offloading architecture. To minimize the overhead and also allow the possibility to offload to remote execution environments, the proposed framework is “loosely” integrated to the cellular network, somewhat similar to the approach taken in SA6 EDGEAPP [TS 23.558] (i.e., over the top from a network point of view).

The Offloading Node (ON) is responsible for handling the interaction with the UE application. It is also responsible to interact with the offloading service on behalf of the application. This includes discovery and registering with the network’s offloading service, represented by a Compute Control Node (CCN). The Offloading Node will also coordinate the actual offloading sequence in coordination with either the CCN, or directly with a local handler at the compute node (CompN), i.e., a compute cluster to execute offloaded modules. The CCN is the corresponding contact for the ON in the offload cluster. It receives the registration, authentications and request from the ONs, and coordinates with the resource manager to find a suitable CompN, typically based on the application requirements expressed in the request by the ON. The Resource Manager is responsible to keep track of the capabilities and availabilities of the allocated resources (i.e., CompNs) assigned to the offloading cluster.

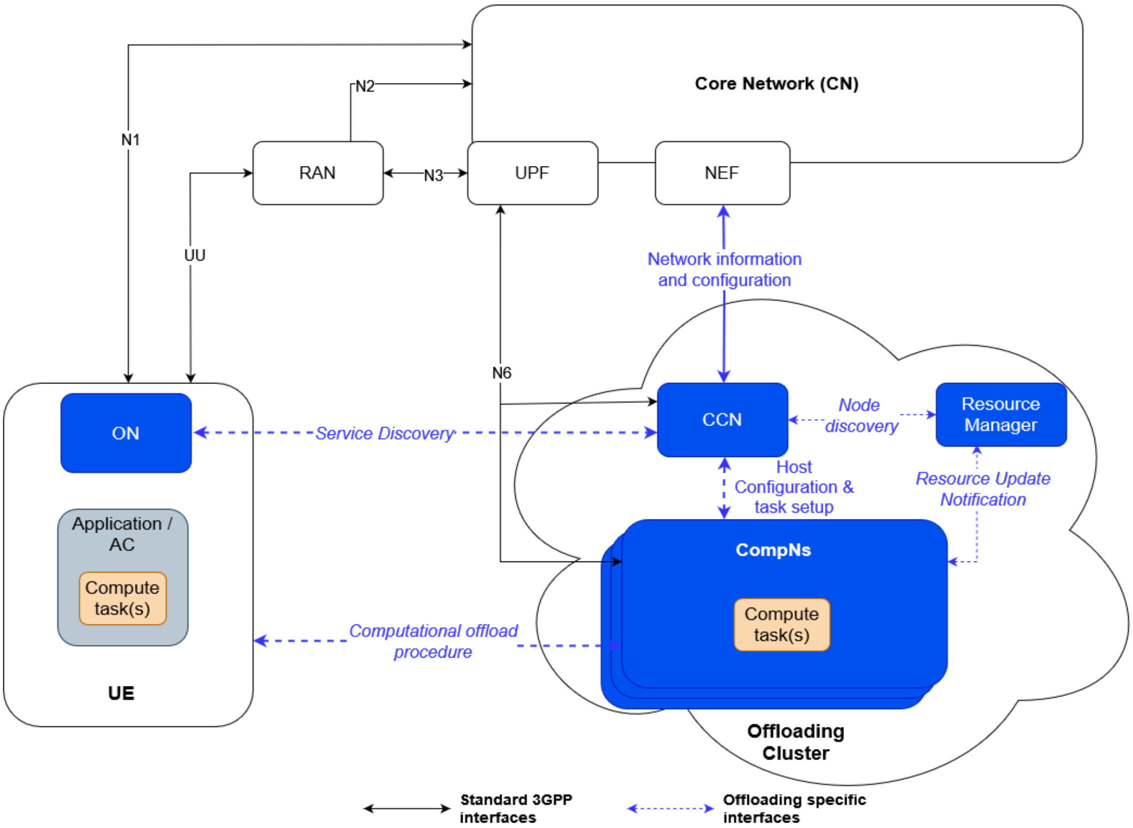


Figure 3-52 Draft architecture: offloading components interacting with the network

The proposed loose integration has the advantages that it requires no changes to RAN or NAS protocols and also allow access to remote execution environments. As a result, such loose integration is less disruptive with respect to existing standards and has no dependencies on UE chipset designs, opening up for faster time to market realization. On the other hand, it still requires richer interfaces and new procedures to interact with the network, for instance through AF via NEF. Some of these procedures are described in the following section 3.3.2.

3.3.2 High-level procedures for compute offloading and service layers

In [HEX224-D33], a staged procedure for computational offloading was introduced, which is further refined in Figure 3-53, to give a more detailed version with the introduction of further stages and service layers. After the “Stage#0: Initial Registration Procedure”, the compute node capabilities are identified in Node “Stage#1: Discovery Phase 1”, followed by the request for computation offloading in “Stage#2: Node Discovery Phase 2”, and, finally, the exchange of computing tasks and compute results in “Stage#3: Computational Offload Procedure”. The different newly introduced stages highlighted in green in Figure 3-53, are further detailed.

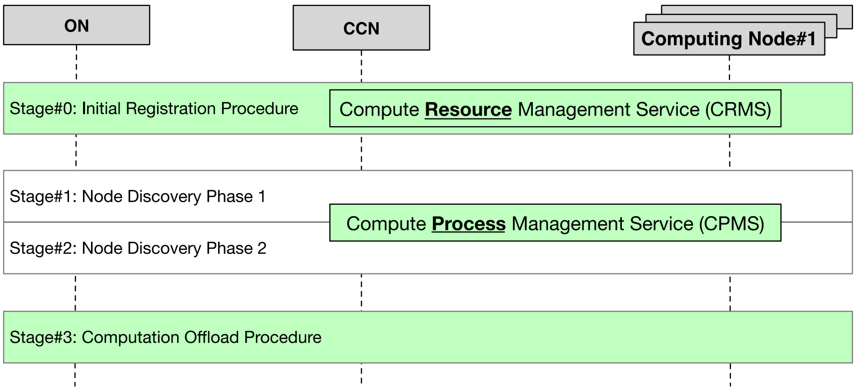

ON
CCN
Computing Node#1
Stage#0: Initial Registration Procedure
Compute Resource Management Service (CRMS)
Stage#1: Node Discovery Phase 1
Compute Process Management Service (CPMS)
Stage#2: Node Discovery Phase 2
Stage#3: Computation Offload Procedure

Figure 3-53: Computation Offload Procedure.

3.3.2.1 Compute service layers

The different stages of computation offloading are associated with service layers that exist amongst the nodes to establish a process and provide functions to maintain synchronization amongst nodes. As illustrated in Figure 3-53, two service layers for computational offloading are introduced. The first is the Compute Resource Management Service (CRMS) layer, responsible for registration, assignment and request of compute services and discovery of nodes. The other one is the Compute Process Management Service (CPMS) layer, which regulates the process management (i.e., the discovery and allocation of compute resources) and event management (i.e., maintaining the status of the requests).

The initial registration procedure in CRMS is shown in Figure 3-54, where different nodes register to CCN either for compute offloading request, i.e., ON, or to offer compute support (i.e., CompN), or both. The CRMS in CCN is responsible for checking the node registration requests and whether the node is allowed to provide or to request computation support, and further updates its database categorizing the node capability, compute capacity and the associated IDs. Each node can update its state in the CRMS and switch between requesting or offering compute resources. When a node no longer wishes to request offload or to provide compute support, it can also deregister from the CRMS. Examples of CRMS and CPMS databases are given in Annex.

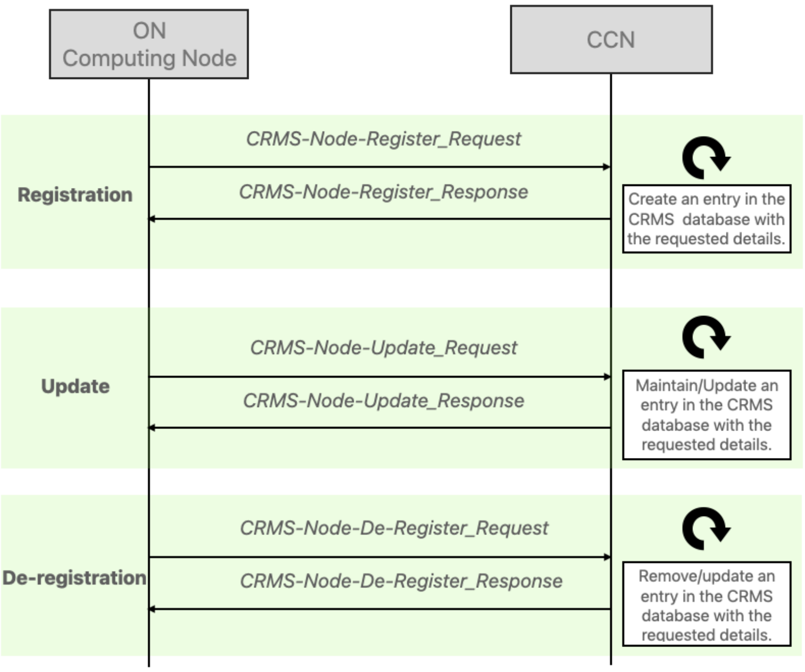

ON
Computing Node
CCN
Registration
CRMS-Node-Register_Request
CRMS-Node-Register_Response
Create an entry in the CRMS database with the requested details.
Update
CRMS-Node-Update_Request
CRMS-Node-Update_Response
Maintain/Update an entry in the CRMS database with the requested details.
De-registration
CRMS-Node-De-Register_Request
CRMS-Node-De-Register_Response
Remove/update an entry in the CRMS database with the requested details.

Figure 3-54: The initial registration procedure in CRMS.

3.3.2.2 High-level message exchange for compute offloading

The message exchange, foreseen in the Stage#1: Node Discovery Phase 1 and Stage#2: Node Discovery Phase 2, shown in Figure 3-53, was introduced in [HEX224-D33]. Here, the messaging exchange for Stage#3: Computational Offload, is described in the following. The initial registration procedure in CRMS is presented in Figure 3-54.

The Computational Offload Procedure is illustrated in the message sequence chart (MSC) in Figure 3-55.

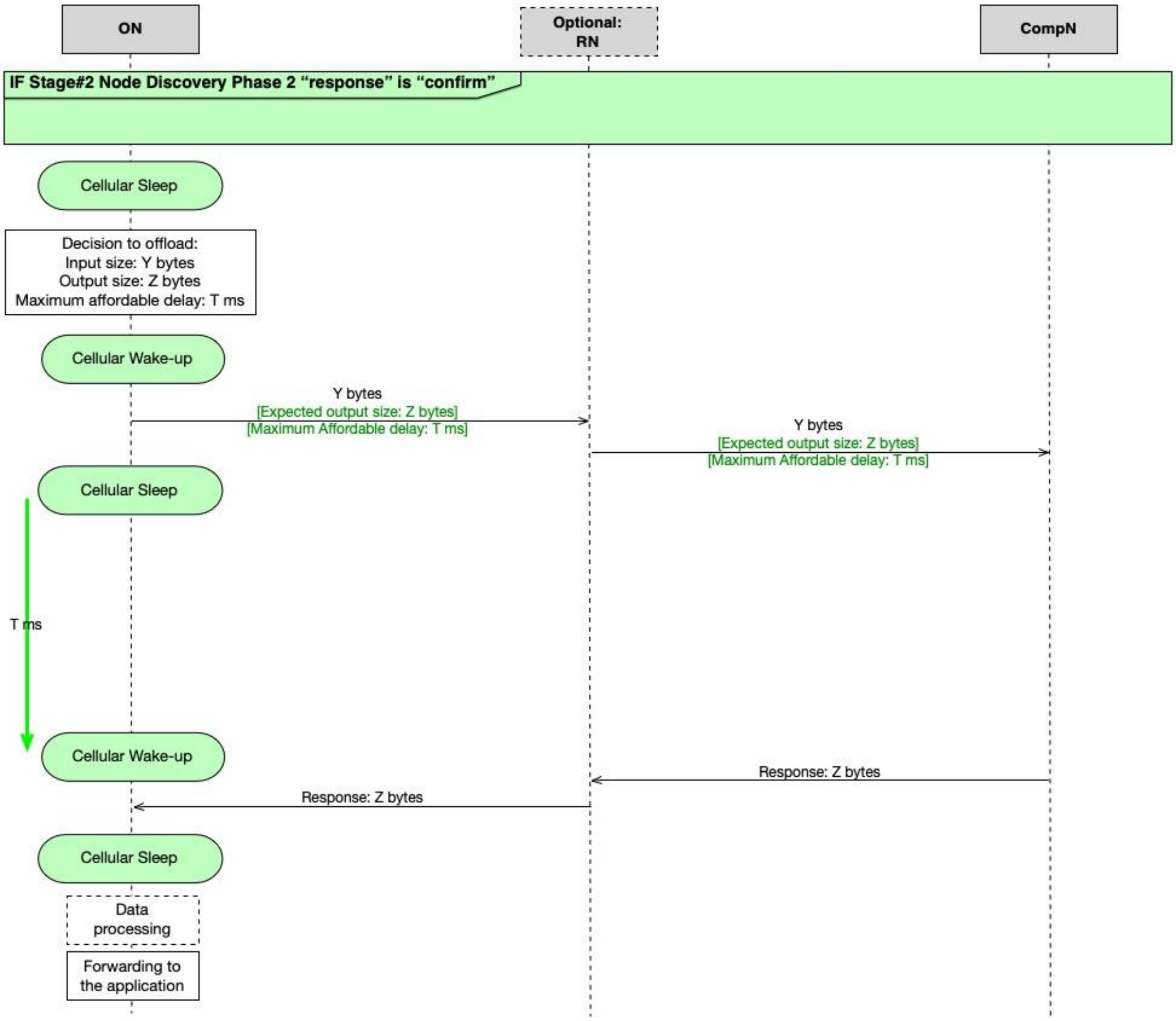

ON
Optional: RN
CompN
IF Stage#2 Node Discovery Phase 2 "response" is "confirm"
Cellular Sleep
Decision to offload: Input size: Y bytes Output size: Z bytes Maximum affordable delay: T ms
Cellular Wake-up
Y bytes [Expected output size: Z bytes] [Maximum Affordable delay: T ms]
Y bytes [Expected output size: Z bytes] [Maximum Affordable delay: T ms]
Cellular Sleep
T ms
Cellular Wake-up
Response: Z bytes
Response: Z bytes
Cellular Sleep
Data processing
Forwarding to the application

Figure 3-55: Computation offload procedure.

An ON in need of computation offload would request specific CompN(s) to perform the requested task(s) from the CCN (Discovery Phase 2). Computation loads and results can be directly exchanged between ON(s) and CompN(s) or routed over an optional Routing Node (RN), as given in Section A.4.1 . After sending a compute task of size Y [bytes], ON expects the compute results of size Z [bytes] within T [ms]. Being always *a priori* aware of Y, Z, and T, ON can enter power saving state after sending the compute task, thus avoiding unnecessary monitoring tasks or even unnecessary transmissions. Depending on whether the compute results size Z [bytes] is fixed or variable, there are two variants for the realization of the Computational Offload Procedure.

If ON is aware of the size of compute results, DL transmission of compute results can be planned before they are ready, i.e., via *configured* scheduling. This variant of proposed computation offload procedure enables increased sleep time of the ON, resulting in energy efficiency, and is further detailed in Section B.4. If the compute results are foreseen to have variable size, they may be sent via *dynamic* scheduling, such that the DL transmission is controlled by ON, as described in Section B.4.

3.3.3 Compute offloading decision-making algorithm

The application of computation offloading as a pivotal enabler for 6G networks has been extensively explored in the literature across various contexts, enabling compelling Extended Reality (XR) and Virtual Reality (VR) applications, autonomous vehicles, and remote industrial control. This section specifically investigates the integration of computation offloading within non-terrestrial and Unmanned Aerial Vehicle (UAV) networks, facilitating ubiquitous computing services for remotely deployed Internet of Things (IoT) devices located outside the coverage area of terrestrial base stations. It should be noted that UAVs are particularly suited for this role due to their unique characteristics, including flexible deployment, adaptive mobility, and distinct trade-offs between communication delays and computational capabilities. However, the broader challenge of compute offloading decision-making based on communication and computation requirements extends beyond UAVs and NTNs in general. The fundamental problem of compute offloading strategies is relevant and generalizable to various heterogeneous computation offloading settings and applications, where optimizing resource allocation is crucial for ensuring efficiency and seamless service delivery in dynamic network environments.

In more detail, consider a UAV-mounted Multi-access Edge Computing (MEC) server, hovering above an area of remotely deployed IoT devices. Also, consider a Low Earth Orbit (LEO) satellite serving as a relay to a cloud server. Due to the very limited computing resources onboard, the IoT devices fully offload part of their computation tasks for remote processing either to the edge server of the temporarily deployed UAV or to the satellite, which then forwards the data for processing in the cloud. Each computing option results in different transmission, propagation, and computing times, as well as energy costs. The UAV-mounted edge server has limited computing power compared to the cloud, causing delays in task processing, whereas offloading via satellite links incurs significant communication time and energy costs for IoT devices. Thus, a versatile decision-making process is required to balance this trade-off based on the network state and application demands. A high-level overview of the integrated network under consideration is presented in Figure 3-56.

In this context, the problem of determining the most beneficial amount of task to be offloaded to either the UAV or the satellite by each IoT device arises and is addressed. The goal is to minimize the aggregate time and energy overhead experienced by the IoT devices, stemming from computation offloading and remote processing at the UAV or the cloud through satellite relaying arises. Specifically, the computation offloading time is calculated as the sum of the transmission time to the UAV and the transmission and propagation times from the IoT devices to the satellite and from the satellite to the ground station. This total delay represents the end-to-end delay, which is also evaluated in the simulations. The computation offloading problem is formulated as a non-cooperative game among the IoT devices that compete for the shared pool of computing resources of the temporarily deployed UAV. This modelling approach enables the development of a distributed solution for computation offloading decision-making, where IoT devices are treated as players in a game, interacting with one another until the most beneficial offloading strategy in terms of total time and energy overheads is determined, ensuring that no device has an incentive to alter its strategy.

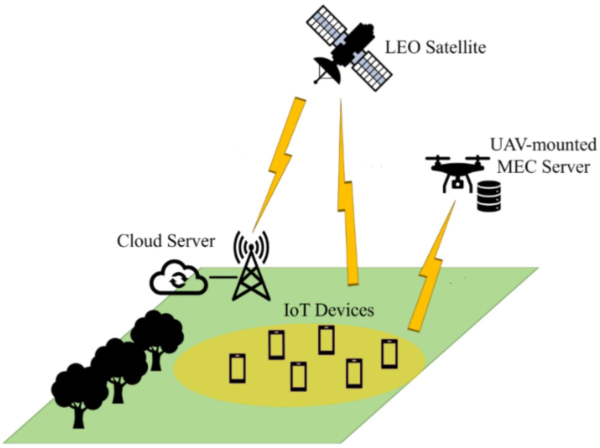


Figure 3-56: High-level overview of the integrated NTN/UAV network for computation offloading in remote IoT.

For the game-theoretic modeling and solution, two types of games and their respective equilibrium points are examined. First, the problem is formulated as a non-cooperative game in satisfaction form, where the devices do not explicitly target the minimization of their total time and energy overheads but strive to achieve an acceptable time and energy overhead tradeoff. This game concludes the so-called Satisfaction Equilibrium (SE) point, allowing for more flexibility in the computation offloading decisions of the IoT devices. Also, the typical non-cooperative game in normal form is considered that converges to the well-known Nash Equilibrium (NE) point, where each IoT device achieves the minimum possible aggregate overhead, given the decisions and offloading strategies of the other devices.

In the following, the performance of the computation offloading decision-making in integrated NTN and UAV networks is studied under the two equilibrium concepts, i.e., the NE and the SE. The numerical results are derived from modeling and simulation, considering 10 remotely deployed IoT devices with increasing computational workloads, as represented on the horizontal axis of Figure 3-57 by their device IDs. Figure 3-57(a) illustrates the percentage of each IoT device's task offloaded to the UAV under the NE and SE points. It is observed that as the IoT device ID increases, a smaller portion of its task is offloaded to the UAV in both equilibrium types. This occurs because tasks with higher computational intensity result in greater overhead in the event of UAV failure, as the UAV has limited computing power. In such cases, the task must be further forwarded to the satellite and the cloud, leading to increased overhead. Consequently, IoT devices with higher IDs tend to favor offloading to the more reliable and secure computing resources of the cloud via satellite relaying to meet their computational needs and avoid the risk of UAV failure. When comparing the two equilibria, it is observed that the SE achieves a fairer and more balanced utilization of the fragile CPR for all IoT devices, stemming from pure utility satisfaction rather than maximization.

Last, a scalability analysis of the solution under the SE point is performed by varying both the number of IoT devices and the aggregate task size threshold $\overline{C_U}$ in CPU cycles that the UAV can process concurrently without failure. Figure 3-58 verifies that as the number of IoT devices increases, the amount of offloaded data to the UAV decreases to effectively share the common pool of computing resources, whereas the opposite trend is observed for increasing value of $\overline{C_U}$. This is because higher values of $\overline{C_U}$ indicate a greater ability of the UAV-mounted edge server to process larger amounts of computation tasks in parallel. However, the opposite trend is observed for the IoT devices' mean incurred time and energy overheads. Specifically, the overhead increases with the number of IoT devices in the system and decreases as the parallel processing capability of the UAV gets higher regarding the parameter $\overline{C_U}$.

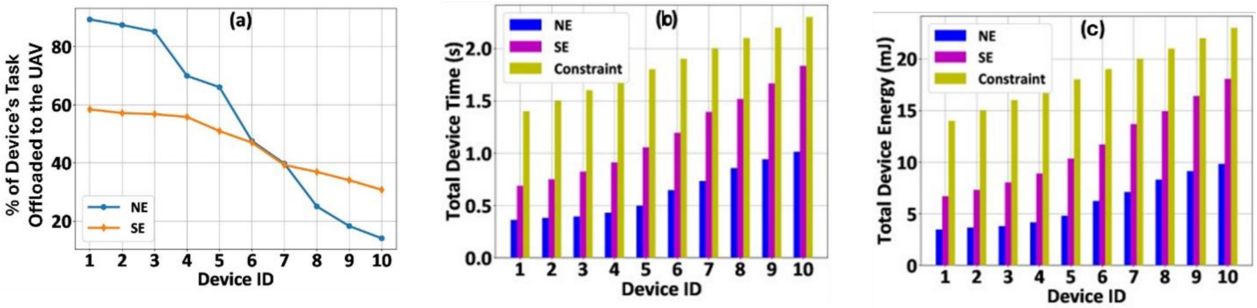


Figure 3-57: Pure operation for increasing values of computing workload in the system, indicated by the IoT device ID in the horizontal axis in terms of (a) percentage of device's task offloaded to the UAV, (b) total device time, and (c) total device energy.

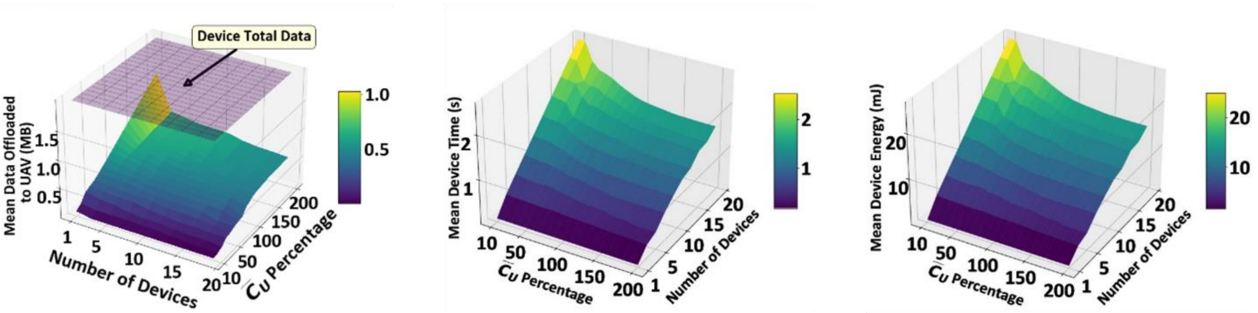


Figure 3-58: Scalability analysis for increasing numbers of IoT devices and aggregate task size threshold (C_U) for the UAV in terms of (a) mean data offloaded to the UAV, (b) mean device time, and (c) mean device energy.

3.3.4 Summary compute offloading

Table 3-9 summarises the compute enabler.

Table 3-9 Summary of the Compute offload enabler

Description	The compute offloading enabler deals with architecture enhancements and protocols to support compute offloading from mobile devices to compute nodes using network communication resources and, potentially, also computation resources. This is done while ensuring that the complexity of the communication protocol is not increased as well as satisfying the latency constraints, trustworthiness, power consumption, resilience. Dynamic offloading strategies, for computational offloading in non-terrestrial networks, balancing time, energy, and communication costs, while optimizing the resource sharing.
Key take-aways	Measurable KPIs: Decreased communication and computation latency, communications costs and power consumption, E2E latency Non-measurable KPIs: Computation resiliency, quality of computation/data accuracy, device complexity The proposed compute offloading architecture framework is "loosely integrated" in the cellular network, with no significant changes to RAN and NAS protocols, thus making it less disruptive to existing cellular standards, allowing for faster time to market realization. In addition, several procedures are detailed, for example on how to handle the offload node discovery, node registration and offload procedures. The framework and the procedures should ensure that both the QoE for the communication as well as the required resiliency and quality of computation are achieved, while preserving the energy efficiency.
Requirements	There is a need to support a connection between the device offload functions and the network offload functions. Whether the network functionality has to be standardized is for further study. Novel or extension of existing interfaces to interact with the network, e.g., via AF and the NEF. Identify network signaling needs to support proposed procedures. New roles of network components (UE, core network, RAN).

Standard relations & regulations	3GPP TS 23.501 System architecture for the 5G System (5GS)

3.4 Proof of Concepts

3.4.1 Optimized Placements

3.4.1.1 Optimized Application Placement

With the advance towards the realization of 6G networks, the convergence of flexible, trustworthy network architectures with beyond communication services (BCS) emerges as a pivotal requirement to address the multifaceted demands of next-generation applications. Building upon the foundations established in the previous deliverables, our approach synthesizes these components into a cohesive framework that not only enhances network resilience and adaptability but also ensures optimal performance and security across diverse use cases.

Dynamic Topology Management

Topology formulation is a critical aspect of this framework, ensuring that the network can dynamically adapt to changing conditions. The procedure begins with node discovery, where network components such as autonomous devices, access points, and sensors are identified and cataloged based on their capabilities, trust levels, and resource availability. This process enables the network to be flexible in its formation, with nodes being included or excluded dynamically based on real-time assessments. The node discovery process relies on a combination of trust evaluation metrics and cost-benefit analyses to determine the role of each node within the network, ensuring that nodes with higher trustworthiness and efficiency are prioritized.

The trustworthiness of nodes is evaluated through continuous monitoring of node behavior, and historical performance. Each node is assigned a trust score based on criteria such as data integrity, response times, and security compliance. This ensures that the network remains resilient, as untrusted or compromised nodes can be excluded from critical tasks, reducing the risk of data breaches or service disruptions. The trust management system not only enhances network security but also improves performance by dynamically optimizing node participation based on trustworthiness and resource efficiency.

$$\text{Maximize} \sum_{j=1}^{J} T_j Y_j - \lambda\left(\sum_{j=1}^{J} K_j Y_j + \sum_{j=1}^{J}\sum_{i=1}^{I} X_{ji} C_{ji}\right) \tag{4-3}$$

Where

$$T = \sum_{k=1}^{n} w_k \frac{f_{j,k} - f_k^{min}}{f_k^{max} - f_k^{min}} \tag{4-4}$$

Subject to:

$$\sum_{i=1}^{I} X_{ji} L_i \leq Cap_j Y_j, \forall j \tag{4-5}$$

$$\sum_{i=1}^{I} X_{ji} E_i \leq CapE_j Y_j, \forall j \tag{4-6}$$

$$X_{ji} \leq Y_j, \forall i, j \tag{4-7}$$

$$Y_j \in \{0,1\}, X_{ji} \in \{0,1\} \tag{4-8}$$

Moreover, resource optimization within this architecture leverages AI/ML models to predict traffic patterns, manage energy consumption, and allocate computational resources across the network efficiently. By anticipating network demands, such as sudden increases in traffic due to bursty applications or the need for additional computational power for data processing, the resource optimization algorithms ensure that the network operates at peak efficiency. This optimization is particularly vital in 6G environments, where latency, energy consumption, and bandwidth constraints must be balanced to maintain the quality of service across various applications.

Orchestration and AI-Driven Application Placement

The core of the approach is an integrated network architecture that joins flexible topology instantiation with BCS-driven service optimizations. The layered architecture depicted in Figure 3-59 illustrates the functional separation into three layers: Producers (Layer 0), Expose Network Functions (ENFs) (Layer 1), and Application Functions (AFs) (Layer 2). This layered architecture optimizes data flows for Beyond Communication Services by clearly defining the roles of each layer in data production, processing, and consumption.

- **BCS Data Producers (Layer 0)** generate essential data, such as sensor readings, telemetry, and location information, providing the raw inputs necessary for subsequent network functions.
- **Expose Network Functions (ENFs - Layer 1)**, like the Sensing Function, Quality on Demand, and Device Location ENFs, act as intermediate nodes, transforming the raw data into actionable insights required by application functions. These functions enhance the adaptability of the network by ensuring data is processed and made available in real time.
- **Application Functions (AFs - Layer 2)**, such as Navigation and Object Detection AFs, utilize the processed data from ENFs to execute higher-level applications, demonstrating the framework's capability to support complex and latency-sensitive services like autonomous navigation and real-time object detection.

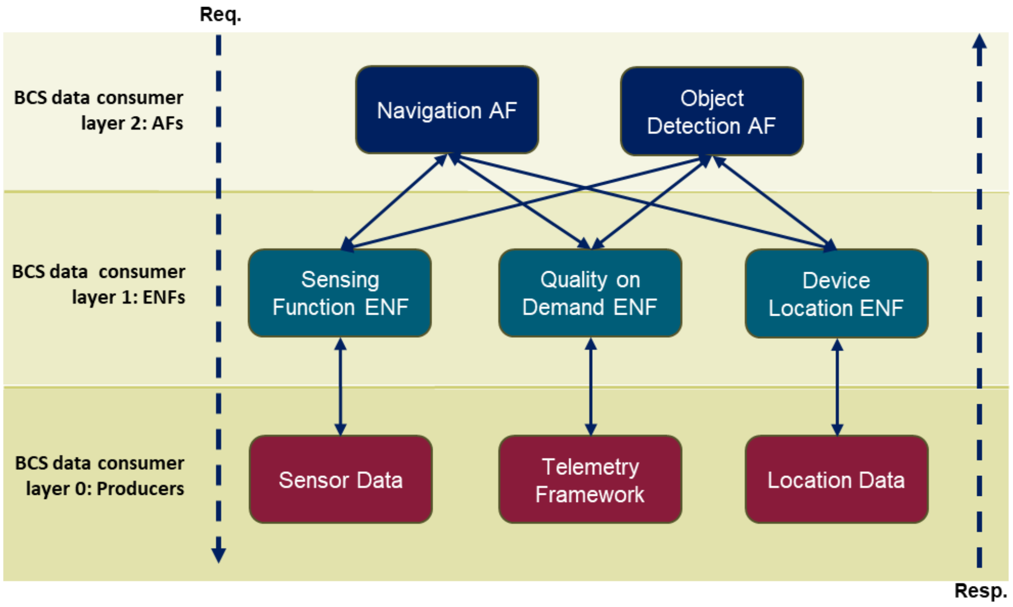


Figure 3-59 Application-Layer and Network Function Interactions in 6G BCS Optimization.

As depicted in Figure 3-64, the framework integrates an Orchestrator that manages both compute resources and application placements. The orchestrator plays a central role in ensuring that the data exposed by the ENFs, such as Quality on Demand, Sensing, and Device Location, are accessible by the respective AFs of the services. These functions are responsible for enabling key services such as real-time navigation guidance and accurate object detection and tracking, among others, ensuring that the applications running on the network meet their performance and QoS requirements. Under the orchestrator, an AI Application Placement Module enables real-time decision-making, determining where applications should be deployed within the network. This AI module leverages the network data exposed via the ENFs to optimize application placement in layer 2, balancing factors such as latency, energy consumption, resource availability, and application performance. By using AI-based node selection strategies, the orchestrator ensures that applications are allocated to the optimal compute continuum locations, maximizing both resource efficiency and service quality. This orchestration framework provides the flexibility required to dynamically manage application lifecycles and resource utilization across a heterogeneous 6G infrastructure. The AI-driven placement mechanism offers a scalable solution for handling diverse use cases, ensuring that BCS applications can operate seamlessly under varying network conditions and performance requirements.

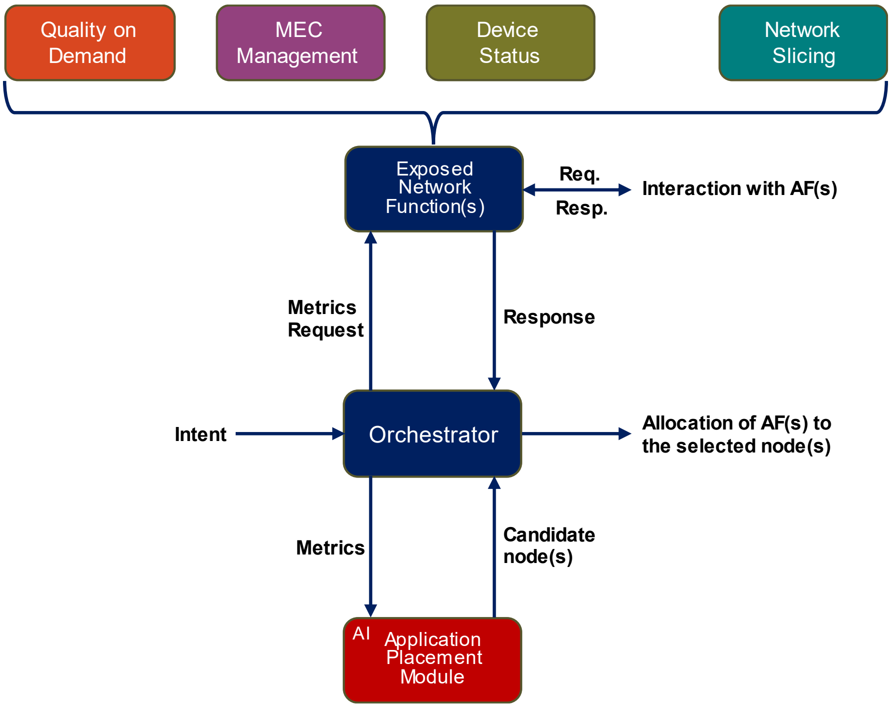


Figure 3-60 Orchestration of 6G Network Capabilities Using CAMARA-Compliant APIs

Warehouse Inventory Audit Scenario

To validate the efficacy of this holistic framework, a PoC within a warehouse inventory audit scenario was implemented. This scenario involves a cluster of Autonomous Mobile Robots (AMRs) and UAVs operating within a dynamic warehouse environment that pose significant challenges for network coverage and stability. The PoC demonstrates the deployment of a Flexible Topology Node (FTN) that dynamically adjusts its position to maintain network connectivity and support autonomous operations of the worker nodes. In instances of local connectivity loss or out-of-coverage tasks, the FTN autonomously repositions itself using AI-driven algorithms to restore network connectivity and ensure uninterrupted task execution. This dynamic adjustment is facilitated by real-time data from node performance metrics and trust evaluations, enabling the FTN to make informed decisions about optimal placement and resource allocation. The successful deployment of this flexible topology within a complex industrial scenario underscores the practical viability of our approach, ensuring both the operational efficiency and security of 6G networks. Implementation and results can be found in coming deliverable D2.6.

3.4.1.2 Optimized Service Placement

3.4.1.2.1 Introduction

In modern Edge Computing (EC) environments, applications that access services deployed across various compute sites must adhere to stringent network and computational requirements. Traditional approaches address these demands by deploying multiple service instances at different locations and selecting the most suitable instance based on factors like proximity and resource availability. However, as application demands increase and pre-deployed services may not always align optimally with user needs, there is a growing necessity for more dynamic and efficient resource management strategies. To tackle this challenge, we introduce the Integration of Network and Compute (INC) approach, which jointly optimizes network and compute resources through an innovative decision-making functionality. This approach not only selects the optimal existing service instance but also enables on-demand deployment of services based on real-time network and compute metrics. As provided in the next subsection, evaluations using a DQN-based reinforcement learning algorithm demonstrate that the INC approach outperforms traditional proximity-based and random allocation methods, particularly under high request loads, by ensuring higher satisfaction of application requirements and more balanced resource utilization.

3.4.1.2.2 Service Placement Procedure

Different applications, where a user is accessing a service deployed in a compute site, are associated with stringent requirements in terms of network and compute. Addressing these requirements would require a new approach to jointly optimize these two resources. In a traditional scenario, such as in Edge Compute (EC), a service has multiple instances that are deployed at different locations, and achieving the stringent requirements of the application would be translating in selecting an optimal service instance that satisfies both network and compute requirements formulated by each application. A more advanced scenario would enable on-demand deployment of services as per the request of the application. Motivations for such scenario include the following: i) an optimal compute site that satisfies the application requirements may not have the requested service pre-deployed, ii) reduce resource consumption by running a service only when required. The aim is therefore to select a compute site, that satisfies network and compute requirements of the application, where the requested service needs to be deployed. These two scenarios (service selection and service deployment) are illustrated in Figure 3-61: Scenarios for selection/deployment of .

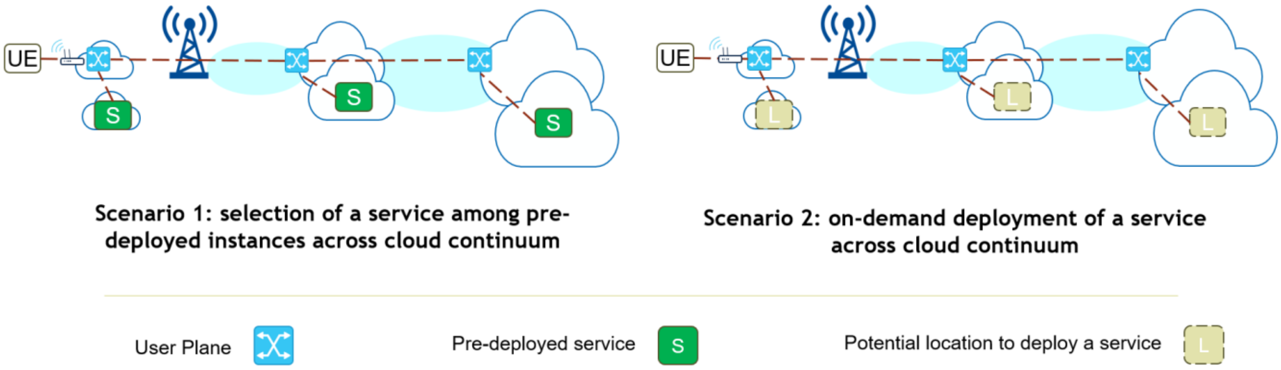

UE
S
S
S
UE
L
L
L
Scenario 1: selection of a service among pre-deployed instances across cloud continuum
Scenario 2: on-demand deployment of a service across cloud continuum
User Plane
Pre-deployed service
S
Potential location to deploy a service
L

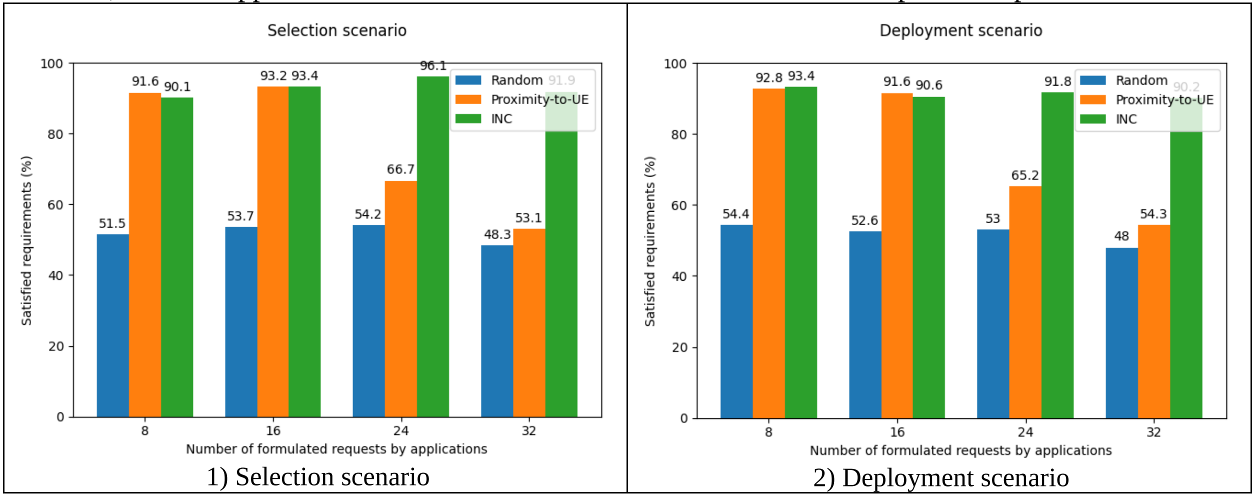
Figure 3-61: Scenarios for selection/deployment of services

In order to address the previous scenarios and ensure optimal selection of service instance or a compute site, we have introduced the INC approach that aims to jointly optimize network and compute processes in a way to meet application requirements. This approach implies the introduction of a new functionality that is responsible for decision-making. The generic procedure for the INC approach is as follows:

- Metric collection: the INC functionality acquires network and compute metrics. The network metrics are primarily related to the delay to target compute sites. The compute metrics are across the continuum cloud and are related to the running services (resource utilization of the service in case of service selection scenario) or the target compute site (resource availability of the compute site in case of service deployment scenario).
- Request formulation: the application formulates a request for service selection or service deployment. The request includes network and compute requirements and is received by the INC functionality.
- Decision-making: based on network and compute requirements formulated by the application, and network and compute metrics already collected, the INC functionality decides on the optimal service instance to serve the application, or the target compute site to deploy the service.
- Decision enforcement: the INC Functionality formulates a request for decision enforcement from the target compute site (e.g., deployment of the service) and the network (e.g., user plane configuration across the cloud continuum).

To evaluate the proposed INC solution, we have considered a DQN-based reinforcement learning algorithm to perform optimal decision-making by the INC functionality. Furthermore, we have considered two baseline approaches. The first one is based on random allocation, where the decision for selecting a service instance (scenario 1) or a compute site (scenario 2) among extreme-edge and edge clouds is made randomly. As for the second baseline approach, and similar to the current specifications in EC, the selection of a service instance (scenario 1) or a compute site (scenario 2) is based on the proximity to the UE. The evaluated topology consists of a mobile network, connected to an edge-cloud with enough resources, serving 4 Access Points. Each of the latter serves 2 UEs and is connected to an extreme-edge cloud with capacity to accommodate 4 services. The UEs formulates requests, to select or deploy services, while expressing their network and compute requirements.

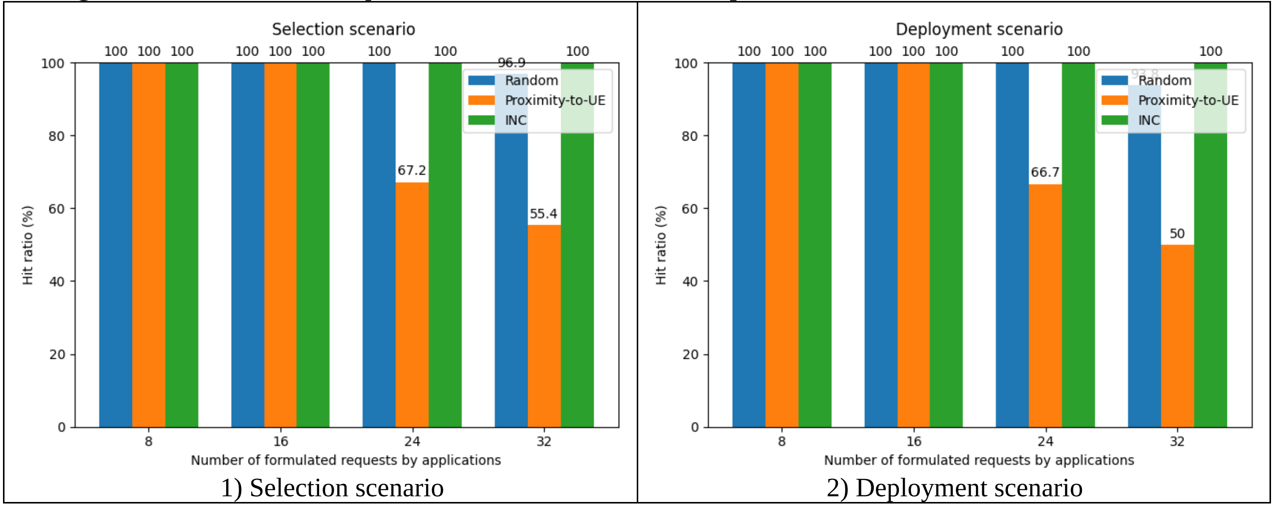
We have evaluated the ratio of satisfied requirements across various number of requests formulated by the users. The obtained results are depicted in Figure 3-62: evaluation of ratio of satisfied . As we can see from this figure, both INC and Proximity-to-UE outperform random approach. In addition, for smaller numbers of requests, Proximity-to-UE and INC provide similar results. This is valid for the two scenarios. This is explained by the fact that the amount of compute resources available in the extreme-edge cloud is enough to accommodate all the requirements of the applications (i.e., delay is always small). However, as the number of formulated requests increases, the proposed INC approach outperforms the Proximity-to-UE solution. Indeed, as the extreme-edge cannot satisfy all the requirements, a global optimization that takes into account both network and compute requirements is needed. Unlike the Proximity-to-UE solution that selects the nearest locations, the INC approach collect metrics from the different locations and perform optimal decision.



1) Selection scenario 2) Deployment scenario

Figure 3-62: evaluation of ratio of satisfied requirements.

We have also evaluated the hit ratio, which is defined as the ratio that the selected service has enough resources to serve user (scenario 1) or that the selected site has enough resource to deploy the service (scenario 2). The obtained results are shown in Figure 3-63: evaluation of the hit . For a small number of requests, all three approaches (proximity-to-UE, INC, and random) achieve a high hit ratio. However, as the number of requests increases, the hit ratio of the proximity-to-UE approach declines. This is because proximity-to-UE prioritizes the nearest sites, leading to unbalanced resource utilization. While both INC and random approaches maintain a high hit ratio with a large number of requests, the INC approach ensures optimal resource allocation, resulting in a higher ratio of satisfied requirements, as demonstrated in previous evaluations.



1) Selection scenario 2) Deployment scenario

Figure 3-63: evaluation of the hit ratio.

3.4.1.3 *Summary of Optimized Placement Enablers*

Optimized application placement utilizes dynamic topology management and AI-driven orchestration to deploy applications in optimal locations, balancing critical factors such as latency, energy consumption, and resource availability while ensuring network security through continuous trust evaluations. Concurrently, optimized service placement employs the INC approach, which leverages reinforcement learning algorithms to jointly optimize network and computational resources. This ensures that stringent application requirements are met, and that resource utilization remains balanced and scalable under varying demand conditions. Table 3-10 summarizes the findings of the optimized placement enabler.

Table 3-10 Summary of the Optimized placement enabler

Description	The optimized application placement approach utilizes dynamic topology management and AI-driven orchestration to deploy applications in optimal locations. This approach balances critical factors such as latency, energy consumption, and resource availability while ensuring network security through continuous trust evaluations. The INC approach aims to jointly optimize network and compute processes in a way to meet application requirements. This approach implies the introduction of a new functionality that is responsible for collecting network and compute metrics, receiving application request (including delay and compute requirements), and deciding the optimal service or deployment location.
Key take-aways	The optimized application placement procedure ensures optimal deployment of applications by balancing latency, energy consumption, and resource availability. It also enhances network resilience and performance through dynamic topology management while maintaining security via continuous trust evaluations and monitoring. For smaller numbers of requests, Proximity-to-UE and INC provide similar results (edge-clouds that are in the proximity to the UE can accommodate all the requirements). As the number of formulated requests increases, the proposed INC approach outperforms the Proximity-to-UE solution (nearest edge-clouds cannot satisfy all the requirements, and a global optimization that takes into account both network and compute requirements is needed)
Requirements	The optimized application placement approach requires the collection and analysis of network metrics such as latency, energy consumption, and resource availability. It also necessitates AI-driven orchestration and decision-making capabilities, along with a robust trust management system for continuous monitoring and evaluation of network nodes. The INC approach requires the collection of network and compute metrics. It also requires the collection and the exposure of network and compute metrics.
Standard relations & regulations	3GPP TS 23.501 System architecture for the 5G System (5GS)

3.4.2 Component PoC #B.2: Distributed Model training and inference

The impact on energy consumption and computation overhead by offloading two NN model layers between generalization node and output nodes is investigated. The energy estimator model that is briefly presented in Section 3.1.2.2 is deployed into the distributed learning clients to quantify the energy savings in an offloading scenario between network and application. Figure 3-64 and Figure 3-65 present the signalling that occurs between the Generalization Node (e.g., network) and the Output Nodes (e.g., external application), where these nodes are previously introduced in Section 3.1.2.2.

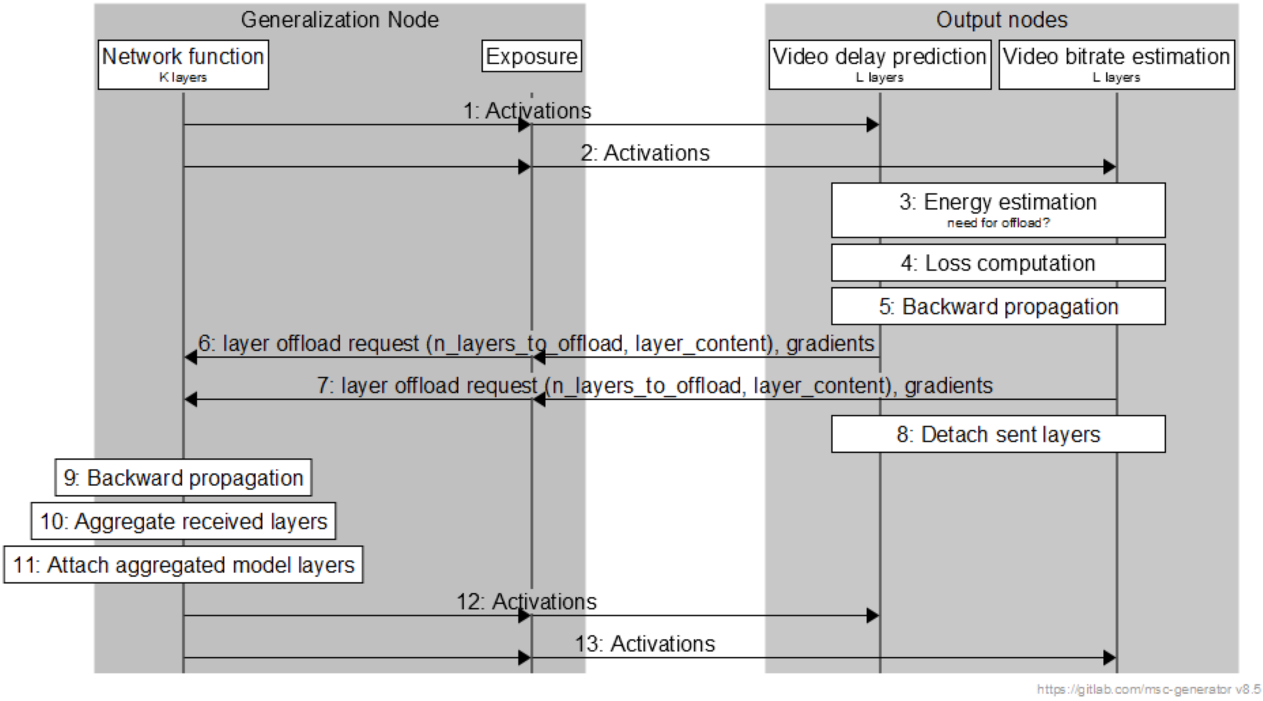


Figure 3-64: Signalling diagram for offloading of NN model layers from Output Nodes to the Generalization Node. Generalization Node is illustrated as Network. Output node is illustrated as Application.

In Figure 3-64, during a model training in an vFL setting, activations, which are the learned and extracted common representations for delay and video bitrate estimation tasks, are sent from Generalization Node at the Network to the Output Nodes at the external application. Example scenario is that an application detects that there is low battery level on the hosting device and the energy requirement that is estimated to perform the training is high. Application then sends a request to the Generalization Node. The offload request contains information such as the number of layers. In step 8, the output nodes detach the offloaded layers (first 2 layers of their local NN models in this example). The Generalization Node receives the offloaded NN model layers from the output nodes. As there are two tasks running in two output nodes, the weight matrices received for both tasks need to be aggregated at the Generalization Node. The aggregation is performed via averaging operation. The aggregated model layers are then attached as last 2 layers of the Generalization Node. This aggregation process helps to save energy as it scales with the number of offloaded layers. However, the model performance is expected to be impacted due to the averaging of the model layers that were received from the Output Nodes.

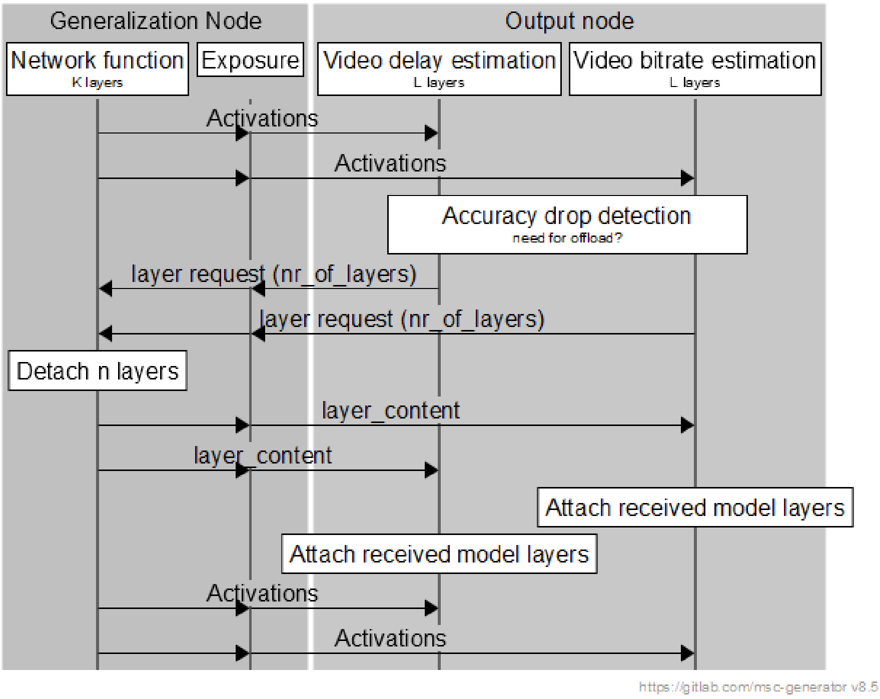


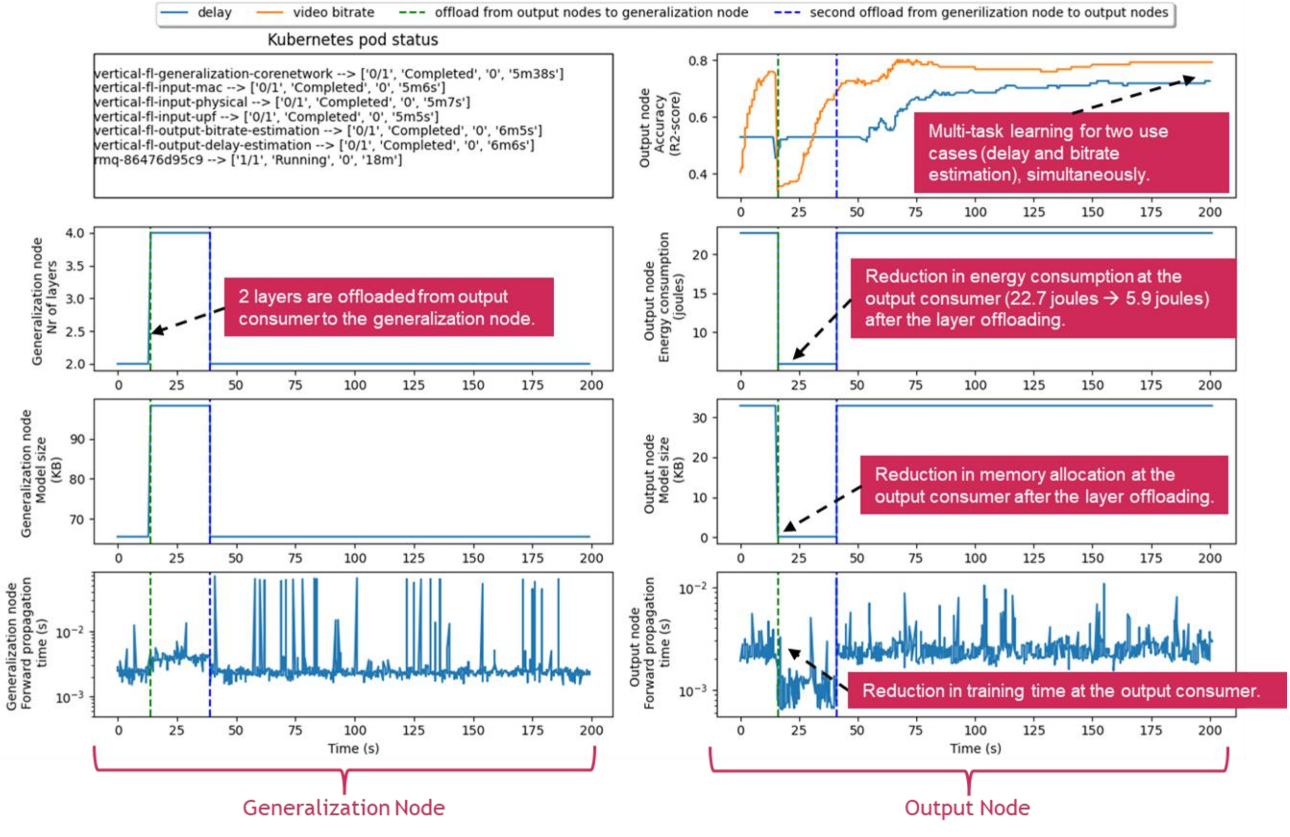
Figure 3-65: Signalling diagram for offloading of NN model layers from Generalization Node to the Output Nodes. Generalization Node is illustrated as Network. Output node is illustrated as Application.

When there are sufficient resources at the application, the output nodes at the Application may request to receive back some model layers from the Generalization Node. It might be that Output Nodes do not converge fast with a shallow local NN model. In that case, Output Nodes send request to the Generalization Node to offload some model layers as illustrated in Figure 3-65. In the request message, it also includes information on how many layers it wants to receive from the Generalization Node. Generalization Node acknowledges the request by detaching the requested number of layers from the end of its local NN model, and then delivers the model layers to the Output Nodes in the Application. This delivery is a broadcast, since the same model layer content is sent to both Output Nodes. The Output Nodes then attach the received model layers to the beginning of their NN models, and the training continues. Observe that in this case, the NN model weights are not aggregated and remained unchanged, hence no degradation of model performance is expected.

We start the demonstration by deploying 7 Kubernetes pods into the Kubernetes cluster. One pod for enabling the communication between the pods via a message bus; two pods for delay estimation and video bitrate estimation tasks at the output nodes (potentially can be deployed at the external application); one pod at the generalization node (potentially can be deployed at the network); and three pods for the input features that receive raw data features that is locally available. The input nodes have the input features but not the output label; while the output nodes only have the output labels but not the input features. They do not share model or local raw data in between each other.

The snapshot result from the demonstration of PoC#B.2 is illustrated in Figure 3-66. During the training, two layers of NN model are offloaded from the output nodes to the generalization node as illustrated via green vertical dashed line. The model size in bytes has increased from below 70KB to above 90KB at the generalization node after receiving the offloaded model layers. The model size at the output nodes decreased from 32KB to 128B. This offload increased the forward propagation time at the generalization node but decreased it more in the output node, which resulted in overall decrease in the training time. When two layers of NN model layers were offloaded from output nodes to the generalization node, based on results obtained from 5 iterations of experiment, the mean change in forward and backward propagation times at one output node were observed to be x0.3 and x0.13, respectively. The mean change in the generalization node after receiving the offloaded layers was insignificant, x1.05. The reason for this is that, in realistic settings, the generalization node is expected to run in an environment with large computation resources that are already running many other tasks. Therefore, additional computation yielded by additional two layers of NN model does not increase the computation time as it impacts the output node that is running on a computationally weaker device. Moreover, the energy consumption at the output nodes decreased from above 22.7 joules to below 5.9 joules. Such decrease in the energy consumption and computation overhead at the output nodes would increase the battery lifetime of devices that are hosting the applications running on them. One interesting observation is that the model accuracy, i.e., R2 score, is negatively impacted by the layer offload from output nodes to the generalization node. The reason for this is the aggregation of offloaded model layers at the generalization node before attaching to the end of the existing NN model, which impacted the training due to the change in the model weights. The need for aggregation is not obligatory. Therefore, a second alternative would have been to keep the offloaded layers from two different head nodes separate and not aggregate at the generalization node, which then would necessitate additional storage for the offloaded layers. In most cases this is OK since generalization node is expected to be hosted at a computationally rich environment such as network. The latter would limit overall energy savings.

After a few rounds of training at approximately *t* =25 s, the model weights are readjusted via few rounds of gradient and activation exchange and keeps on learning. Eventually both estimation tasks reach a convergence at around *t* =200 s of training, realizing a joint optimization of the two tasks, simultaneously. At approximately *t*=35 s as illustrated via blue vertical dashed line, two layers from the generalization node are offloaded, via broadcasting as the same copy of the model layers are offloaded to two output nodes. In this case, we do not see an impact on the R2-score. This is expected because the model layers thus the model weights just relocated without changing its content and the order of connection.



```
vertical-fl-generalization-corenetwork --> ['0/1', 'Completed', '0', '5m38s']
vertical-fl-input-mac --> ['0/1', 'Completed', '0', '5m6s']
vertical-fl-input-physical --> ['0/1', 'Completed', '0', '5m7s']
vertical-fl-input-upf --> ['0/1', 'Completed', '0', '5m5s']
vertical-fl-output-bitrate-estimation --> ['0/1', 'Completed', '0', '6m5s']
vertical-fl-output-delay-estimation --> ['0/1', 'Completed', '0', '6m6s']
rmq-86476d95c9 --> ['1/1', 'Running', '0', '18m']
```

Multi-task learning for two use cases (delay and bitrate estimation), simultaneously.

2 layers are offloaded from output consumer to the generalization node.

Reduction in energy consumption at the output consumer (22.7 joules → 5.9 joules) after the layer offloading.

Reduction in memory allocation at the output consumer after the layer offloading.

Reduction in training time at the output consumer.

Figure 3-66: Snapshot from the live illustration of the state of deployed pods (row=1, col=1), the nr of model layers at the generalization node (2,1), model size in bytes (3,1 and 3,2), energy consumption at the output nodes (2,2), the forward propagation time in the generalization (4,1) and output nodes (4,2), as well as the accuracy of the delay and bitrate estimation tasks(1,2).

Although presented results above are from a snapshot experiment, it is important to mention that this would be beneficial for two E2E scenarios (autonomous operation and cobot based video surveillance) that are part of the system PoC.

- In the autonomous operation scenario, it is well applicable to resource allocation workflows, and optimal placement and execution of scenario's computational tasks. In the scope of the section, a computation task is defined as "split NN model training and inference". By continuously monitoring performance, energy consumption, NN model layers can be offloaded via functionality allocation mechanism.

- In the zero touch Cobot based video surveillance scenario, the offload of NN model layers can occur either between Cobots of different communication and compute capabilities, or between the management platform and the Cobots.

3.5 Fulfilment of the Novel Services Objective

The work presented in this chapter directly addresses and fulfils the overall goals of WPO3.1, which require developing and analysing a 6G architecture framework that is both AI-driven and capable of beyond communications services. To achieve this aim, the chapter proposes novel architectural enhancements and specific enablers that equip 6G networks with data-centric workflows, sophisticated AI capabilities, integrated sensing, and flexible compute offloading functionality. These additions transform 6G networks into more than standard connectivity platforms, enabling full-scale automation, real-time intelligence, and the seamless management of heterogeneous resources at the edge and in the cloud.

The proposed architecture is based on newly introduced or extended functions, such as the DataOps for handling data lifecycle tasks, the MLOps for end-to-end AI workflow management, the AIaaSF for exposing AI functionalities, and the SeMF for coordinating integrated sensing procedures. Existing network elements operate alongside these new entities in a cohesive environment that aligns with standardization frameworks, such as those specified by 3GPP and CAMARA. The chapter’s technical analysis focuses on issues of data exposure and quality assurance, AI model lifecycle management and privacy-preserving learning methods, distributed computing orchestration, failure prediction strategies, and robust integrated sensing protocols. These elements are demonstrated to work together while balancing conflicting requirements—such as network latency, compute overhead, security, and user privacy—in complex, large-scale scenarios.

In addition to describing how these building blocks fit into the 6G end-to-end system blueprint, the chapter also highlights the importance of well-defined APIs for multi-stakeholder data sharing, the coordination of AI model placement and training, and system resiliency through distributed sensing techniques. The resulting architecture demonstrates that 6G networks can be designed to accommodate data-driven, AI-enabled functionalities as core services, rather than optional add-ons. The documented procedures, protocols, and interface specifications show how networks can progress beyond conventional communication tasks to address the diverse and advanced needs of 6G applications. Through this integrated approach, the objectives of WPO3.1 are met, affirming that 6G systems can sustain complex, AI-centric operations and beyond communications capabilities in a manner that is both robust and aligned with the emerging standards in this domain.

4 Flexible Topologies

4.1 Introduction

Flexible topologies aim to increase coverage, reliability and resource usage efficiency, while reducing power consumption and enabling new types of devices to be part of the cellular system. The three architectural enablers “network of networks”, “multi-connectivity” and “E2E context-awareness management” were analysed in [HEX224-D33], where initial solutions and results were also provided. Related to WPO3.3 of Table 1-1, certain solutions related to new access, flexible topologies, local communications, different types of multi-connectivity, node roles and node coordination are further developed and analysed in this deliverable compared to [HEX224-D33], by providing more details about the architecture and the associated procedures. Control and management solutions for programmable and context-aware transport are also further analysed in this deliverable. Each architectural enabler is mapped to the 6G E2E system blueprint in Figure 4-1, as explained in the introduction of each enabler’s section. As in [HEX224-D33], note that the mapping corresponds to the studies of each enabler in this deliverable.

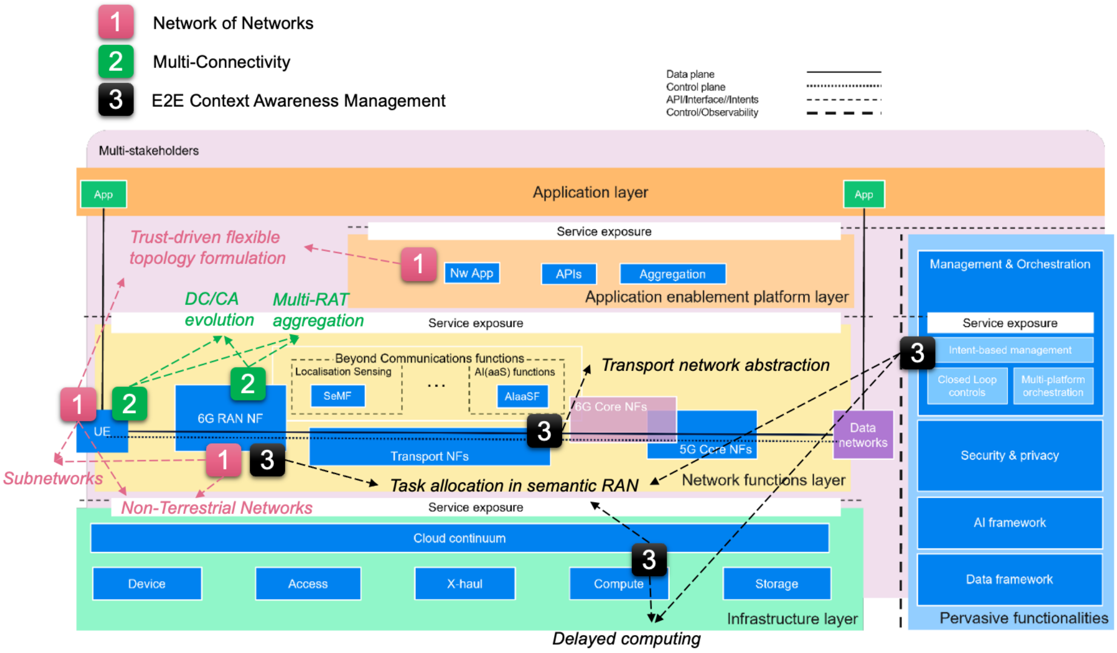


Figure 4-1 Mapping of the network of networks, multi-connectivity and E2E context awareness management enablers to the 6G E2E system blueprint of [HEX223-D23]

4.2 Multi-connectivity

4.2.1 Introduction

The multi-connectivity enabler includes studies that are related to the enablement of multiple paths between the network and the UE. In this deliverable, the evolution of CA and DC, the aggregation of different access networks and multi-server offloading are further analysed. The CA/DC evolution should combine the best features of CA and DC into a single solution to avoid having similar solutions. Details about a procedure that exists in DC but not in CA and how it could be implemented in CA are provided in Section 4.2.2. In the same section, a CA performance evaluation of various deployments is analysed. The enabler also focuses on WLAN – Cellular Aggregation (WCA), to enhance coverage while providing increased reliability. In Section 4.2.3, the protocol stack of the WCA feature is presented and analysed. Finally, a delay evaluation of a system, where a UE is connected to multiple network nodes associated with certain communication and computation resources is presented in Section 4.2.4. As shown in Figure 4-1, all topics of the multi-connectivity enabler in this deliverable affect the UE and the RAN.

4.2.2 CA/DC Evolution

4.2.2.1 Introduction

5G has two solutions for MC: DC and CA. In DC, a master gNB forwards the data to the secondary gNB via the Xn interface. DC can improve throughput and reliability, especially beneficial for networks with varying frequency bands both FR1 and FR2 frequencies. The use of low frequency node (FR1) can provide reliability while the high frequency (FR2) node can boost the user throughput. The UE is configured by both the secondary and the master node, but the actual message is sent by the master node, so there is also a complex control plane interworking exchange between master and secondary nodes. This means that the control of the UE on the network side is distributed (shared), including the UE capability coordination. A flow control over Xn can cause “stalling” of the packet delivery due varying latency of the connections (see [HEX224-D33] for more details). The DL and UL are always coupled which means that the UL on worse connection limits coverage compared to carrier aggregation. In addition, the UE shall split transmit power between the two connections which limits coverage even more.

To activate a secondary node, the network (and the UE) has to monitor adjacent cells’ quality. This may be a rather slow process compared to the usual download times, and therefore sessions may often be completed before SCG is activated. A possible 6G improvement for DC is to improve the flow control mechanism and include fast UE feedback to the master node. This allows the master node to react proactively to avoid “stalling” the UE buffer. Another aspect would be to decouple the UL and DL and (as in CA) allow one UL connection and two DL connections.

CA is used to aggregate bandwidth and increase throughout. Typically, CA is used intra-site due to high demands on latency between the cells (PCell and SCell). CA only employs one scheduler, controlling the cells in the CA so there is no need for a flow control and can thereby avoid stalling problems of DC. UL control scheduling can be either on both PCell and SCell or on only PCell which means that DL and UL are not coupled, and this gives better UL coverage compared to DC. The main disadvantage of CA is that it cannot handle inter-vendor operability. Possible 6G improvements for CA can be a faster method to include a secondary node, e.g., with a pre-configuration of adjacent cells (similar to conditional handover (CHO)). The robustness could be increased by e.g., PCell recovery via SCell or, using CHO and a set of inactive connections for fallback. A proposed way forward for 6G MC is depicted in Figure 4-2. There, the UE can connect via several non-co-located Radio Units (RU) and using LLS to the network, aggregating the resources from each RU using CA like 6G MC.

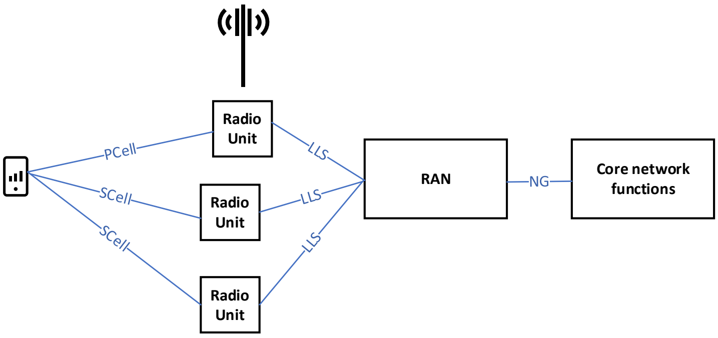


Figure 4-2 6G MC proposal

4.2.2.2 Carrier Aggregation evaluations

In [HEX224-D33], the impact on different bandwidth on the CA performance was investigated. It was concluded that to achieve gain from aggregation, the carriers must be rather equal in terms of capacity and coverage. For example, if one carrier has much higher bandwidth (and thus higher capacity) than the other carrier, the benefit to add the low bandwidth carrier is rather minimal. At the same time, if the packet is small, the master carrier has in many cases ended the session before a secondary carrier is added. This is a continuation of the evaluations in [HEX224-D33], but now focusing on even more deployments, see Table 4-1. Several different deployments are evaluated to investigate the benefits with aggregation of carriers. The first scenarios use only CA between low-band cells with are either co-located or non-co-located). The scheduler when CA is used can be either only within one site or system-wide scheduler, i.e. if CA is done between any cells in the system, the scheduling can take place in any cell with no delays (i.e. the backhaul is ideal between all CA cells). Furthermore, to investigate aggregation between different frequencies, a densified midband is overlayed on the low-band cells, see the three last rows in Table 4-1.

Table 4-1 CA evaluation deployment settings

Name	**CA**	**ISD Low-band/ Mid-band**	**Co-located / non-co-located and scheduler type**
Cell scheduler, co-located,	No	500m/NA	Default scenario: No CA, cell scheduler, co-located cells, only low-band.
Site-scheduler, non-co-located sites	Yes	500m/NA	CA, non co located sites, site scheduler, only low-band
Site-scheduler, co-located sites	Yes	500m/NA	CA for the site cells, and co-located per site
With midband	No	500m/288m	No CA, with mid-band densification
Site-scheduler, with midband	Yes	500m/288m	CA for the site cells, and co-located per site and with mid-band
System-wide-scheduler, with midband	Yes	500m/288m	CA, system wide scheduler and co-located per site and with mid-band.

Other important parameters used in the simulations are shown in Table 4-2. An ideal backhaul is used and there is no control signaling delays (e.g., when SCell is added). Note that the mid band uses 16 Tx/Rx antennas in the gNB, compared to 2 Tx/Rx for the low band. This evens out the coverage differences between the bands to some extent. The users select the mid-band carrier at cell selection if the RSRP is larger than -100 dBm, otherwise the low band is selected. After some initial measurements, the user also notices that there is another carrier exceeding an RSRP of -100 dBm. The network will after some delay then also add the low-band carrier.

Table 4-2 Parameter settings

Parameter	**Low band**	**Mid band**
Carrier frequency	800 MHz	3500 MHz
Bandwidth	10 MHz	100 MHz
Tx/Rx antennas in gNB	2	16
FDD or TDD	FDD	TDD
Traffic	100 MB FTP DL, 80% Indoor Users	

Figure 4-3 shows the relative average user throughput gain compared to default deployment (no CA) for downloading a packet of size 100 MB (called object bit rate) as a function of the user intensity (number of users arriving to the system and starts downloading an object per second).

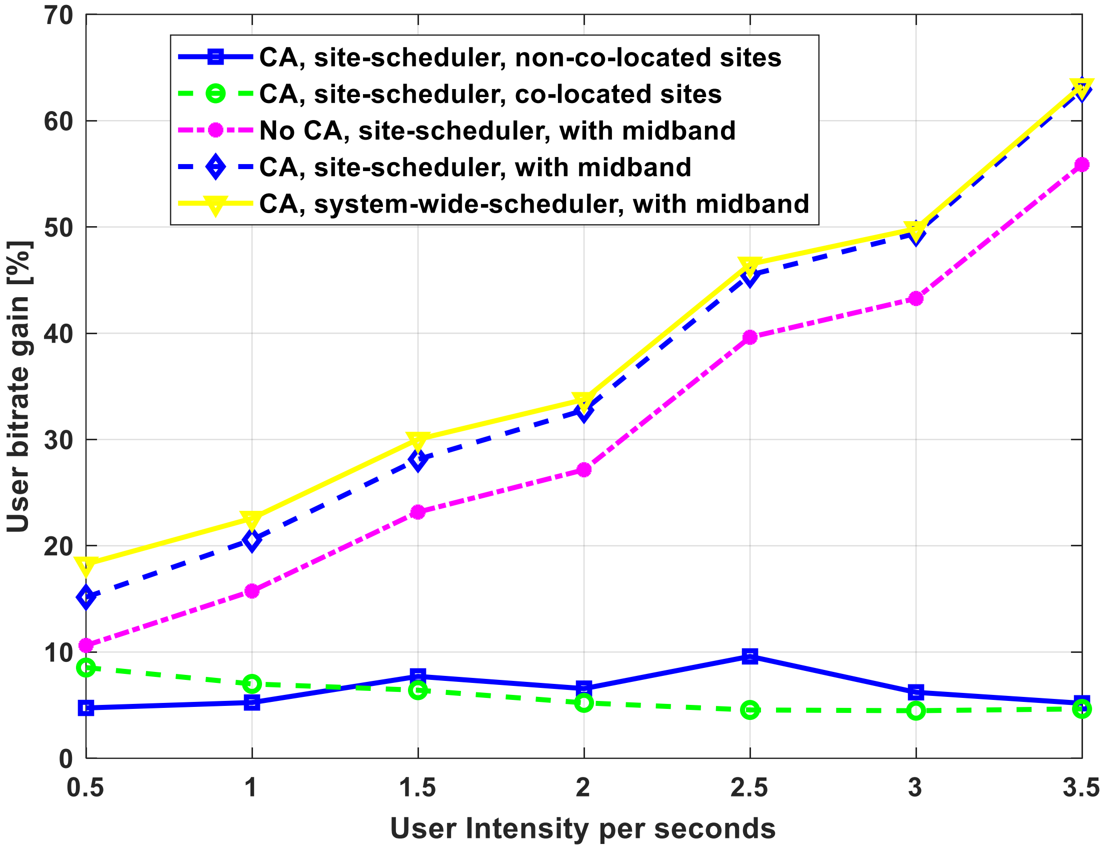


Figure 4-3 Average DL user throughput as a function of the user intensity for different deployments with and without CA

The results show that the CA deployment "CA, site-scheduler, non-co-located sites" (blue) and the "CA, site-scheduler, co-located sites" (green) have gains in similar range (5-10%). The magenta case shows the relative gain when midband is densified without any CA. As can be seen, the gain from densification is rather high, and increases with increased intensity (traffic load). There is also a further gain of around 5-10% by adding CA here on top of the densification, which is shown in the orange (per site scheduler) and yellow curves (system wide scheduler).

4.2.2.3 SCell-aided fast PCell recovery

UE mobility may lead to Radio Link Failure (RLF), especially at the cell edge. In single cell group connectivity (including CA case), when RLF occurs, the UE initiates RRC-Reestablishment procedure and tries to find a suitable cell to continue connectivity. This leads to increased service interruption time, as explained in Section A.5.1 . However, in the case of DC, upon MCG failure (PCell), there are mechanisms to recover the connectivity using the SCG (e.g., PSCell), without the need of initiating the re-establishment procedure. Hence, mechanisms for solving the same issue are required in CA, to reduce the service interruption time. It is proposed that upon RLF on PCell, the UE may use the SCell to recover the connectivity, when CA is enabled. The procedure to realize this is described in more detail in Figure 4-4. Initially, the UE is connected to a PCell (Step 1). When an RLF indication reaches the UE RRC, the UE does not initiate the RRC Re-establishment procedure, if an SCell is configured, as in legacy cellular systems. Instead, the UE RRC prepares failure information and sends this information to the network via the SCell (Step 2). The failure information may include SCell measurements as well as the measurements of the neighbouring cells. When the network receives the PCell failure information from the UE via the SCell (Step 3), it may prepare RRC reconfiguration to configure a new PCell for the UE via the SCell. Once the UE receives a new RRC reconfiguration or indication from the network (Step 4), the UE may initiate handover to a new PCell or configure SCell to become PCell (Step 5). Note that even though the SCell and the Target PCell appear as different nodes in Figure 4-4, they could also be the same (i.e., the SCell eventually becomes the Target PCell and then the PCell). Section A.5.1 includes a flowchart describing the logic at the UE for the SCell-aided fast PCell recovery procedure.

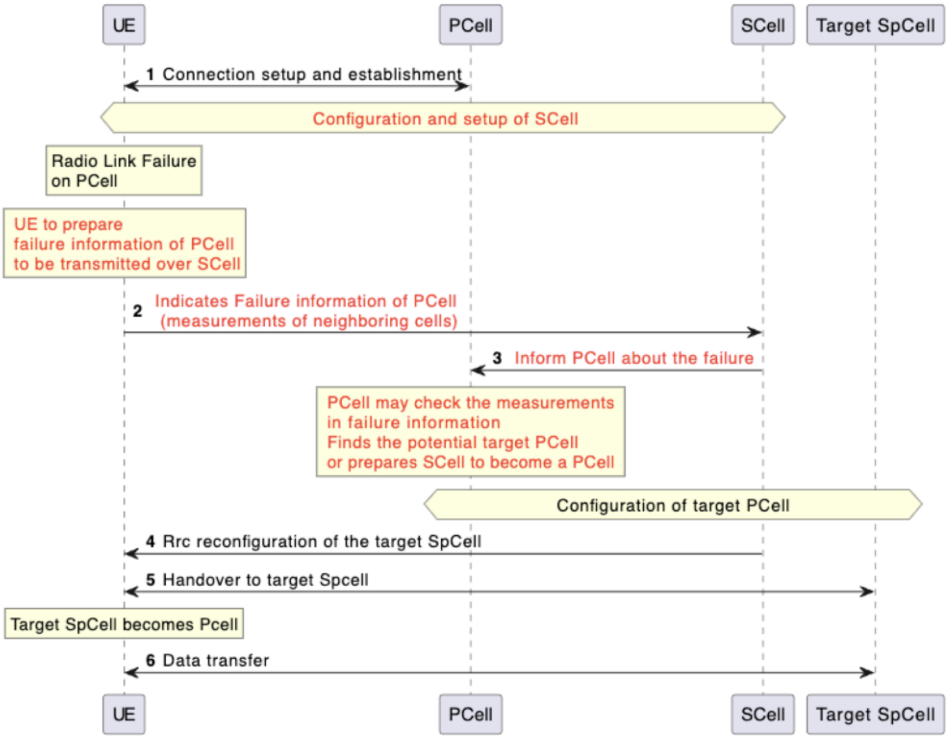

UE
PCell
SCell
Target SpCell
1 Connection setup and establishment
Configuration and setup of SCell
Radio Link Failure on PCell
UE to prepare failure information of PCell to be transmitted over SCell
2 Indicates Failure information of PCell (measurements of neighboring cells)
3 Inform PCell about the failure
PCell may check the measurements in failure information
Finds the potential target PCell or prepares SCell to become a PCell
Configuration of target PCell
4 Rrc reconfiguration of the target SpCell
5 Handover to target Spcell
Target SpCell becomes Pcell
6 Data transfer

Figure 4-4 SCell-aided fast PCell recovery

4.2.3 Aggregation of different access networks

One of the main motivations for an integrated WLAN and 6G solution is the indoor use case (e.g., in-home). In a scenario, where the UE is in cellular coverage of a BS as well as in WiFi coverage of a WLAN Terminal (WT), then the latter can act as a relay. For more details on this scenario please check [HEX224-D33]. Note that in this proposal the WT is connected wirelessly to the BS (i.e., via the Uu interface**)**. This connection may be on a different frequency range in comparison to the direct connection between the UE and the BS (i.e., also a Uu interface). The assumption is that this different frequency range has a better coverage or robustness. A different frequency range also increases diversity in this case.

The protocol stacks of the UE, the WT and the BS for WCA are shown in Figure 4-5. The WLAN Relay Adaptation Protocol (WRAP) is used when a packet is transmitted over the WT. The WRAP header includes all the information required by the WT and the end-point (i.e., BS in UL or UE in DL) in order to determine which UE-BS entities and parameters (e.g., cellular UE ID, radio bearer ID) the two-hop wireless transmission corresponds to. The WT shall read the content of the WRAP header to determine to which logical channel (i.e., WT-UE in DL or WT-BS in UL) it should map the packet. In terms of IDs, a UE cellular ID can be used by the WT for mapping the flows to the intended UEs, while a UE WLAN ID can be optionally used to simplify the operations of the WT. A security layer may be optionally added for the WT-BS link. If it is used, the added value is that it will cipher/integrity protect the WRAP header and add a sequence number on the BS-WT radio bearer. Still referring to Figure 4-5, the content and operation of the WRAP layer in each node corresponds to a WT (i.e., it does not store mapping tables). This means that prior to data transmission, the BS does not have to configure the UE ID – radio bearer mapping to the WT, the UE does not have to send its UE cellular ID to the WT and the UE may optionally send its UE WLAN ID to the BS. At the same time, the BS should configure the UE with the UE-BS radio bearer to WT-BS radio bearer mapping table, which is used in UL. Moreover, the BS should obtain the UE WLAN ID to use it in DL (i.e., either sent by the UE prior to data transmission or sent by the WT during the first UL transmission). Section A.5.2 describes a message sequence chart representing an example of WCA operation with a WT and a DL transmission and compares WCA with legacy solutions.

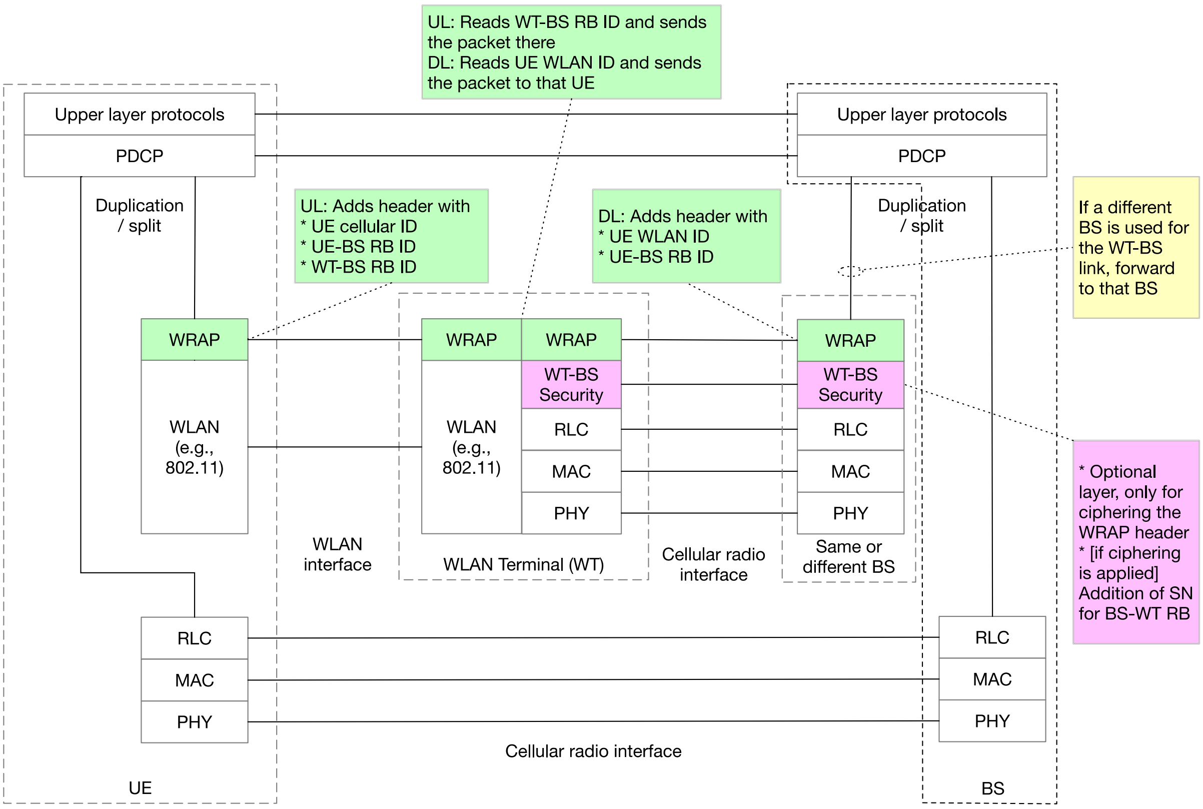


Figure 4-5 Protocol stack for WCA operation. The process and the headers of each layer correspond to the operation with a WT

4.2.4 Multi-server offloading scenario

One of the advancements of multi-connectivity examined in the previous deliverable in [HEX224-D33] concerns the application of the Rate-Splitting Multiple Access (RSMA) technique in multi-server systems to enable concurrent user task offloading to multiple servers. Specifically, heavy Machine Learning (ML) tasks, e.g., image processing, can benefit from multi-server edge offloading. Different video feeds generated from vehicular, healthcare, or security applications - to name a few – can be offloaded to different edge servers for processing, resulting in reduced total task complexity. By jointly optimizing task offloading ratios, uplink transmission powers, data rates, and server computing resources, the goal is to minimize the sum of the users' maximum delays experienced during task offloading and processing across the different edge servers. The formulated min-max-sum resource optimization problem is tackled using conventional optimization techniques and especially by applying the Karush-Kuhn-Tucker conditions. The problem is then decomposed into several independent subproblems, which iteratively provide optimal or near-optimal solutions for the radio and computing resource allocation as inputs to each other, until overall system convergence is achieved. In [HEX224-D33], the simulation topology used for the proposed analytic solution's performance evaluation was analytically provided, while some initial numerical results regarding the proposed solution's behaviour under increasing values of users and edge servers in the system were presented and discussed. The results practically validated the correctness of the analytic solution, concluding higher experienced delays by the users for both communication and remote computing at the edge servers as their number increases in the system, while the opposite behaviour is observed for increasing numbers of edge servers. Finally, the superiority of multi-server offloading against single-server offloading was demonstrated, providing higher flexibility for the system to meet users' requirements with lower end-to-end service delays.

In this deliverable, a more thorough performance evaluation of the proposed analytic solution is provided focusing on: (a) real execution time and performance when solved using solvers from well-known optimization toolboxes, and (b) performance when compared against alternative multiple access techniques, such as power-domain Non-Orthogonal Multiple Access (NOMA) and Orthogonal Frequency Division Multiple Access (OFDMA), in facilitating concurrent computation offloading from a single user to multiple edge servers. The overall work can be found in [DPT+2024].

In Figure 4-6, we characterize and evaluate the effectiveness of the proposed solution compared against the solution of the Sequential Quadratic Programming (SQP) iterative algorithm used in constrained non-linear optimization. Due to the high complexity of the original min-max-sum problem and the inability of the SQP algorithm to result in a stable and feasible solution as the number of users and edge servers increases, the comparison is carried out in a small system of 5 MHz overall bandwidth with M=[2,4] edge servers and N=[2, 5] users, the computation task size of which is set equal to 2000 Mega-CPU cycles.

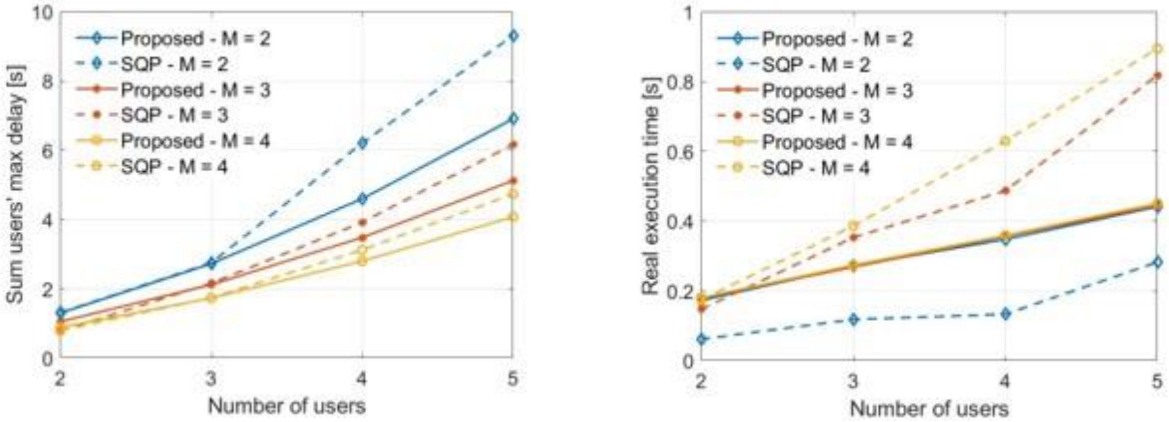


Figure 4-6 Comparative analysis between the proposed and SQP algorithms' solutions in terms of the (a) sum of users' maximum experienced delay (left) and (b) real execution time (right).

In particular, Figure 4-6(a) illustrates the sum of the users' maximum experienced delay among the different edge servers (vertical axis) as a function of the number of users (horizontal axis) and edge servers (different graph coloring), whereas Figure 4-6(b) examines the real execution time in seconds required for each one of the proposed and SQP algorithms to obtain a solution (vertical axis) under different user and edge server numbers. The results reveal that for particularly small systems with N=2 or N=3 users, the two algorithms present comparable performance regarding the minimum achievable value of the optimization objective. On the contrary, as the number of users increases, the proposed algorithm manages to conclude at lower sums of the users' maximum experienced delay among the different edge servers than the SQP, while this performance gap between the two algorithms diminishes as the number of edge servers increases. This is owed to the fact that for a small number of edge servers, the SQP algorithm can quickly pinpoint a local minimum of the problem by potentially sacrificing the minimization procedure's performance. The latter is further corroborated in Figure 4-6(b), where it is demonstrated that the real execution time required for the SQP algorithm to lead to a solution is increasing proportionally to both the number of users and edge servers in the system, being faster than the proposed algorithm for M=2 and slower for M=3 or M=4.

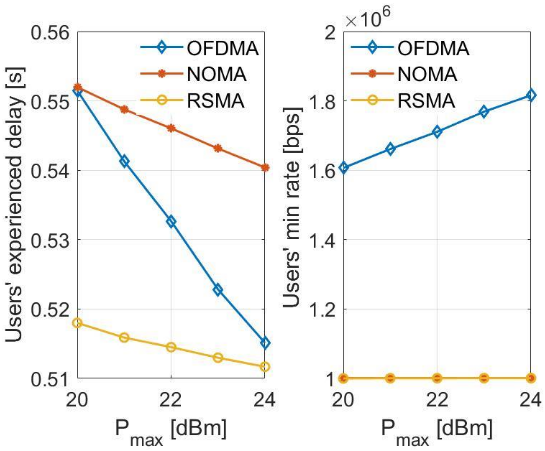


Figure 4-7 Comparative analysis between RSMA, NOMA, and OFDMA-based schemes in terms of the users' (a) overall experienced delay (left) and (b) min data rate (right), under different values of maximum power budget.

For a larger network topology, including N=40 users and M=3 servers, the performance gains of an RSMA-based multi-server edge system regarding the users' experienced delay are explored compared against an equivalent NOMA-based and OFDMA-based implementations. For a given system bandwidth, the OFDMA technique divides this bandwidth into N*M orthogonal frequency chunks, such that each transmission is allocated a dedicated frequency chunk, avoiding interference between users. This contrasts the NOMA and RSMA techniques that allow all transmissions to share the same frequency and time resources simultaneously. Notably, the fewer users and servers present in the system, the larger the bandwidth available per user under the OFDMA technique, resulting to a corresponding trade-off between system scalability and individual user performance. This trade-off is numerically depicted in Table 4-3. Analytically, in Table 4-3, the mean values of the users' overall experienced delays are summarized, considering different values of the minimum acceptable offloading data rate to the edge servers. As the minimum data rate demand increases, the users' experienced delay ameliorates both in the RSMA and NOMA-based edge systems, while the RSMA-based edge system yields at average 28 ms lower experienced delays to the users compared to the NOMA-based one in all examined cases. The OFDMA-based edge system operates only under very low data rate demands, i.e., 0.5 Mbps, for the given available bandwidth, number of users and servers, where its performance surpasses the RSMA and NOMA-based systems. The latter is justified by the fact that, in OFDMA, each separate transmission/ offloading is performed over a distinct frequency band that does not interfere with any other, resulting in a higher data rate and thus, lower users' experienced delay. Lastly, Figure 4-7 illustrates the mean values of the users' experienced delays under the RSMA, NOMA, and OFDMA techniques, but this time considering different values of the users' maximum power budget in the x axis. As the users' power budget increases, they can achieve higher offloading rates and thus, experience lower delays (left sub-figure) under all comparative multiple access techniques, while the RSMA-based solution provides constantly better performance.

Table 4-3 Comparative analysis between RSMA, NOMA, and OFDMA-based schemes in terms of the users' overall experienced delay[s] under different values of each link's minimum data rate.

	Link's minimum data rate [bps]						
Technique	**0.5 Mbps**	**0.75 Mbps**	**1 Mbps**	**1.25 Mbps**	**1.5 Mbps**	**1.75 Mbps**	**2 Mbps**
RSMA	0.6194	0.5570	0.5117	0.4777	0.4521	0.4333	0.4229
NOMA	0.6398	0.5827	0.5404	0.5082	0.4852	-	-
OFDMA	0.5143	-	-	-	-	-	-

4.2.5 Summary of 6G multi-connectivity enabler

The multi-connectivity enabler is summarized in Table 4-4.

Table 4-4 Summary of the MC enabler

Description	Multi-connectivity enables multiple frequency ranges by different physically separated nodes, the aggregation of different radio access technologies, carriers, and access networks.
Key take-aways	Measurable KPIs: Increased robustness, reliability, user and system throughput. Non-measurable KPIs: More efficient management of network resources. Only one architectural option (e.g., only CA) should be defined in the specifications. Proposed CA improvements inspired by the DC advantages and vice versa. New procedure proposed in this deliverable: PCell recovery via SCell solution. Coverage extension or increased reliability via WLAN-based UE relay for remote UEs. Carrier aggregation simulations show most gains when the aggregated connections have similar performance in terms of coverage and capacity. Scenarios where the PCC has much higher BW than the SCC do not lead to any significant gains in terms of capacity when CA is activated. A possible deployment for 6G MC is to use radio units connected to the same gNB using LLS, where the RUs are not co-located.

	The application of the RSMA technique for multi-server offloading enables optimized power and rate allocations, effectively meeting stricter QoS requirements for users compared to the conventional power-domain NOMA technique.
Requirements	More accurate timing synchronization for non-co-located cells using CA instead of DC. Faster addition of cells compared to 5G. Change of UL control signalling for better robustness. New device type needed: WLAN terminal with WCA support.
Standard relations & regulations	CA/DC: RRC protocol modifications and procedures for CA/DC evolution defined in 3GPP RAN2. Different access networks: RAN protocols for aggregation with other RANs defined in 3GPP RAN2

4.3 Network of networks

4.3.1 Introduction

The network of networks enabler includes studies that are related to trustworthy flexible topologies (e.g., subnetworks (SubNW)) or enable new access (e.g., NTN). In this deliverable, NTN and trustworthy flexible topologies are further analysed. Architectural options of NTN, including the enhancements of TN-NTN DC compared to [HEX224-D33] are analysed in Section 4.3.2. More details about the subnetwork architectures and the subnetwork Control Plane (snCP) compared to [HEX224-D33] and the proposal of the subnetwork-specific RRC configuration and the RRC configuration procedure in Management Node (MgtN)-SA mode are stated in Section 4.3.3. In the same section, the authentication, registration, resource allocation and mobility management procedures of a trustworthy, flexible and unstructured network are analysed. As shown in Figure 4-1, the NTN and subnetworks studies affect the UE and the RAN, while the trust-driven flexible topology formulation is related to the UE and the application enablement platform layer, as explained in Section 4.3.3.

4.3.2 Non-Terrestrial Networks

4.3.2.1 NTN architecture

There are several architecture options 6G NTN (see e.g., [6GN23-D31], [HEX224-D23]), however, two main architecture options for 6G NTN are foreseen (see Figure 4-8):

- RU (or remote radio head) on-board: The RU is on the satellite, the rest of the gNB functions are on the ground
- gNB on-board: The whole gNB is located on the satellite.

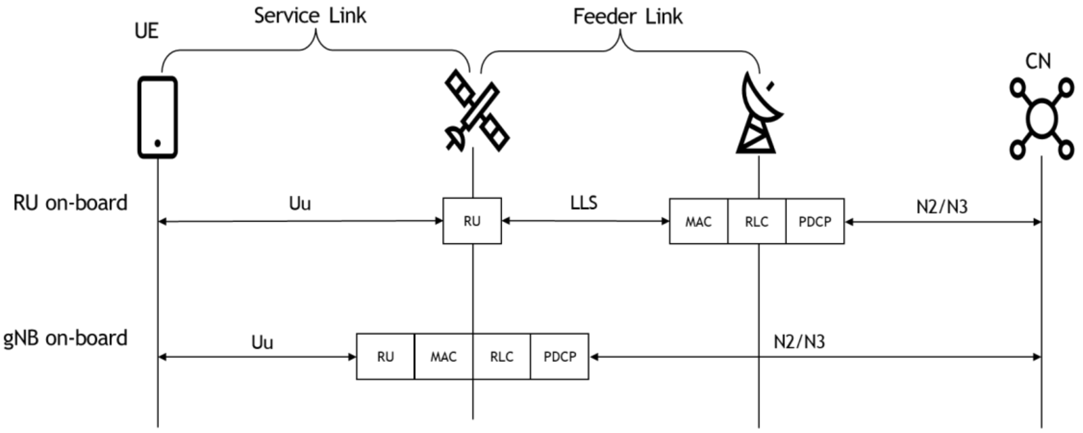


Figure 4-8 6G NTN main architecture options and how their protocol stack is applied for the different options.

For the RU option, the satellite can decode and construct the L1 (or part of L1 with some lower layer split-option) and therefore can apply decode and forward relaying. Using an RU onboard the satellite still requires the scheduling to be performed on the ground BS since MAC, RLC and PDCP is located in the ground station. The advantage here is the low HW and SW impact as well as low weight. The main problem for this option may be the needed capacity for transmitting the LLS split data (for the service link and the Inter-Satellite Link (ISL)), see Table 4-5.

Table 4-5 Summary of the different 6G NTN architecture options

	Dynamic function relationships	ISL / multi-hop	Capacity / performance	HW/SW impact on satellites	Standard impact
RU on board	RU dynamic association to be implemented in the LLS interface	Extension of LLS routing for multi-hop	The data LLS using L1 split option “7-x“ [38.801, Ch. 11] requires several times more capacity than the gNB onboard option, (depending on the actual LLS for NTN), this may be difficult to scale the LLS for ISL.	Minimal	No functional split impact
gNB on board	- Dynamic gNB association to AMF - Handling of dynamic gNB configurations	Multi-hop via satellite L3-L1	Control plane traffic and user plane traffic	High with full gNB	Possible impact on the N2 and N3 interfaces to support dynamic association.

The second option is to have the full gNB onboard. The main benefit here is that the functions that need close interaction are in one place, e.g. scheduling, beam management etc. The scalability for FL and ISL is relatively good here since only CP and UP traffic need to be transmitted (see Table 4-5). The disadvantages may be that there is no clear option for multi-hop ISL; however, the Integrated Access and Backhaul (IAB) could be reused if a 3GPP solution is desired. Also, another disadvantage is that the one is that the interfaces between RAN and CN are not designed with mobility in mind. In addition to this, there may be need to handle some sort of N2/N3 multi-hop solution for the full gNB option. Therefore, a more future proof way forward could be to handle the N2/N3 multi-hop via some NTN-specific L3/L2 solutions rather than via 3GPP-specific ones.

Regarding UEs, there are also different options. One option is to use devices with extra capabilities such as extra number of antennas. The other option is to use a regular device. In 2024, some companies will deploy satellites that can connect to ordinary 3GPP UEs without any extra components [EFH+23].

Another option discussed in [6GN24-D35] is a distributed architecture, with a functional split in which only the RU and the low PHY are placed in the service satellites, whereas all the rest of the DU, CU and if necessary, CN functionalities are located in the feeder satellites (see Figure 4-8).

4.3.2.2 TN-NTN Faster switching

Two options are foreseen for TN-NTN faster switching while using DC framework as baseline, depending on which node (TN or NTN) will be the Master Node (MN) and which one will be the Secondary Node (SN). The option where the TN node is the MN is termed as “Option 1” and is depicted in Figure 4-9. UE performs initial context setup with TN network and TN is configured as Master node and NTN as secondary node. NTN secondary node is selected either based on UE measurements or UE location. NTN circuitry is switched off. User data transfer takes place between UE and CN via TN network. When UE slips towards the end of TN coverage it provides an indication to TN. UE may internally activate NTN circuitry and TN node sends activation message to NTN node. Alternatively, TN sends a message to UE for activating NTN and UE perform timing synchronisation with NTN. Now, user data transfer takes place between UE and CN via NTN network. UE may perform TN measurements and as soon as it comes back to good TN coverage, it sends this indication again to NTN. NTN may either directly perform the switch if UE is time aligned with TN or UE performs time alignment before the switch.

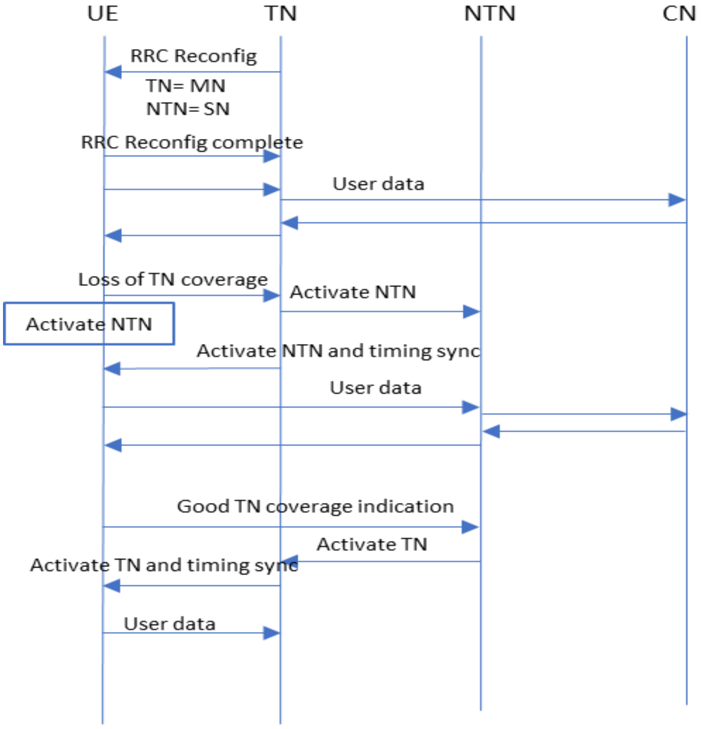


Figure 4-9 TN-NTN Faster switching with TN as MN and NTN as SN

The option where the NTN node is the MN is termed as “Option 2” and is depicted in Figure 4-10. In this option NTN acts as master node and TN as Secondary Node. UE connects to TN and performs user data transfer. UE indicates loss of TN coverage to NTN and NTN performs the switch immediately. UE NTN circuitry is always activated because NTN acts as a Master node and already synchronised and able to perform measurement. In this case, UE may keep TN circuitry also switched on because the interruption may be temporary. So, this option does not have the benefit of UE power saving as compared to Option 1.

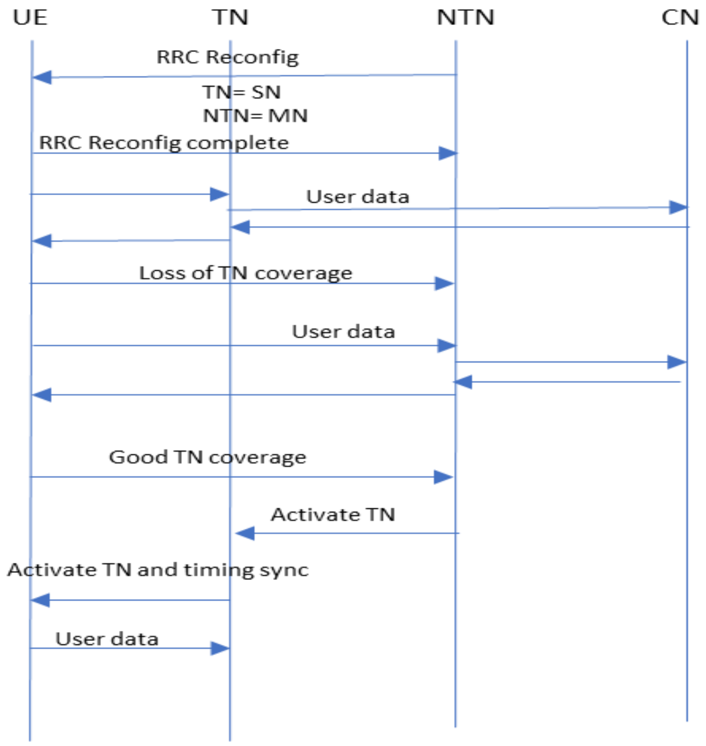


Figure 4-10 TN-NTN Faster switching with NTN as MN and TN as SN

We have used DC as a framework to allow pre-configuration for handover and faster switching. This is different when compared to the traditional DC, which is mainly used for throughput enhancements. This faster switching procedure could alternatively be called as L3 HO including the benefits of L2 Triggered Mobility (LTM) and Conditional HO + Dual Active Protocol Stack (DAPS). The edge of TN coverage could be based on fine tuning of existing measurement events or defining new events so that this message is not lost, a similar motivation as CHO. The other leg is active before the radio connection for main leg is broken, similar motivation as DAPS. “Activate TN” could be explicit or implicit action from the UE to switch on NTN circuitry when “Edge of TN coverage” event was triggered and either performed when TN indicates or coordinated internally within the UE between TN and NTN stack. Timing synchronisation will be performed with NTN prior to TN coverage is lost, incorporating the main benefit of LTM.

4.3.2.3 Inter-satellite link based on optical link

For regenerative payloads, ISLs between satellites should be able to handle large amount of traffic compared to bent-pipe architecture. Some of the functions from RAN architecture like CU-DU split interface or the interface towards edge compute etc. will contribute towards the traffic between different satellites and ISLs. ISLs could be based on RF or optical links.

One of the developments happening in optical domain is the use of optical switches. There is no need to convert optical signals to electrical signals and vice versa and the switching takes place using optical signals. ISLs offer the opportunity to deploy optical switches due to isolated network deployments up in the space and probably having no legacy network issues. This will require further study if RNL/TNL protocol stack can be optimised and whether there is a need for IP protocol in the protocol stack. Figure 4-11 below shows an example of F1 interface between CU-DU, but it could be any other interface between satellites.

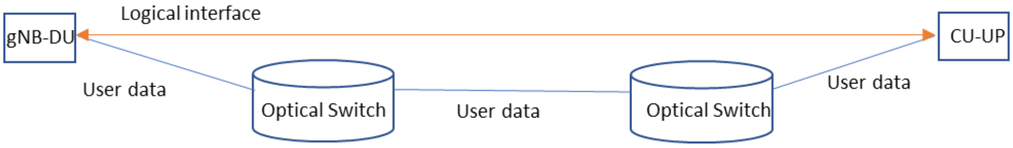


Figure 4-11 ISL interface using optical switch.

User data travels through optical switches without conversion to electrical signals. User plane protocol stack for transporting RLC PDUs may look like as shown in Figure 4-12:

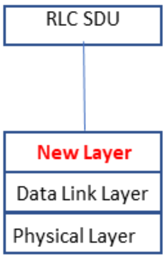


Figure 4-12 Example of user plane protocol stack for ISL with optical switches.

Physical layer and data link layer may be implemented in optical switch and a new layer replacing the IP layer may be needed for end-to-end routing purpose.

4.3.3 Trustworthy Flexible Topologies

4.3.3.1 Subnetworks

As described in [HEX224-D33], a UE may assume the new role of a MgtN and aid to form a SubNW voluntarily by the participating UEs, based on mutual trust. The MgtN is the SubNW's primary node and it can communicate with the Base Station (BS) and other UEs. In addition, the notion of the lightweight SubNW snCP was also introduced therein. As explained in [HEX224-D33], the snCP includes the procedures that run within the SubNW and the offloaded UE CP information, therefore it is transparent to the NW. In this deliverable, the foreseen subnetwork architectures are presented, and more details are provided about the SubNW CP. Moreover, the issues related to the RRC configuration of the UEs in a subnetwork are discussed, followed by the introduction of the so-called "lean RRC-reconfiguration".

The MgtN Standalone (MgtN-SA) architecture is presented in Figure 4-13(a). In this architecture, the UE is only connected to the MgtN and the MgtN has access to the network via the BS. This means that all communication between the UE and the network will go through the trusted MgtN. The network's coverage is increased with this architecture in the scenarios where the UE would have been out-of-coverage if it was not connected to the MgtN.

The BS-MgtN DC architecture is illustrated in Figure 4-13(b). In this architecture, the UE has two connections: a main connection with the BS and a secondary connection with the MgtN. The main connection with the BS is primarily used for procedures that require connection to the overlay network, such as mobility between cells and initial configuration. The UE connection is lost only if this main connection with the BS is lost. The secondary connection with the MgtN is used for configuration control within the subnetwork. It can also be used for relaying configuration control from the BS if needed, for increasing the reliability of transmitting data to the BS and to ensure the maintenance of the main connection to the BS. Both connections may be used for data transmission.

The BS-SA with MgtN support architecture is illustrated in Figure 4-13(c). The UE in this architecture also has two connections: a main connection with the BS and an additional local connection with the MgtN. Nevertheless, this architecture can be considered as BS-SA from the perspective of the BS, since all UP and CP communication will take place on the BS-UE link. The MgtN-UE link is used so that the MgtN can support the UE with different CP functionalities.

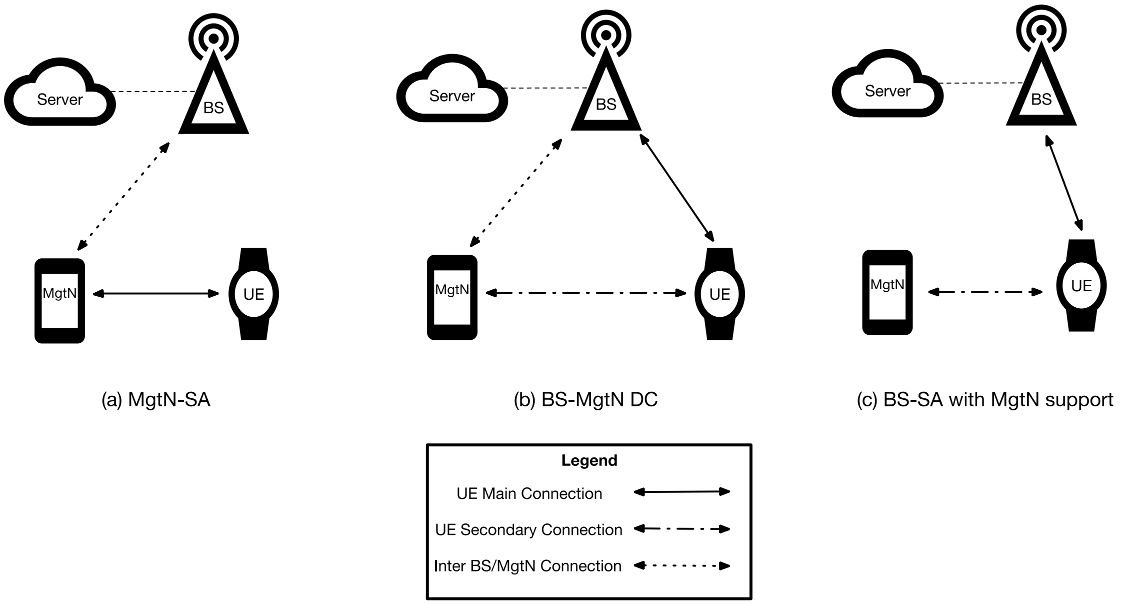


Figure 4-13 Subnetwork architectures

In a SubNW, the UEs may offload their CP or part of it towards the MgtN. To that end, the SubNW may use a new lightweight snCP between the UE and the MgtN [HEX224-D33], which is transparent to the NW. The snCP would include the configuration and procedures that take place within the subnetwork, as well as the offloaded UE CP information. An architectural option is when the UE offloads its CP to another UE without the BS's awareness. This enables lower capability UEs and/or power saving by letting more capable UEs in the SubNW do the heavy lifting, as it will be described in the following for the RRC configuration procedure. The fact that the target UE does not support a full cellular CP and supports only snCP is abstracted from the BS. Another architectural option includes UEs in the SubNW supporting both the cellular CP for procedures related to the BS, as well as an snCP only for SubNW-specific procedures. All foreseen architectural options based on the snCP all described in Section A.6.1

Currently, a BS may configure many UEs with many configurations, such as System Information Blocks (SIB) for cell-common configurations and RRC reconfiguration for both cell-common and UE-specific configurations. The cell-common configuration is the same for all UEs of a specific cell. It may be encountered that even UE-specific configurations are very similar for similar devices, which may lead to a waste of radio resources for configuring the same parameters and procedures on multiple devices. The 3GPP RRC specification [38.331] is defined in a very general way that handles various types of devices and hence most of the features therein are defined as optional. Having a large number of variations in the specification results in a significant complexity at the UE side, which in turn limits the creation of more processing-limited and power-efficient devices. In a non-transparent SubNW architecture [HEX224-D33], the BS is aware of the presence of the MgtN and the SubNW. In that case, the BS would not only need to configure UEs with the macro cell configuration, but also with subnetwork configurations. A continuous synchronization between the BS and the MgtN before configuring the UE would further increase the overall configuration procedure delay. At the same time, if the BS manages each UE individually, this would be resource inefficient and would increase the overall system complexity with a growing number of devices. Therefore, there is a need to have a more resource-efficient SubNW configuration and to enable lean RRC reconfiguration via the snCP.

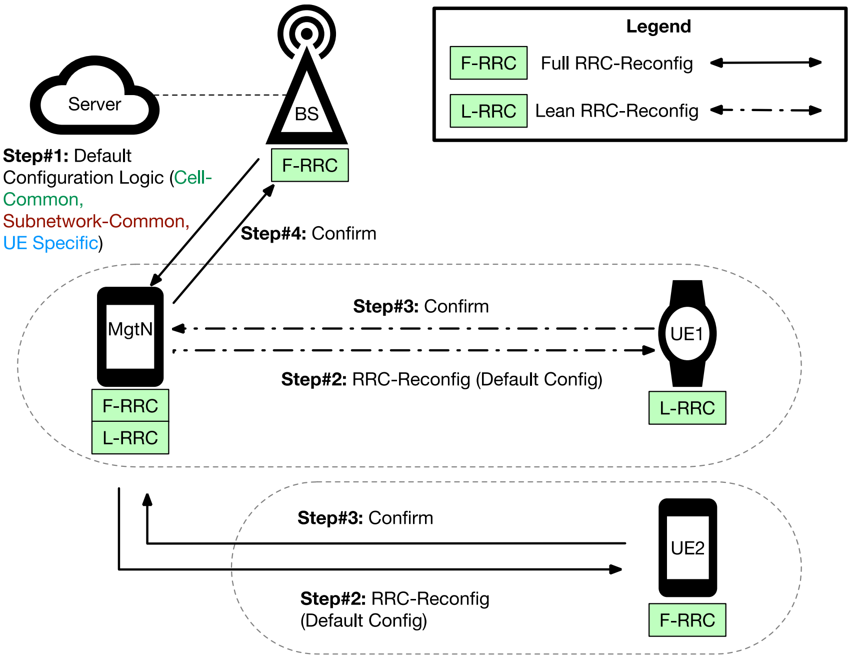

Legend
F-RRC
Full RRC-Reconfig
L-RRC
Lean RRC-Reconfig
Server
BS
F-RRC
Step#1: Default Configuration Logic (Cell-Common, Subnetwork-Common, UE Specific)
Step#4: Confirm
MgtN
F-RRC
L-RRC
Step#3: Confirm
Step#2: RRC-Reconfig (Default Config)
UE1
L-RRC
Step#3: Confirm
UE2
Step#2: RRC-Reconfig (Default Config)
F-RRC

Figure 4-14 RRC configuration with MgtN-SA

The concept is that RRC configuration-related procedures (e.g., decode RRC-Reconfiguration, validate final config) shall be delegated from the UE to the MgtN. When needed, the MgtN shall convert the final configuration from full RRC-Reconfiguration (F-RRC) format into lean RRC-Reconfiguration (L-RRC) format and provide it to the UEs via the snCP. For L-RRC the information shared is assumed to be specified and agreed between the MgtN and the NW (i.e., known by both the NW and the MgtN), however the exact messaging exchanged between the MgtN and the UEs within the SubNW is not specified and kept up to implementation. It could also be envisioned that L-RRC is used by the NW to configure directly the UE in some scenario (e.g., limited or down-sized configuration). This is depicted in Figure 4-14 and is described as follows. In Step 1, the BS instructs the MgtN with the configuration structure of the collocated UEs. The configuration shall be given in the following layers/groups: Cell Common, UE Specific and the newly proposed Subnetwork Common. The Subnetwork Common includes the configuration that would be common for all UEs that are part of the SubNW. In Step 2, the MgtN decides which default configuration shall be used, based on the logic configured from the BS. Still referring to Figure 4-14, at this step and only for UE1, which only supports L-RRC, the MgtN decodes the RRC-Reconfiguration of the selected configuration, merges the new and the old configurations, validates the final configuration and send the lean RRC configuration, via the snCP, to UE1 in an L-RRC format. Afterwards, UE1 apply the received configuration to the lower layers and send a confirmation to the MgtN (Step 3) which then sends a confirmation to the BS (Step 4).

One of the benefits of the MgtN-aided RRC configuration concept is the reduction of the UE’s configuration time via the introduction of the proposed common SubNW configuration and the self-managed SubNW configuration. With the former, there is no need for the NW to configure the SubNW common configuration via a dedicated configuration for each UE, while the latter results in relaxing the synchronization requirement between the BS and the MgtN. This concept also enables low-capability devices to use L-RRC to reduce their complexity. The F-RRC is considered decoupled from the L-RRC, since the NW will continue providing the configuration using the F-RRC and the NW does not need to know the details of the L-RRC. At the same time, the low-capability device does not need to know the details of the F-RRC, because the MgtN shall convert the configuration to the L-RRC. High-capability devices, which support both F-RRC and L-RRC, may use the L-RRC to save power. For example, after establishing a connection with the MgtN, a high-capability device may request the MgtN support for using L-RRC instead of F-RRC for power saving purposes.

More details regarding MgtN-aided RRC configuration are stated in Annex A.6.2 .

4.3.3.2 UAV-assisted trustworthy flexible topologies

The incorporation of UAVs into flexible network topologies marks a significant advancement in the realization of trustworthy 6G networks. UAVs enhance network adaptability, extend coverage, and improve reliability, particularly in dynamic and challenging environments where traditional Mobile Network Operator (MNO) infrastructures may be insufficient.

Central to the framework is the dynamic deployment of UAVs as mobile access points (AP), which complement terrestrial nodes by enabling rapid repositioning in response to fluctuating network demands and environmental constraints. The orchestration of UAV movements is managed by an Ad-hoc Network Controller (ANC), which continuously monitors network performance metrics and adjusts UAV positions to optimize coverage and resource allocation. The Trust Manager and Trust Evaluation Function (TEF) are integral components that ensure the reliability and security of the network. By extending the trustworthiness metric T to include UAV-specific parameters such as flight stability, energy reserves, and communication link integrity, the framework ensures that only UAVs meeting stringent trust criteria are incorporated into the network topology. The trustworthiness metric is defined as:

$$\mathrm{T_j} = \sum_{\mathrm{k=1}}^{\mathrm{n}} \mathrm{w_k} \left(\frac{\mathrm{f_{j,k}} - \mathrm{f_k^{min}}}{\mathrm{f_k^{max}} - \mathrm{f_k^{min}}} \right) \tag{4-1}$$

where j indexes the UAVs, k indexes the parameters being evaluated, w_k represents the weight of the k-th parameter, and $f_{j,k}$ encompasses both functional parameters (e.g., latency, throughput) and UAV-specific non-functional parameters (e.g., energy efficiency, operational cost) and $\mathrm{T_j}$ is the trustworthiness index of the j-th UAV. This comprehensive metric facilitates informed decision-making within the Flexible Mesh Selection Function (FMSF), which dynamically configures the network topology by selecting UAVs and terrestrial nodes that maximize overall trustworthiness while minimizing deployment and operational costs.

AI/ML-driven resource optimization is a foundation of the framework, enabling real-time adjustments to UAV positions and resource allocations based on predictive analytics and historical data patterns. Machine learning algorithms predict traffic load variations and potential coverage gaps, allowing the ANC to preemptively reposition UAVs to maintain optimal network performance. The optimization objective is formalized as:

$$\text{Maximize} \sum_{\mathrm{j=1}}^{\mathrm{J}} \mathrm{T_j Y_j} - \lambda \left(\sum_{\mathrm{j=1}}^{\mathrm{J}} \mathrm{K_j Y_j} + \sum_{\mathrm{j=1}}^{\mathrm{J}} \sum_{\mathrm{i=1}}^{\mathrm{I}} \mathrm{X_{ji} C_{ji}} \right) \tag{4-2}$$

Subject to the constraints:

$$\sum_{\mathrm{i=1}}^{\mathrm{I}} \mathrm{X_{ji} L_i} < \mathrm{Cap_j Y_j}\,, \forall \mathrm{j} \tag{4-3}$$

$$\sum_{\mathrm{i=1}}^{\mathrm{I}} \mathrm{X_{ji} E_i} < \mathrm{CapE_j Y_j}\,, \forall \mathrm{j} \tag{4-4}$$

$$\mathrm{X_{ji}} < \mathrm{Y_j},, \forall \mathrm{i,j} \tag{4-5}$$

$$\mathrm{Y_j} \in \{0,1\}, \mathrm{X_{ji}} \in \{0,1\} \tag{4-6}$$

where, λ serves as a weighting factor balancing trustworthiness and cost, $\mathrm{Y_j}$, is a binary variable indicating the selection of AP j, $\mathrm{X_{ji}}$ is a binary variable for the active link between traffic source (TS) i and AP j, $\mathrm{C_{ji}}$ represents the connection cost between TS i and AP j, $\mathrm{K_j}$ denotes the deployment cost, and $\mathrm{L_i}$ and $\mathrm{E_i}$ represent the traffic load and energy requirement of TS i, respectively. $\mathrm{Cap_j}$ and $\mathrm{CapE_j}$ are the throughput and energy capacities of AP j. This optimization ensures that selected UAVs enhance trustworthiness–a comprehensive metric encompassing both QoS and security aspects–while adhering to capacity and energy constraints, achieving a balanced and cost-effective network deployment. Following this procedure, in practical applications such as disaster recovery or large-scale outdoor events, UAV-assisted topologies exhibit exceptional flexibility and resilience. For example, in disaster response operations, UAVs are dynamically deployed to establish temporary communication links in areas where terrestrial infrastructure has been compromised. The trustworthiness metric guarantees that UAVs deployed in such critical situations achieve an optimal balance between reliability and performance, facilitating uninterrupted communication and coordination among emergency responders.

Following the selection procedure, advanced node discovery protocols and efficient algorithms like Minimum Spanning Tree (MST) are employed to optimize UAV placement and interconnectivity. Utilizing an isometric grid approach, the node discovery process efficiently approximates optimal UAV positions, reducing computational overhead and enhancing scalability. The subsequent MST formation, as seen in Figure 4-15, minimizes connection costs and ensures robust network connectivity, even in highly dynamic environments. Energy efficiency is addressed through AI/ML algorithms that prioritize energy conservation by optimizing UAV flight paths and minimizing redundant movements. Additionally, the framework incorporates energy harvesting techniques, enabling UAVs to autonomously return to designated charging stations when battery levels fall below predefined thresholds. This extends operational autonomy and ensures sustained network performance.

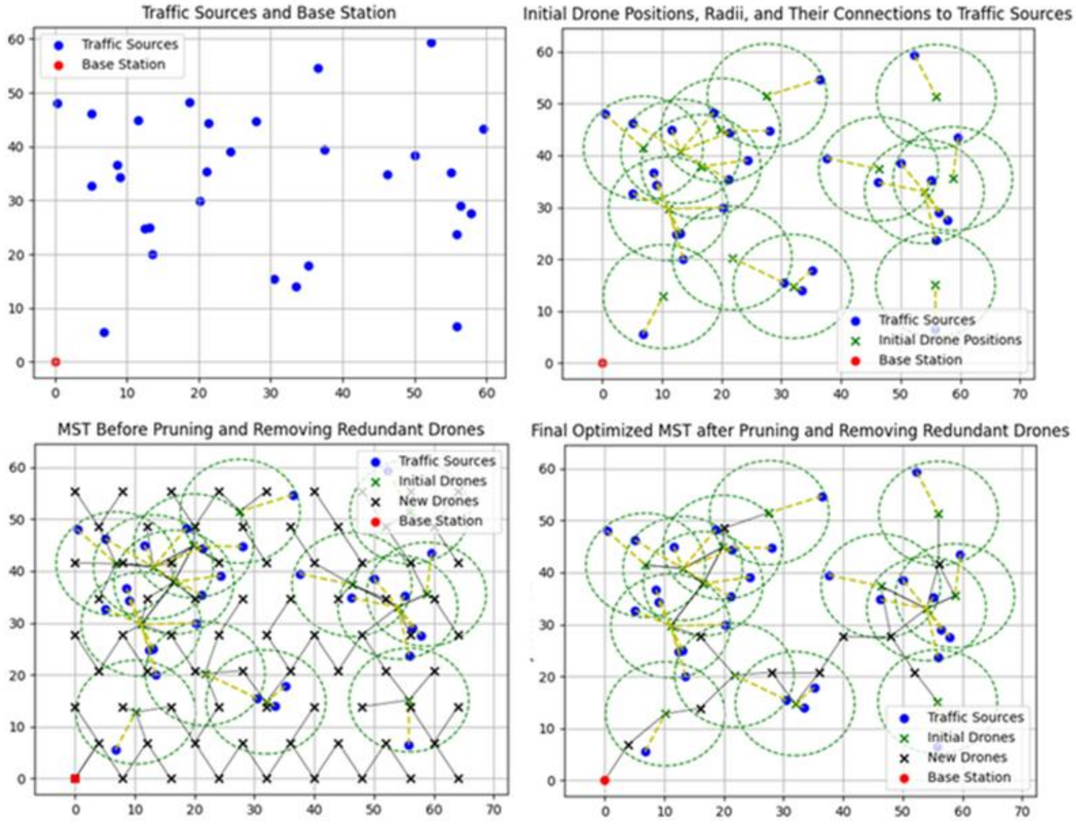


Figure 4-15 Flexible Network Topology Using Drone Positioning

The operationalization of UAV-assisted trustworthy flexible topologies relies on seamless coordination among network functions to ensure robust authentication, resource allocation, and mobility management. As illustrated in Figure 4-16, the Flexible Topology Network (FTN) initiates its workflow through authentication and registration processes mediated by the NEF and AMF. Upon successful registration, the FTN requests resources via the NEF, which forwards the request to the SMF and the PCF to enforce QoS policies and optimize resource distribution. During this phase, the AI/ML-driven Resource Optimization Component dynamically refines allocations based on real-time network conditions, ensuring alignment with the trustworthiness metrics and cost-efficiency objectives defined earlier. Mobility management is facilitated through continuous node discovery and adaptive data routing, with the AMF and User Plane Function (UPF) collaboratively maintaining uninterrupted connectivity as UAVs dynamically reposition. Throughout this process, metrics such as latency, throughput, and energy consumption are monitored and leveraged for real-time optimization, enhancing network performance and reliability. This closed-loop integration of trustworthiness-aware policies, AI/ML optimization, and mobility protocols ensures that the network adapts seamlessly to environmental or demand fluctuations while preserving the reliability and resilience which is central to UAV-assisted 6G deployments.

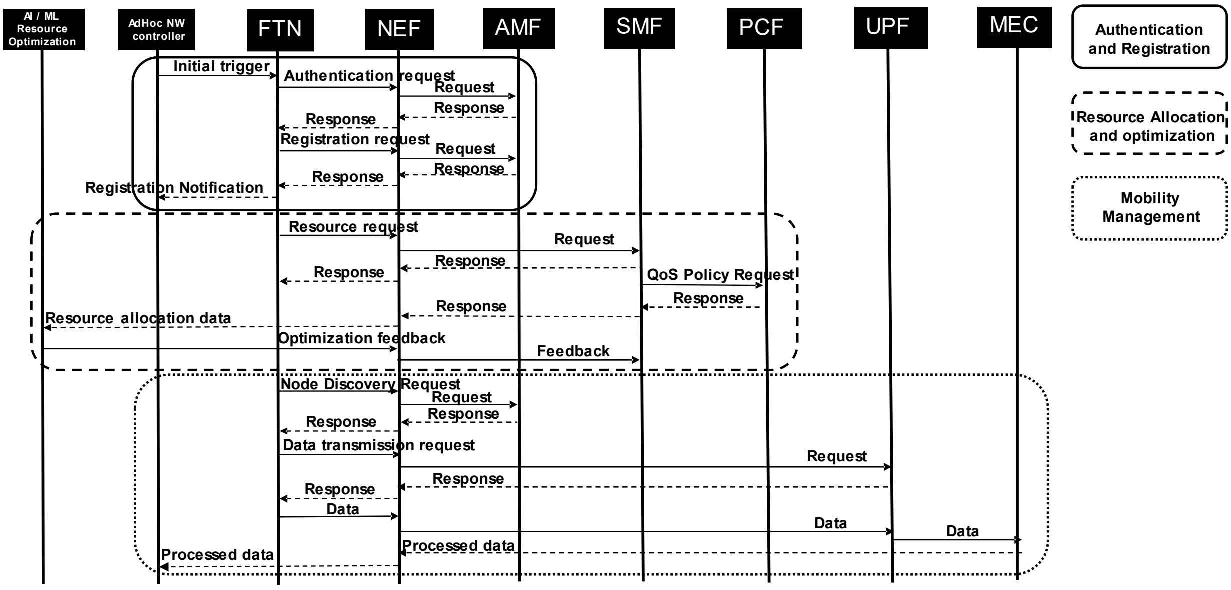


Figure 4-16 Message sequence chart for authentication, registration, resource allocation and mobility management of a trustworthy, flexible and unstructured network

4.3.4 Summary

The network of networks enabler is summarized in Table 4-6.

Table 4-6 Summary of the Network of Networks enabler

Description	Design of terrestrial subnetworks and NTN to create a seamless and ubiquitous communication system.
Key take-aways	Measurable KPIs: Increased coverage, reduced interruption time and time in outage, increased availability and reliability, reduced complexity. Non-measurable KPIs: service continuity. Multiple architectural options for NTN and subnetworks are needed to achieve the coverage requirements. Novel procedures for NTN and flexible topologies to reduce UE complexity and enable seamless mobility. New index for quantifying coverage inequality, which can be used for network planning and by regulators. More details are provided in Section B.1. Operational Resilience: Reliable network deployment in scenarios like disaster recovery and outdoor events through UAV-assisted flexible topologies. Seamless Network Integration: Interaction with functions such as NEF, AMF, and PCF for authentication, resource allocation, and mobility management, ensuring service continuity. The most likely options for 6G NTN architecture are “RU on board” or “gNB on board”.

Requirements	New RAN equipment (e.g., satellites, drones) and types of UEs (e.g., MgtN) are required for the proposed systems to work. NTN architectures should be specified. A TN-NTN dual connectivity procedure should be introduced to switch faster to NTN in the cases where there is no good TN coverage. Inter Satellite link protocol stack should be introduced. Complimenting the transparent and non-transparent architectures, the MgtN-SA, BS-MgtN DC, BS-SA with MgtN support architectures should also be specified. A new lightweight subnetwork CP between the MgtN and the UEs in a subnetwork can be introduced. Procedures such as "RRC configuration with the aid of the MgtN", which take advantage of the snCP, should be specified. Subnetwork-common RRC configuration should be specified so that all UEs in a subnetwork are configured based on it. Integration of advanced UAV control and management systems with existing network infrastructure to facilitate dynamic deployment and real-time optimization of UAV-assisted topologies. Implementation of robust trust management protocols and AI/ML-driven resource optimization algorithms to ensure the reliability, security, and energy efficiency of UAV operations within the flexible network framework.
Standard relations & regulations	NTNs should improve coverage. However, the notion of coverage needs to be revisited for 6G with the new diverse services and requirements and other related KPIs. Regulators will have to develop new measures to follow up and regulate national coverage. Various new architectures imply new needs to measure coverage. On the topic of TN-NTN DC, the measurements reported by the UE or evaluated by the network should provide sufficient information such that TN to NTN switchover is done in a timely manner. These procedures would impact the RAN and could be defined in 3GPP RAN2. Inter satellite link protocol stack could be defined in 3GPP RAN3. The architecture of the subnetworks, the new UE roles and their responsibilities in a subnetwork could be defined in 3GPP RAN2.

4.4 Context-aware management

4.4.1 Introduction

The context-aware management enabler includes studies that enable network and compute components to dynamically adapt to contextual changes, effectively meeting the services' and users' QoS and QoE requirements. Therefore, the studies' focus ranges from transport network abstraction and programmability to context-aware computational offloading decision-making. Section 4.4.2 provides specific examples of transport network abstraction toward context-aware resource orchestration, especially accounting for the scalability and resiliency aspects of the proposal in end-to-end network scenarios, which were not covered in the previous deliverable [HEX224-D33]. Section 4.4.3 examines an approach for intelligent switching and decision-making between delayed and approximate computing options, enabling the computing infrastructure to adapt dynamically to support various delay-sensitive or accuracy-sensitive tasks. Numerical evaluation results are included in this deliverable, validating the effectiveness of the approach. Furthermore, more details about context-aware switching using P4 programmability are provided in Section 4.4.4 along with the description of the experimental setup used to assess the performance of P4 programmable switches. Last, in Section 4.4.5, the analytic problem formulation and solution to the joint robotic function configuration and offloading policy optimization are presented for robot applications, following the high-level problem description in the previous deliverable [HEX224-D33]. As shown in Figure 4-1, the studies under context-aware management affect the UE, RAN, transport network functions, and the compute infrastructure, particularly for computation offloading services, ensuring seamless adaptation across the entire network ecosystem.

4.4.2 Transport network abstraction

The transport network plays a crucial role in connecting different parts of a mobile 3GPP network. These connections include those between the RAN and the CN, commonly known as mobile backhaul. Additionally, they encompass connections within the RAN itself, such as those between a RU) and a Distributed Unit (DU), referred to as fronthaul or lower layer split.

To enforce the context-aware transport [HEX224-D33], a resource orchestrator creates an abstracted view of the transport resources and triggers the SDN transport controller for resource handling to satisfy the QoS associated to a slice. It also performs E2E admission control to ensure the expected QoS for active and incoming services.

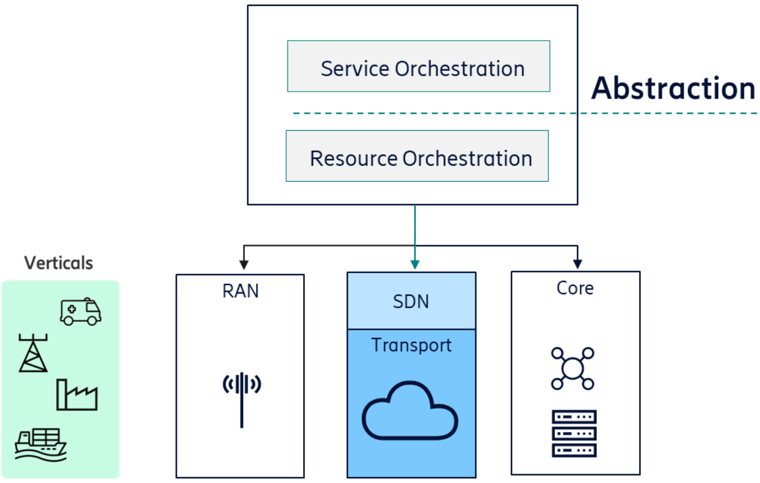


Figure 4-17 Network orchestration and abstraction

Figure 4-17 represent a high-level sketch of the network orchestration and abstraction scenario. Here a Service Orchestration functionality manages the delivery and lifecycle of network services, ensuring that they are provisioned, configured, and maintained in an efficient manner. A Resource Orchestration function handles the allocation and management of network resources, ensuring that they are optimally utilized to meet service requirements. By logically separating the service from the infrastructure technologies, the abstraction technique makes independent services from technologies, allowing these two elements to independently evolve. Additionally, the abstraction enables a clear separation of responsibility and roles among the infrastructure provider and the service provider. In the middle, the abstraction layer serves as an interface between the physical network components and the higher-level orchestration functions, providing a unified and simplified view of the network infrastructure. The service orchestrator places all network functions on the abstract view to guarantee the QoS of the considered slice.

Section A.7 reports details on the method for determining the abstraction of the transport network from the physical network using an exemplary meshed topology. By abstracting various network functions and their interconnections, operators can concentrate on essential aspects including redundancy for failure recovery and resilience without being overwhelmed by details of the infrastructure. Section A.7.2 focuses on demarcation points between radio and transport. These demarcation points help in simplifying the mapping of transport paths, whether for user plane or control plane traffic, by serving as “logical markers” for routing and recovery strategies. The appendix reports the concepts of focusing on redundancy and recovery at intra-site and inter-site levels with some exemplary scenarios of failures and recovery.

4.4.3 Delayed vs approximate computing

Under the prism of context-aware management, the previous deliverable [HEX224-D33] examined the particular scenario of computation offloading, by introducing a framework to prioritize delay-sensitive over delay-tolerant tasks for offloading to the edge or cloud. Building on this, the current deliverable goes a step further by exploring the tradeoff between accuracy and delay in approximate and delayed computing. On the one hand, under *delayed computing*, the entire task is transmitted to the edge server, which subsequently forwards it to the cloud for exact computation. This process incurs additional latency due to the transmission time required for transferring the task to the cloud. On the other hand, *approximate computing* complements delayed computing to addresses delay-sensitive tasks by processing a compressed sample of data at the edge instead of the entire dataset for faster execution, though it comes with the cost of reduced computation accuracy. Given that each user n has K_n computation tasks to be executed, each user selects one of the two computing options, as illustrated in Figure 4-18. Let x_n denote the number of tasks user n transmits for delayed computing, while the remaining $K_n - x_n$ tasks are sent for approximate computing. Each user's task has a mean data size of d_n bits, and s_n denotes the percentage of data d_n of each task that the user n decides to transmit to the edge server for approximate computing, representing the proportion of data to be compressed before transmission.

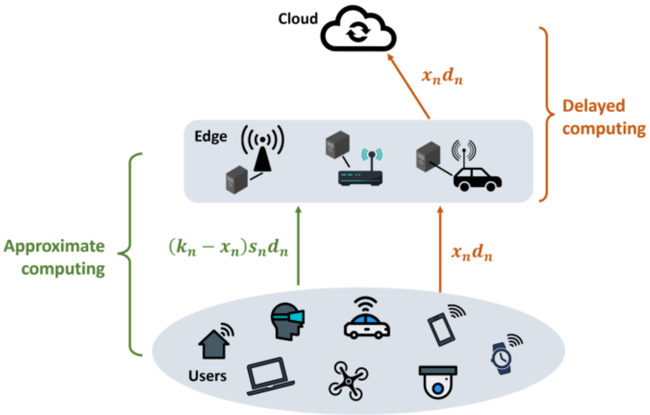


Figure 4-18 High-level overview of the considered computation offloading options, i.e., delayed and approximate computing for context-aware management.

Each user aims to optimize their task execution to meet their minimum accuracy and delay requirements. Thus, the goal of the users is to determine an optimal task allocation strategy x_n across the two execution options and the proportion of tasks s_n assigned for approximate computing and offloaded to the edge server after compression. By tuning these parameters, the users can balance the minimization of their experienced delay and the maximization of their processing accuracy. This joint problem is addressed as a non-cooperative game in satisfaction form, allowing the users to autonomously select strategies that satisfy their delay and accuracy requirements while achieving a Satisfaction Equilibrium (SE) for efficient task allocation. The SE point for all users is derived by employing a Reinforcement Learning (RL)-based algorithm, as detailed in [CDP24].

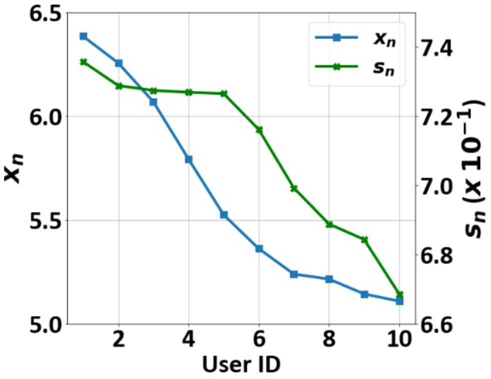


Figure 4-19 Percentage of offloaded data for delayed computing (left vertical axis) and compression rate for approximate computing (right vertical axis) for each user in the horizontal axis.

In the following, numerical results are provided from the preliminary evaluation of the proposed delayed versus approximate computation offloading framework. For the scope of the evaluation, consider an edge-cloud network, consisting of an edge server, which lies 500 m away from a cloud server, and $N = 10$ users randomly distributed around them. Each user has $K_n = 10$ tasks to be executed exactly at the cloud or approximately at the edge. The user's action space for the RL algorithm is the Cartesian product of the set of all possible numbers of tasks to be offloaded for delayed computing and 100 indicative compression level factors for approximate computing. Also, each agent's, i.e., user's, reward is the weighted sum of experienced delay and accuracy.

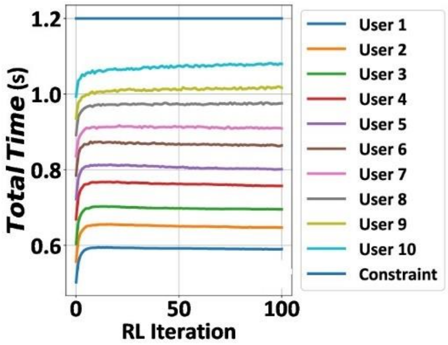


Figure 4-20 Convergence analysis of the devised RL algorithm in terms of each user's total computing time.

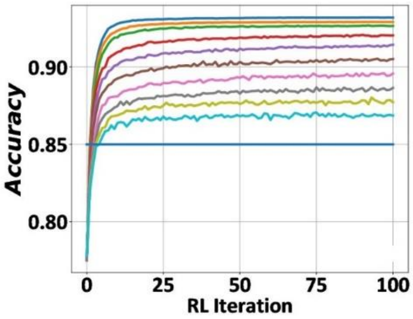


Figure 4-21 Convergence analysis of the devised RL algorithm in terms of each user's achieved computation accuracy.

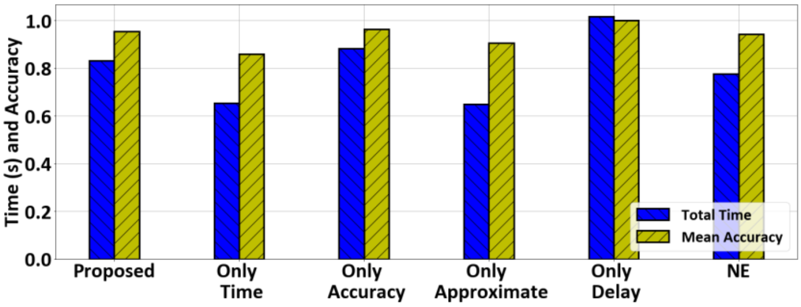


Figure 4-22 Comparative analysis of the proposed delayed-approximate computation offloading framework compared to other benchmark offloading scenarios in terms of total processing time and mean achieved accuracy.

Figure 4-19 depicts for each user the number of tasks x_n offloaded to the cloud and the portion s_n of tasks' data used for approximate computing. Note that as the user ID increases, both the mean data size d_n in bits and the task intensity in CPU cycles/bit increase, resulting in a heavier overall task computational burden. This design choice facilitates straightforward evaluation of the impact of task intensity in the offloading decisions. Also, identical delay and accuracy constraints are considered for all users. The results reveal that as the user's ID increases and the tasks become more intensive, fewer tasks are offloaded to the cloud to meet the user's delay constraints and minimize the transmission time. The data offloaded for approximate computing are more compressed, i.e., a smaller portion s_n of tasks are offloaded. Nevertheless, all users successfully satisfy their delay and accuracy constraints, as further verified in Figure 4-20 and Figure 4-21, which illustrate the convergence of the RL-based algorithm determining the SE regarding the incurred delay and achieved accuracy for each user, respectively. Specifically, the lower ID users achieve lower delay and higher accuracy due to their less intensive tasks, with user IDs presented in the legend of Figure 4-20 for the reader's convenience.

Last, a comparative evaluation of the proposed framework is performed against five alternative scenarios: (i) Only Time and (ii) Only Accuracy: Users' decisions (x_n, s_n) are based on a satisfaction game with their utility focusing solely on time, and on accuracy, respectively; (iii): Only Approximate: All users transmit their tasks to the edge server $(x_n = 0)$ for approximate computing, with compression s_n determined by a satisfaction game; (iv): Only Delay: All users transmit their tasks to the cloud server for delayed computing $(x_n = K_n)$ for all users, with s_n not applicable); and (v) Nash Equilibrium (NE): Users' decisions are derived via a normal-form game with identical utility functions, where each user aims to maximize its utility rather than solely satisfying its constraints. Figure 4-22 illustrates the mean experienced delay and accuracy of users' tasks across the various comparative scenarios. The results show that the Only Time scenario minimizes delay by focusing solely on time constraints but sacrifices mean accuracy. Conversely, the Only Accuracy scenario maximizes accuracy at the cost of increased delay. The Only Approximate scenario significantly reduces delay by leveraging the edge server but lowers mean accuracy as all tasks are approximated. The Only Delay scenario achieves optimal accuracy through exact computing but incurs high latency. Finally, the NE scenario produces similar time and accuracy values to the proposed framework (SE), with the SE slightly increasing delay to achieve higher mean accuracy.

4.4.4 Programmable and context-aware transport

Figure 4-23 shows high-level overview of the proposed context-aware transport implementation. Let's consider that in the transport network there may be possible alternative paths for sending packets to edge server or to data centre. Switches have to choose the right path for each packet, based on the flow tables they contain. For the purposes of this implementation, SDN Domain Controller (P4-based) can provide some additional information, which is not forwarded within packets, to support the process of the path selection. Applying P4 language to create a switch allows the definition of the metadata (context) according to the needs of the operator and the implementation of simple modifications in the future according to the assumptions.

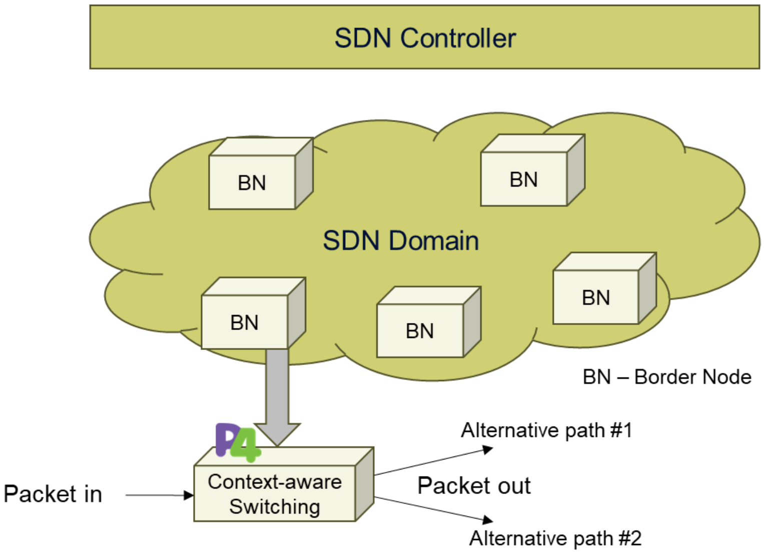


Figure 4-23- High-level overview of context-aware transport implementation.

To define a potential application (e.g. computation offloading scenario discussed in section 4.4.3), let's assume that edge server(s) are located near users and data centre is somewhere in the network. Both could process users' requests, but data centre could be lower operation cost at the cost of higher latency. By default, requests can be processed in the edge servers. Users could choose whether they would like to pay less or have lower latency to a server. To achieve this goal, at least border node (BN) has to know the context of each packet (e.g. preferred data centre over edge server to handle requests in this use case) to be able to modify the default routing decision. Such context is not forwarded within packet but must be obtained from another source, like SDN Domain Controller. Also, such context can vary over time.

The presented implementation has been done using P4 language and is dedicated to two selected switches: Behavioural-Model (BMv2) [BMV2] programmable switch and Native In-Kernel SDN Switch (NIKSS) [OPK+22][NIK24]. At the level of data plane only IPv4 protocol was implemented with extension to support context-aware transport. Details of the implementation are provided in Figure 4-24.

First, incoming packets have to be parsed, by reading the information available directly from its headers, like source or destination IP address. Next, based on that data, metadata can be obtained. This is the right place to read the context of the packet. Using the data from headers and additional metadata, packets could be subjected to additional processing (if needed) before being forwarded.

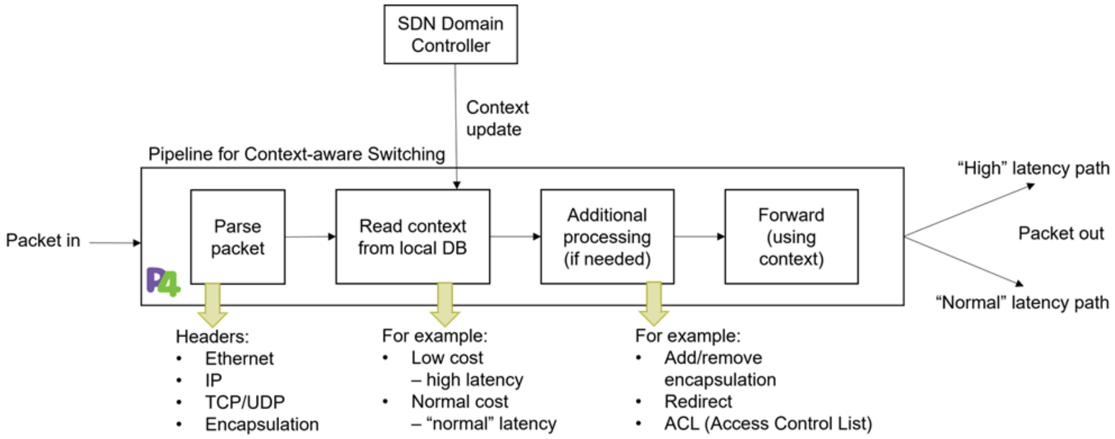


Figure 4-24- Details of context-aware switching.

Context of the context-aware transport has to be identifiable when processing the packet. For this purpose, integer value is used, called context ID. This number is obtained during packet processing pipeline using source IP address, assuming that (in this case) users can be identified by their address. Later, context ID is used to route packets, in addition to the routing table (along with destination address). In this implementation context ID has ternary matching in routing table, so not every entry has to take care of context ID. For example, it is possible to create last resort routing entry, similar to idea of default route. Routing tables and metadata related to context-aware transport (like context ID) are programmed into data plane program. Control plane is responsible for inserting these values. In this study, control plane and link with intermediate modules were not implemented.

Data plane performance measurements of throughput in kpps (kilo packets per second) were performed using IXIA traffic generator and one server running software switch. Both devices were connected with two 100G links each. Experimental setup is shown in Figure 4-24. Measurement methodology was based on [RFC2544]. Each of selected software switch is compared using the same hardware and software environment.

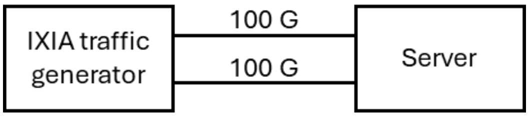


Figure 4-25- Experimental setup for measurement software switches.

Comparison of switches' performance for context aware transport is shown in Figure 4-26. Two above-mentioned P4 software switches, BMv2 and NIKSS were included in the comparison. Moreover, the results were compared with the results obtained for Open vSwitch (OvS) [OVS24] with kernel mode data path. At the same time The OvS switch was configured to forward packets without looking for packet context, whereas the BMv2 and NIKSS switches had dedicated P4 programs running, which implemented context-aware transport. In these cases, the P4 programs include logic to look for packet context. It should be noted that BMv2 is only a reference switch, not intended for production use [BMV2] while in the case of the NIKSS switch better performance results are expected [OPK+22]. Based on obtained results (Figure 4-26), packet forwarding rate for each switch is independent from packet size, but BMv2 has about ten times lower performance than OvS and NIKSS. NIKSS switch has better performance than OvS, but evaluated P4 program is relatively simple.

With more sophisticated programs it is expected that the performance of NIKSS switch will be lower [OPK+22].

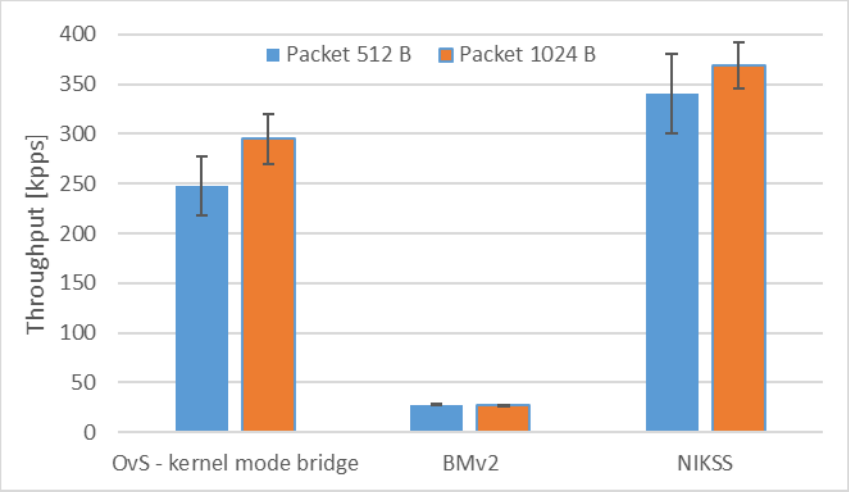


Figure 4-26- Comparison of switches' performance for context aware transport.

As a key take-away, P4 programmable switches give us more flexibility to offer context aware transport and, depending on the switch, comparable performance to Open vSwitch.

4.4.5 Semantic RAN

In this subsection, it is introduced a system model that captures the necessities for a semantic RAN system, based in a robotic use case, structured within a Multi-Layer Graph (MLG) framework (See Figure 4-27). The model breaks down the system's elements into:

1. **Robot Application Classes**: Each robotic application class, referred to as an application π, is defined by three primary parameters: (i) a target latency (L_π), representing the maximum allowable delay between the robot and the edge; (ii) a quality factor (Q_π), such as the required minimum accuracy for a given task; and (iii) the application duration (T_π), which indicates how long the application is expected to run. A binary variable (h_π) is assigned to each application to denote whether its constraints are strict (hard) of flexible (soft). Example application include packet delivery, security surveillance or robot cleaning.
2. **Functions and configurations**: A function $(f \in F_\pi)$ is the basic operation that support the execution of an application. These functions, such as autonomous navigation or object detection, serve as the fundamental building blocks. Applications can be modelled as directed graphs, where vertices represent the functions, and edges denote dependencies between them these dependencies can be bidirectional, indicating that some functions may rely on each other outputs.

   Each function can be implemented with different configurations $(c \in C_f)$. For instance, an autonomous navigation function might use various sensors configurations (e.g., LiDAR or depth cameras), or a computer vision task could use multiple machine learning (ML) models, each with a different balance of accuracy and resource consumption. The combination of a function with a specific configuration defines a "robotic function", denoted as $f_C = (f, c)$.

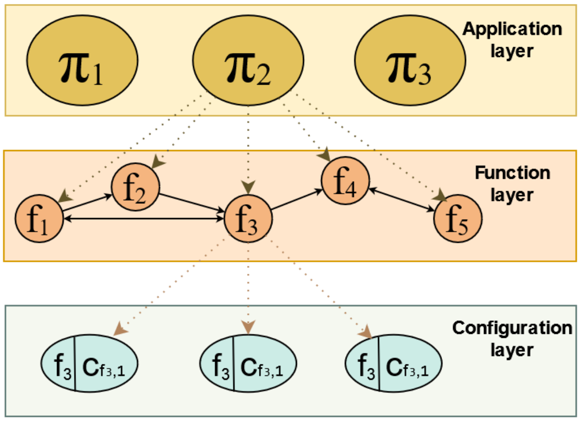


Figure 4-27- Multi-Layer Graph for high level representation of the System Model.

3. **Data Compression Factor**: Different applications tolerate varying levels of data compression, especially in task like image processing for computer vision. To model this, a compression factor is introduced ($z_{f_c} \in (0,1]$), where $z_{f_c} = 1$ indicates maximum data quality, and the quality decreases as the compression factor decreases.
4. **Radio and Computation Resources**: The system leverages resources form the edge server, including radio resources (e.g., Radio Blocks) and computational resources (e.g., CPU, CPU or RAM) for executing functions. These resources are represented by a set $K = \{1, \dots, K\}$ and a vector $S = [S_1, S_2, \dots, S_K]$ that details the available amount of each resource type. The specific resource allocated to a robotic function are denoted as ($s_{f_c} = [s_{f_{(c,1)}}, s_{f_{(c,1)}}, \dots, s_{f_{(c,K)}}]$) , representing the resource allocation for that function.
5. **Offloading Policies**: One of the advantages of using mobile robots in this system s the flexibility to offload certain functions either entirely or partially to local execution. The system supports a set of offloading policies (P), with each function being assigned a specific policy (p_{f_c}). For example, a vision processing task may either be executed at the edge, using a more robust ML model, or locally on the robot, where a lightweight model is used.
6. **Battery Consumption**: The remaining battery level of a robot is represented by $E_{tot} \in [0,1]$, where $E_{tot} = 1$ indicates a full battery, When the battery levels fall below a specified threshold (b), the system directs the robot to recharge. Battery consumption for each function depends on the VPU usage and the offloading policy. The energy consumed by a robotic function is modelled as:

$$e^p_{f_c}(\mu^p_{f_c}, T_\pi) = \phi_{robot} \cdot \mu^p_{f_c} \cdot T_\pi \tag{4-7}$$

   where $\mu^p_{f_c}$ represents the average CPU wake-ups per second for the function, and ϕ_{robot} is the energy consumed per CPU wake-up.
7. **Application Quality**: The quality of a function, denoted as $q^p_{f_c}$, is determined by a quality function $q^p_{f_c}(z_{f_c})$, which depends on the input data's compression level (z_{f_c}).
8. **Application latency**: The latency of a function, represented as $l^p_{f_c}$, is calculated using a latency function $l^p_{f_c}(s_{f_c}, z_{f_c})$, which accounts for both the compression factor and the hardware resources allocated.

The interplay between resource allocation, compression, latency, and quality is non-trivial, particularly given the complex, non-linear behaviours of machine learning models and varying network conditions. Instead of attempting to develop a detailed mathematical model to account for every factor, a data-driven approach is preferred. This approach uses regression models to estimate quality and latency, simplifying the problem while maintaining accuracy.

4.4.5.1 Problem Formulation

Let x_{f_c} be a binary decision variable, where $x_{f_c} = 1$ if configuration c is selected. Two key conditions must be satisfied: (i) if no configuration can be assigned to any function in the set F_π , the application is discarded; and (ii) each function must follow exactly one configuration. In addition to selecting a function configuration, the framework also determines the appropriate offloading policy p from the set of available policies P for each robotic function. A binary decision variable $y^p_{f_c}$ is used, where $y^p_{f_c} = 1$ if policy p is selected. Each robotic function must adhere to a single policy, and if no appropriate policy can be chosen, the entire application class will be rejected.

As might expected, each robotic function must meet the latency and quality targets specified by the application. To achieve this, the framework must determine the appropriate resources s_{f_c} , compression factor z_{f_c}, function configuration c, and offloading policy p, satisfying latency and quality constraint. Since the robots rely on batteries, if the predicted energy consumption exceeds the current battery level, the application cannot continue. Plus, when the battery level reaches a certain threshold, the robot will be sent for recharging, resulting in the termination of the allocated application. Furthermore, the resources allocated to a robotic function at the edge (S_{f_c}) must not exceed the available total budget of resources (S). The following condition ensures this: $S_{f_c} is\ feasbile \leftrightarrow S_{f_c,i} \leq S_i\ \forall i = 1, \ldots, k\ \forall f_c \in F_\pi$.

Considering all these constraints, the goal is to select the optimal function configuration, offloading policy, and compression factor for each function that (i) maximizes the number of robotic functions allocated in the system, (ii) minimizes the resources used per function at the edge, and (iii) minimizes the energy consumption of the robot. This is formalized in the following objective function:

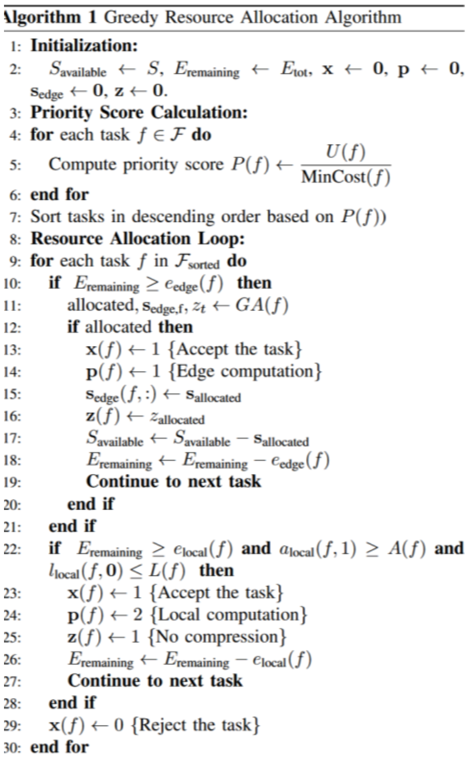
$$max_{x,y,z} \sum_{f\in\mathcal{F}_\pi} \sum_{c\in C_f} \sum_{p\in P} x_{f_c} y_{f_c}^{p} \left(E_{tot} - e_{f_c}^{p}\right) - \frac{1}{K} \sum_{i\leq K} \frac{S_{f_c,i}}{S_i} \tag{4-8}$$

4.4.5.2 Solution: Greedy Resource Allocation Algorithm

To address the optimization problem formulated, the proposal is a Greedy Resource Allocation Algorithm. This algorithm aims to maximize the overall utility by making sequential decisions that prioritize functions offering the highest immediate benefit relative to their resource cost. The Greedy Algorithm operates by evaluating each robotic function's potential contribution to the objective function and selecting configurations and policies that maximize utility while satisfying all constraints. The key steps of the algorithm are: (1) **Priority Calculation:** For each function f_c, compute a priority score based on the expected utility gain and the minimal resource cost required to satisfy the application's latency and quality constraints. (2) **Function Sorting:** Arrange the functions in descending order of their priority scores to determine the sequence in which they will be considered for resource allocation. (3) **Resource Allocation**: Iteratively allocate resources to functions based on their priority, ensuring that resource capacities are not exceeded and that all constraints are met. (4) **Policy and Configuration:** For each function, select the optimal policy and configuration that maximize utility while adhering to latency, quality and energy constraints. (5) **Compression Factor Determination:** Adjust the data compression factor z_{f_c} to balance the trade-off between data quality and transmission latency, optimizing function performance under the selected configuration and policy.

For simplicity of the algorithm, the configuration c is considered to be static, and the two policies available are as follows: Edge Offloading ($edge$), and Local Computation ($local$). Figure 4-28 depicts the detailed algorithm steps. The process starts with the initialization of all vectors that are going to be used (2). Then, at (5) for each robotic function, the framework estimates the priority score given the Utility Estimation (U_f), and the minimum cost ($MinCost(f)$) that determine the minimal resource cost necessary to meet the constraints, considering all feasible policies. Depending on this priority score, the candidate functions are sorted in descending order (7). Now, for each f in the shorted list, if the remaining battery of the robot is bigger than the expected energy consumption for policy $edge$, a Greedy Allocation function $GA(f)$ seeks to find a feasible combination of resources, and compression factor (11) that meets all the above mentioned constrains. When $GA(f)$ finds a feasible solution, the Boolean value $allocation$ is set to "True". If so, the framework accepts the task (13), select $edge$ as policy (14), the resources allocated and the compression factor is stored (15)(16), and current resources available and battery are updated (17)(18).

The second case (22) arises when the remaining battery level of the robot is bigger than the expected energy consumption for policy $local$, and the quality ($A(f)$) and latency ($L(f)$) constraints still be meet. If so, the framework accepts the task (23), sets the policy to be $local$ (24), and the compression factor to be 1 (25), which means there will not be any compression factor due to not upstreaming any data. If none of the above cases fits with the context of the robot, the function is discarded (29), discarding also the application.



```
Algorithm 1 Greedy Resource Allocation Algorithm
Initialization:
    $S_{\text{available}} \leftarrow S$, $E_{\text{remaining}} \leftarrow E_{\text{tot}}$, $\mathbf{x} \leftarrow \mathbf{0}$, $\mathbf{p} \leftarrow \mathbf{0}$, $\mathbf{s}_{\text{edge}} \leftarrow \mathbf{0}$, $\mathbf{z} \leftarrow \mathbf{0}$.
Priority Score Calculation:
for each task $f \in \mathcal{F}$ do
    Compute priority score $P(f) \leftarrow \frac{U(f)}{\text{MinCost}(f)}$
end for
Sort tasks in descending order based on $P(f))$
Resource Allocation Loop:
for each task $f$ in $\mathcal{F}_{\text{sorted}}$ do
    if $E_{\text{remaining}} \geq e_{\text{edge}}(f)$ then
        allocated, $\mathbf{s}_{\text{edge,f}}, z_t \leftarrow GA(f)$
        if allocated then
            $\mathbf{x}(f) \leftarrow 1$ {Accept the task}
            $\mathbf{p}(f) \leftarrow 1$ {Edge computation}
            $\mathbf{s}_{\text{edge}}(f,:) \leftarrow \mathbf{s}_{\text{allocated}}$
            $\mathbf{z}(f) \leftarrow z_{\text{allocated}}$
            $S_{\text{available}} \leftarrow S_{\text{available}} - \mathbf{s}_{\text{allocated}}$
            $E_{\text{remaining}} \leftarrow E_{\text{remaining}} - e_{\text{edge}}(f)$
            Continue to next task
        end if
    end if
    if $E_{\text{remaining}} \geq e_{\text{local}}(f)$ and $a_{\text{local}}(f,1) \geq A(f)$ and $l_{\text{local}}(f,\mathbf{0}) \leq L(f)$ then
        $\mathbf{x}(f) \leftarrow 1$ {Accept the task}
        $\mathbf{p}(f) \leftarrow 2$ {Local computation}
        $\mathbf{z}(f) \leftarrow 1$ {No compression}
        $E_{\text{remaining}} \leftarrow E_{\text{remaining}} - e_{\text{local}}(f)$
        Continue to next task
    end if
    $\mathbf{x}(f) \leftarrow 0$ {Reject the task}
end for
```

Figure 4-28 Greedy Algorithm for Solving the Problem.

The Greedy Approach reduces computational complexity by avoiding exhaustive searches, making it suitable for real-time applications. The algorithm's straightforward logic facilitates easy implementation and integration into existing robotic systems. By considering resource availability and constraints dynamically, the algorithm adapts to changing system conditions.

Some consideration to have in mind is that this algorithm does not guarantee a globally optimal solution, as it focuses on immediate gains rather than long-term benefits. But it ensures that all constraints are strictly enforced, suitable for deterministic scenarios. Efficient resource allocation extends the robot's operational time and improves application performance.

4.4.6 Summary

The context-aware management enabler is summarized in Table 4-7.

Table 4-7 Summary of context-aware management enabler.

Description	Mechanisms to allow network and compute components to dynamically adapt to the context to ensure the expected E2E QoS.
Key take-aways	Measurable KPIs: User device energy consumption, network energy consumption, end-to-end network delay, accuracy of computation task, computing resource utilization Non-measurable KPIs: Reliability, adaptability, flexibility QoS guarantees for a specific slice can be achieved by creating an abstract view of transport resources and network functions, simplifying network management as the network scales. Implementation of context-aware path selection, switching, and packet processing can be achieved using P4 programmability based on the context provided by an SDN domain controller. Optimal semantic orchestrators for robotic use cases to maximize the number of allocated robotic functions in the system while minimizing the consumed energy at the robot ends. Intelligent task offloading decision-making and seamless switching between different computing options, such as (a) delayed computing and (b) approximate computing, to minimize the time and energy consumed across the network.

Requirements	Transport network abstraction techniques must account for and incorporate resiliency parameters to ensure effective fault recovery. Resource allocation and decision-making solutions of low computational complexity are required to ensure real-time network and compute elements adaptation and reconfiguration. Context-aware management mechanisms must account for potential conflicting requirements (e.g., delayed or approximate computations) and be able to determine the most beneficial decision for system performance. Accurate exposure of application, network, and compute components' status and capabilities is essential for effective network and compute adaptation and reconfiguration. Accuracy, Latency and Energy functions are required to determine what the current scenario is and to make appropriate decisions. Programmable Border Nodes are required to fully support context-aware transport.
Standard relations & regulations	The task allocation in semantic RAN affects and is affected by the ETSI MEC architecture [MEC035], the harmonization with 3GPP [ETSI36], and the O-RAN architecture [OAD24, OSA24].

4.5 Fulfilment of the flexible topologies objective

In this chapter, the development and analysis of new access and flexible topologies systems, including multi-connectivity solutions, as well as control and management solutions for programmable and context-aware transport have been described. NTN and trustworthy flexible topologies were introduced as part of the network of networks enabler, which includes NTN architectural options and TN-NTN DC, subnetworks with new node roles (MgtN) and node coordination via various architectural options, the subnetwork CP and novel procedures. Regarding the multi-connectivity solutions, proposals for the CA/DC evolution have been stated in detail, along with a novel procedure for CA, while WCA addresses the multi-RAN approach. Last, solutions for optimizing the use of the transport network infrastructure and packet switching have been presented as part of the context-aware management of transport resources. Additionally, flexible allocation of edge resources and computation offloading decision-making across different computing options have been proposed to adapt the compute infrastructure based on the context.

5 Transformed Architecture for 6G

5.1 Introduction

As detailed in Chapter 3, 6G brings a large set of new services and use cases that will shape how the next generation of (mobile) communications between physical, digital and human world takes place. Despite the excessive potential of this new use cases, a key challenge is how to ensure this shift while achieving the key 6G sustainability targets (that includes not only environmental sustainability but also economic and social sustainability [HEX22-D13]). Most of these use cases (i.e., cobots, telepresence, etc.) require fundamental changes and improvements in the E2E network design to provide continuous coverage and QoS. Chapter 4 details new flexible access mechanisms that provide higher coverage (with respect to 5G equivalent) such as NTN and subnetwork and more flexible deployment options for access and compute networks.

This chapter complements earlier parts by investigating the required architectural changes in the network to ensure a simple operation without loss of performance for 6G. To this end, one key technical requirement of 6G is context awareness in network management. In particular, the high degree of heterogeneity among service types and the conflicting requirements imposed by these services forces providers to reconsider the conventional way of purely relying on the offered service QoS requirements. In 6G, the network architecture needs to support a higher degree of flexibility which would allow itself to be reshaped according to the service KPIs and KVIs.

Modularization gives a new degree of flexibility, revisiting and extending the conventional NF composition to various KPIs and KVIs. Although not being a new concept in literature, the means to extend the modularity to the network design has not been possible until 6G due to the lack of technical enablers. However, the maturity of the virtualisation technology and the extended availability of compute nodes bring modular network architecture design as one of the key aspects of flexible networks (see Objective WPO3.3). Although the description does not suggest standardizing the notion of a module in this chapter, the notion is used to indicate software defined containers that can contain part of a NF or multiple NFs and can be altered according to the needs of the network. This adaptability and high degree of design flexibility leads to interaction and orchestration challenges between different modules and entities (e.g. RAN, core, edge, etc.). This chapter introduces these challenges and proposes several different perspectives and solutions.

Section 5.2 explores the modular 6G design, starting from the analysis on how to design a module considering different composition options and granularities. Then, it further investigates how the modular entities should interact with each other. Following some basic definitions, modularisation examples and their implications on the network performance are presented at the end of this section. Although the impact of these enablers on the 6G blueprint in [HEX224-D23] are quite extensive, a simple mapping is presented in Figure 5-1, where the first three enablers are detailed.

The proposed "transformed architecture" also considers the combination of cloud technology and its extension towards the edge and extreme edge to build a modular, scalable and extendable architecture (see Objective WPO3.3). In this transformed architecture, Sections 5.3 and 5.4 present both new enabling concepts (such as 6G network slicing and intend based management) as well as report the outcome of work in WP6 which tackles the transformation of the architecture enabling the creation of a new Telco Cloud infrastructure gathering and integrating resources from multiple cloud providers and integrating them in single virtual one. This new 6G slicing enabler is mapped with enabler 4 in Figure 5-1.

Following this line of thought, Section 5.3 focuses on the integration of multiple resources from diverse sources into the cloud continuum, as shown by the enabler 5 in Figure 5-1. In the same way, Section 5.4 (in coordination with WP6) introduces the new orchestration paradigms required to operate and construct the Cloud Continuum, in line with enabler 6 of Figure 5-1.

Finally, Section 5.5 analyses the readiness of quantum technology for seamless integration of novel services, such as quantum synchronization and quantum semantic communication, as well as the challenges of trustworthiness associated with these services, is also addressed in this chapter.

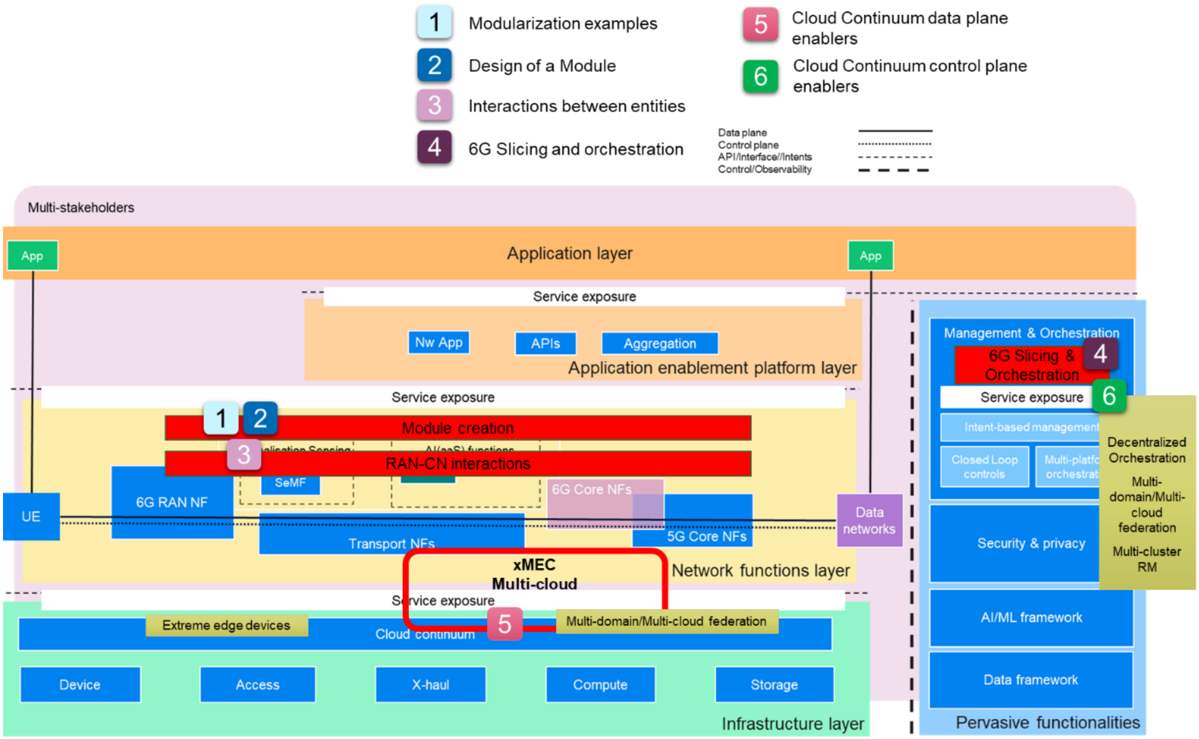


Figure 5-1 Mapping the architectural enablers to the blueprint

5.2 System architecture for 6G core network

Section 5.2.1 describes how the 6G design can be modularised and the key limitations and challenges that would rise from such a modular architecture. Section 5.2.2 further details the holistic design of this new architecture, evaluating different options for the interactions between entities (e.g., RAN, core or edge). Finally, Section 5.2.3 presents examples of how the core NFs and RAN can be modularised and it discusses the implications of these exemplary modularisations on the performance.

5.2.1 Design of a module

5.2.1.1 Introduction

Following a modular design in 6G architecture has many advantages, such as enhancing fault identification, network upgradability, implementation and deployment flexibility, scalability management, etc. However, this comes at the cost of increased complexity in managing and testing the increased number of system's modules as well as in managing and communicating the relevant context data across dependent modules, which may also lead to an increased latency for complex tasks whose execution spans across multiple modules. It is important to evaluate the impact of modularization over different domains to judge on the best level of granularity that should be adopted within each domain. Accordingly, general design principles related to modularisation are first provided in section 5.2.1.2, focusing on principles how to design so called self-contained modules for 6G. In section 5.2.1.3, a study related to redesigning the 5G control plane with more coarse-granular modules is presented along with the evaluation results that reveal the impact of this design choice on the procedure completion time and the volume of communicated messages. Finally, section 5.2.1.4 demonstrates the performance impact of disadvantageous conditions on procedure completion time in Cloud-Native NFs (CNF-based 5G Core). The results showcase the vulnerability of the current 5G design and modularization choices towards the aforementioned networking conditions. Based on the conducted studies, there is a tradeoff when selecting the granularity level of a design. Each section has its own conclusions based on the scenario under consideration.

5.2.1.2 Design principles

The SBA of 5GC was introduced with 5G; and by adopting cloud friendly protocols, the idea was to make the functional architecture more suitable for cloud deployments. Another objective with the SBA was to create a more flexible and extensible architecture where services provided by NFs can be reused to realize other use cases than what was initially intended. One way to validate the extensibility of SBA is to count the number of NFs in the 5GC. In Rel-15 there are 22 NFs in the 3GPP specifications. In Rel-17 the number of NFs is 45. In addition, the number of service-based interfaces (SBIs) has increased at an even higher rate. Although the increase in number of NFs, NF provided services, and related interfaces shows that the SBA is in fact extensible, the actual reuse of NF provided services remains low, at least lower than expected when introducing SBA. Further, having to deploy and operate many NFs can increase the cost of integration and can increase the signalling overhead between those NFs.

With the advent of 6G some new services are anticipated, see section 3, that come with new demands on the architecture. In section 2.2, there is an analysis of some options for the 6G architecture, which provide the levers needed to support the new 6G services.

For 6G, it should be possible to reuse a majority of 5G NFs and only introduce new or re-designed NFs when necessary to support new services or to optimize the system (e.g. to reduce signalling or increase security). Efforts shall be made to make sure that 6G can interwork with 5G. This might be done by allowing a larger degree of bundling of NFs relying on standardized interfaces, hence, enabling a shift from individual NFs to self-contained modules compiled for specific procedures. Thus, a module can be a compilation of functions needed to be a certain procedure or a set of NFs working together. The objectives are that use of modules may reduce signalling and communication overhead, while increasing performance and introducing a single point of contact for the new procedure.

Design principles

Some important characteristics of a module is that it hides some of an NF procedure internals (i.e. make it more self-contained), making room for vendor innovation while reducing the need for complex procedures in standards, e.g. context relocation procedures. The SBA standard provides different alternatives for such modules. In [HEX224-D33] different approaches were discussed.

The following design principles are identified for the establishment of module.

i) **Separation of concerns**. This is achieved by avoiding splitting responsibility for a certain objective of a procedure across different modules (or services), e.g. so that large amounts of information can be contained in the module rather than being signalled several times to different NFs. Consequently, when possible, merge functions that have tight connections. The motivation for this principle is to reduce signaling between NFs, thus improve manageability and possible evolution.

ii) **Information only by request**. This principle states that the NF interested in information requests the information. With this principle the network avoids providing unsolicited information that it is not needed. Note that there may be cases where this principle might impact performance in which case it has to be weighted whether to give more importance to this principle or to performance.

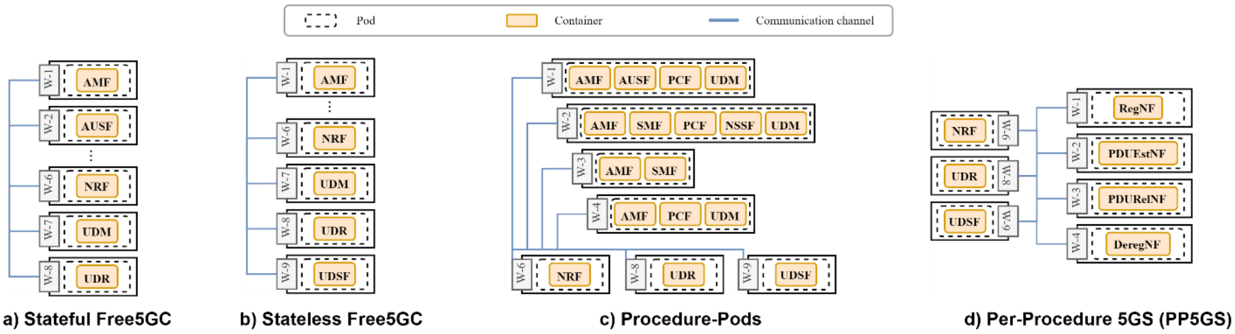
iii) **Avoid unnecessary functional proxies**. The intention of this principle is that all modules can communicate directly with other modules, without some proxy between them. With this principle long signalling chains can be avoided. Further, proxying in application logic is avoided which leads to less coupling between NFs. However, there is (at least) one case where there benefits of using a proxy outweigh this principle, namely the AMF between the RAN/CN and the UE and other NFs, e.g. AMF guarantees that there is a single NF in charge of handling UE mobility and UE reachability (instead of duplicating such functionalities to many CN NFs), etc. (see Section 5.2.2 for more analysis).

Smaller transactions: If a procedure requires that more than two modules need to communicate; aim to redesign the modules to reduce the need for inter-module communication.

5.2.1.3 Procedure-based functional decomposition:

The control plane of 5G core system is designed as a group of NFs that interact with each other based on defined procedures to deliver a certain functionality such as registering the UE to the system or establishing a PDU session. In the following, a new design for the control plane is examined, where each system procedure is consolidated into a single NF. For example, the UE registration procedure is designed and implemented as a single procedure-based NF where the execution logic, used to span multiple NFs (AMF, AUSF, PCF, NRF, UDM, and UDR) in 5G, is consolidated into a single NF. This work is based on [GSH+23], where more details on the design, implementation and evaluation can be found.

In the following, the conducted evaluation to quantify the impact of this coarse granular design on the system's performance in terms of the volume of signaling and procedure completion time is elaborated.



a) Stateful Free5GC b) Stateless Free5GC c) Procedure-Pods d) Per-Procedure 5GS (PP5GS)

Figure 5-2. Different evaluated implementations and deployments.

The open source Free 5GC [Free5GC] implementation is used as a baseline to implement the procedure-based functions. Four implementation and deployment options, shown in Figure 5-2, are evaluated:

- Stateful Free 5GC (Figure 5-2a): This is the baseline implementation where each 5GC NF is deployed in a separate Node.
- Stateless Free 5GC (Figure 5-2b): This is a modified version of the baseline 5GC implementation where statelessness is implemented in this system by deploying the UDSF instance to be used by NFs to store their application state/context.
- Procedure-Pods (Figure 5-2c): In this deployment, the NF instances involved in a certain procedure are deployed in the same Pod. The inter-NF communication takes place within the Pod except for traffic sent to NRF, UDR and UDSF which are shared among all NFs and deployed once for the entire cluster.
- Per-procedure 5GS (PP5GS) (Figure 5-2d): This is the procedure-based NFs described earlier. In this case the inter-NF dependencies are broken while executing the control plane procedures. While each per-procedure NF is deployed in a separate node, the NRF, UDR and UDSF are shared among the NFs under test.

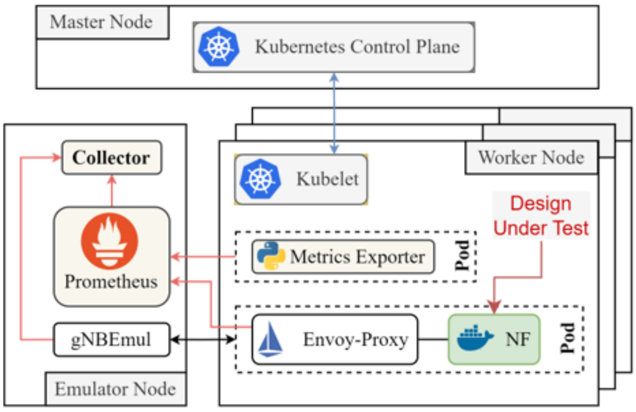


Figure 5-3. Setup for the evaluation of the different systems. The cluster is orchestrated using Kubernetes.

A cluster of 9 workers is built and orchestrated using Kubernetes [K8S] as shown in Figure 5-3. A gNB emulator is implemented to mimic requests coming from UEs to the 5G core system. Different metrics are collected using Prometheus software [PRO22]. Four different control plane procedures are evaluated: UE Registration, PDU Session Establishment, PDU Session Release, UE Deregistration. The inter arrival time for new UEs is set to 3ms (uniformly distributed). The duration of each measurement campaign is set to 45 seconds and repeated 8 times per scenario.

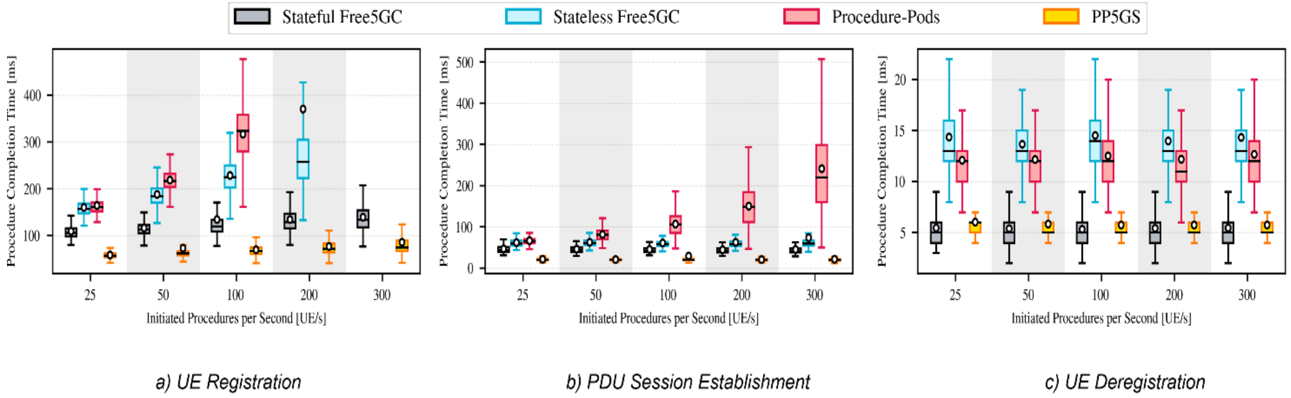


Figure 5-4. Procedure completion time as a function of initiated procedures per second for different control plane procedures

The Procedure Completion Time (PCT) (in ms) for the UE Registration and PDU Session Establishment and UE Deregistration control plane procedures as a function of initiated procedures per second (in UE/s) are measured and displayed in Figures 5-4a, 5-4b and 5-4c, respectively. The PCT for the Registration procedure and PDU Session Establishment procedure are respectively 42% and 50% shorter in the case of PP5GS compared to Stateful Free5GC. On the contrary, the PCT for the Deregistration procedure is 8% longer (~1 ms) in the case of PP5GS compared to Stateful Free5GC. In general, we observe that PP5GS reduces PCT for complex procedures.

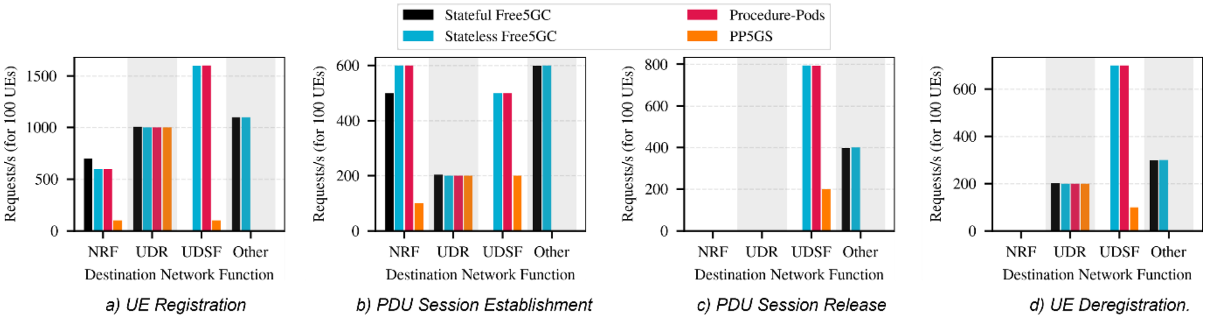


Figure 5-5. Number of requests per second triggered at different destination NFs in different control plane procedures.

Figure 5-5 shows the number of HTTP requests per second triggered at different destination NFs during the execution of different control plane procedures. The measurements are collected for the scenario where there are 100 new UEs introduced to the system per second. It is observed that the requests per second for the UE Registration procedure is 57% and 72% less in the case of PP5GS compared to Stateful Free5GC and stateless Free5GC, respectively. The requests per second for PDU Session Establishment procedure is 61% and 72% less in the case of PP5GS compared to Stateful Free5GC and stateless Free5GC, respectively. In the case of PDU Session Release procedure, the requests per second is 50% and 83% less in the case of PP5GS compared to Stateful Free5GC and stateless Free5GC, respectively. Finally, the requests per second for the UE Deregistration procedure is 40% and 75% less in the case of PP5GS compared to Stateful Free5GC and stateless Free5GC, respectively.

In conclusion, a coarse granular design of the control plane of the core network using a design such as the PP5GS can reduce the signaling between NFs, shorten the PCT as well as simplifies the management of the UE context and NFs state. However, this comes at the cost of reduced flexibility in deploying the coarser granular modules as well as increased overhead in terms of fault isolation and less efficient scalability management of the coarser granular modules.

5.2.1.4 Performance impact of disadvantageous conditions in CNF-based 5G Core

As alternative deployment models of the 5G and 6G Core Network are considered, both via designing and creating new Network Functions (DataF, AIaaS), and via evaluating a disaggregated core network (both logically and geographically, such as Cloud Continuum), it becomes more apparent that these enablers should not occur at the cost of reliability of the network. Therefore, our study explores the reliability of the SBI-based 5G Core Network, subjecting different Network Functions to Chaos Engineering [CHE24] experiments to model faults and failures occurring in real systems.

As section 5.2.1.3 demonstrated, there is a trade-off between performance and flexibility when considering the design of a modular 6G architecture. As the adoption of virtualization and CNF-based deployments progresses in order to enable more flexible networks, the complexity level rises in line with modularity. Moreover, as the networks are comprised of multiple dynamic and ephemeral elements, reliability becomes harder to guarantee.

In 5G Core, sources of delay in procedure completion time are (1) Inter-NF networking, (2) Insufficient compute resource allocation and implementation inefficiency (3) Resource contention and I/O Bottlenecks (4) Node-related hardware bottlenecks (CPU scheduling, Memory allocation). In this contribution, the focus is placed on Inter-NF networking.

The PCT for the user registration procedure is measured and treated as a benchmark value for insights into vulnerability of different Network Functions towards network-related Chaos Experiments.

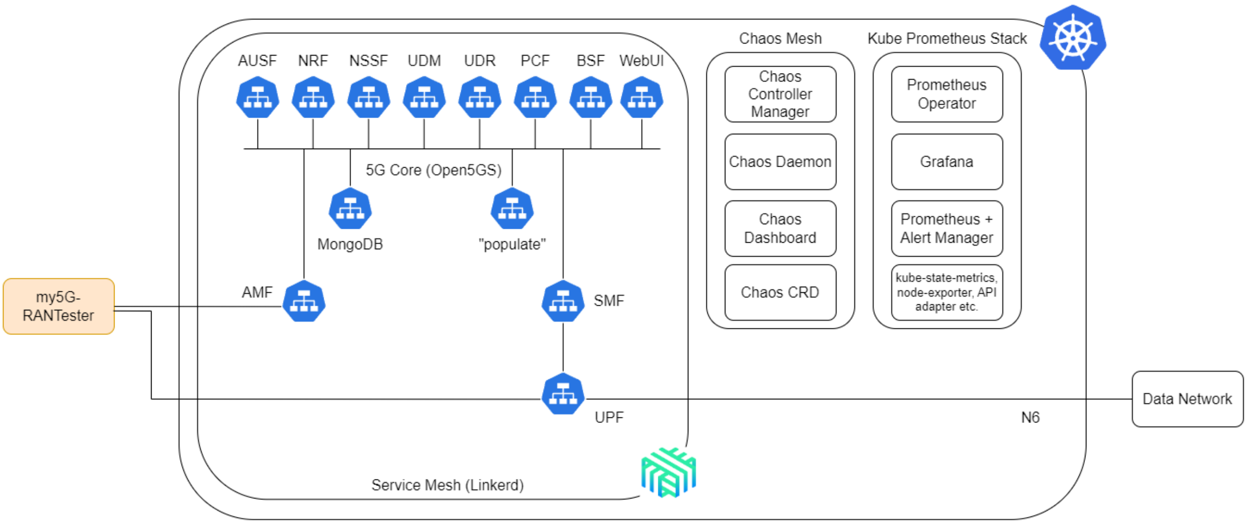


Figure 5-6 Experimental Setup

The Experimental Setup is shown in Figure 5-6. On the Kubernetes Cluster [K8S], we deployed the Open5GS Open Source 5G Core implementation (version 2.1.0, via a Helm Chart [HEL24]). On top of that, the Kube Prometheus Stack [PRO22] was deployed to monitor the resource usage and "health" of the Network Functions.

For generating traffic, we have chosen my5G-RANTester [MY5G], which is an open-sourced gNB/UE emulator, configured with the necessary parameters (such as the mcc, mnc, sst, sd and msin) to allow it to conduct the 5G registration and PDU session establishment procedures, initiated via communicating with the AMF Network Function.

The registration and PDU session establishment procedures were performed for the same user in a loop – as soon as the combined procedure was finished, the user was de-registered and the gNB and UE were terminated. After that, the procedure was once again carried out.

For modelling the failures in the Experimental Setup, a Chaos Engineering tool, Chaos Mesh [CHM24] was used. Chaos Mesh is compatible with Kubernetes and allows us to conduct Chaos Experiments for selected pods, which encapsulate Network Functions.

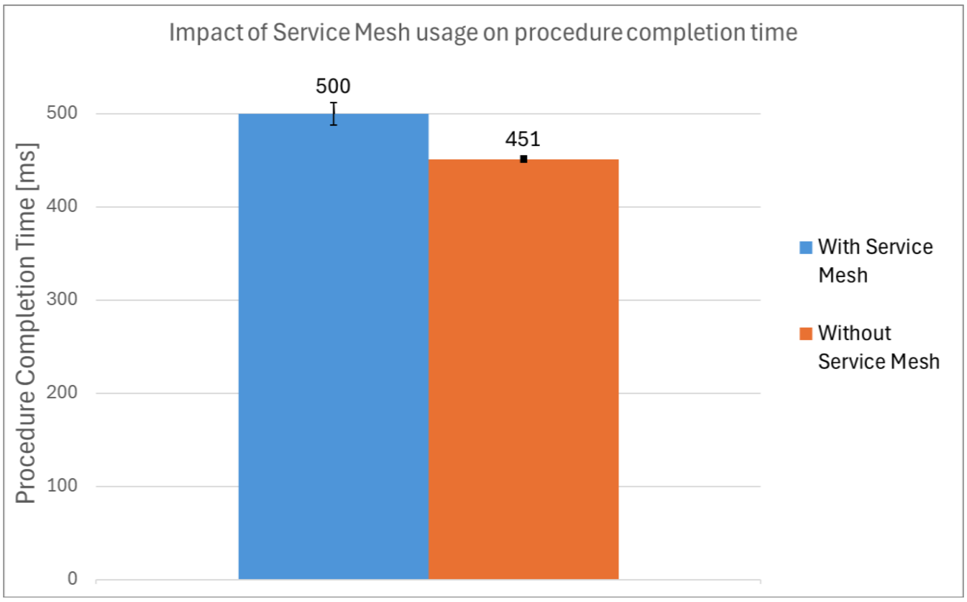


Figure 5-7: Increase in Procedure Completion Time caused by the addition of Linkerd Service Mesh

Introducing the Linkerd Service Mesh into our 5G core deployment results in 10.8% increase (visible in Figure 5-7) in the Procedure Completion Time for the Registration procedure.

Performed Chaos Experiments[3] were focused on emulating the following type of failures:

- Networking failures – focused on emulating disadvantageous networking conditions for network functions, one at a time. We tested the impact of added delay and jitter for seven network functions that play the important role in the registration and PDU session establishment procedures.

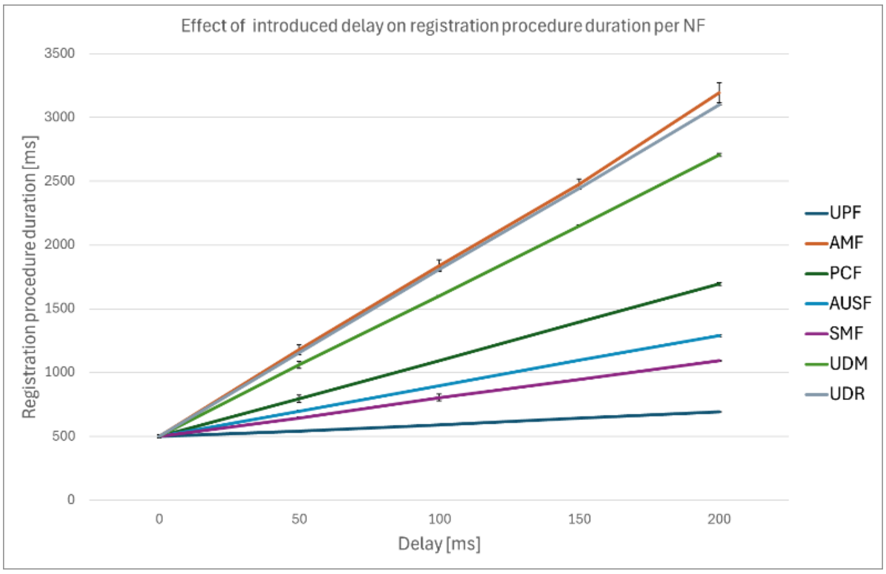
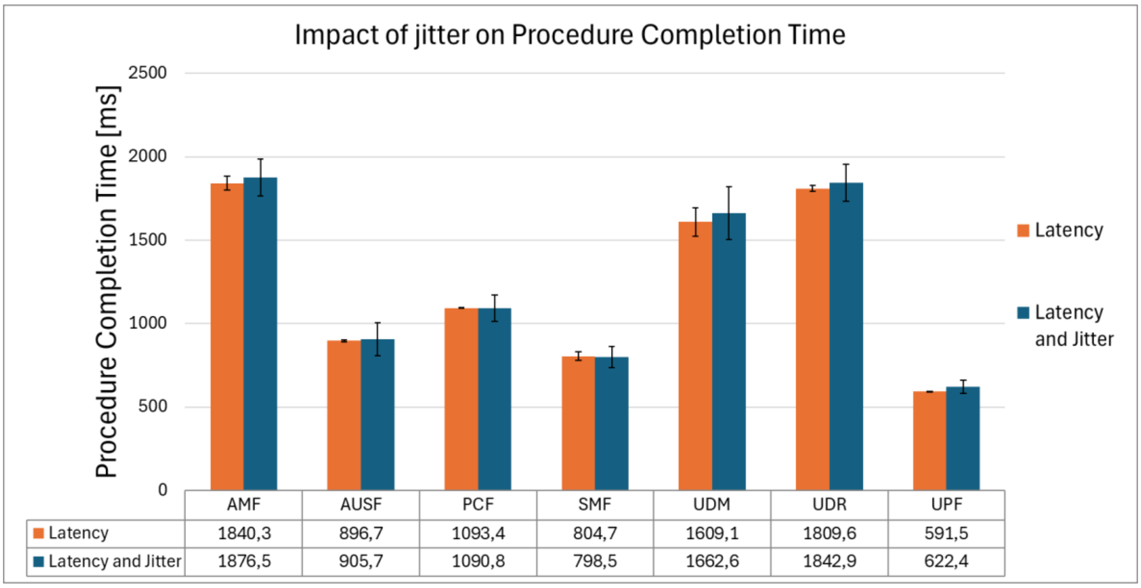


Figure 5-8 : Effect of introduced delay on registration procedure duration per Network Function

The obtained experimental results (see Figure 5-8) show the effect of added delay towards a single Network Function on procedure completion time. The linear effect is to be expected – the number of times a given Network Function participates in the 5G procedure is constant. The increase of PCT values for a single Network Function represents the number of times that the Network Function participated in the procedure.

[3] NetworkChaos - Fault type name from Chaos Engineering tool, Chaos Mesh

The outstanding impact of additional delay introduced towards the AMF function is a direct result of how often the AMF function participates in the registration procedure, however the results for UDM and UDR functions are more significant than anticipated. The increase in PCT can occur due to Open5GS utilizing a MongoDB database instance to store client related information, making the connection between UDR and the MongoDB database vulnerable to introduced network Chaos.



	AMF	AUSF	PCF	SMF	UDM	UDR	UPF
🟧 Latency	1840,3	896,7	1093,4	804,7	1609,1	1809,6	591,5
🟦 Latency and Jitter	1876,5	905,7	1090,8	798,5	1662,6	1842,9	622,4

Figure 5-9: Effect of introducing 100ms of Jitter alongside 100ms of delay on registration procedure duration per Network Function

Next, jitter was introduced in addition to delay towards selected network functions. The results can be seen in Figure 5-9, where the introduction of jitter hasn’t resulted in a significant increase of PCT. For most network functions, the increase was less than 3%, with the User Plane Function (UPF) seeing the biggest increase, at 5.24%.

Further analysis of vulnerability of 5G Core towards disadvantageous networking conditions is available in the Annex A, section A.8.

In conclusion, introducing Service Mesh to the 5G Core deployment resulted in a double-digit percentage PCT increase. The Service Mesh proxies participating in communication between network functions have impacted the PCT. Based on the obtained results, a significant impact of Chaos Experiments introduced towards the UDM and UDR network functions on the procedure completion time of the registration procedure can be observed. Open5GS utilizes the MongoDB database for storage of client-related data. Due to this, the increased communication between the network functions and the database instance is vulnerable towards network degradation. Introducing jitter along with delay towards network functions has not shown to impact the average PCT. Depending on the value of jitter, the range of extreme values increases, but the average value of PCT remains similar.

5.2.1.5 Summary

Table 5-1 summarizes the design of a module enabler.

Table 5-1 Summary of the analysis on design of a module

Description	There is a tradeoff between performance and flexibility when considering the design of a modular 6G architecture. As the adoption of virtualization and CNF-based deployments to enable more flexible networks increases, the complexity level rises in line with modularity. Moreover, as the networks are comprised of multiple dynamic and ephemeral elements, reliability becomes harder to guarantee. This enabler analyses the different module creation options and their requirements and impacts on the network (both advantages and disadvantages).

Key take-aways	A new design, i.e., PP5GS, of Core CP was introduced and implemented, where the new group of NFs encompasses all the interactions between 5G Core NFs to complete a certain procedure. Quantitative analysis of the new design revealed: • Gains in terms of time needed to complete the execution of a certain control plane procedure • Reduction in the total number of messages needed for these procedures However, this design reduces the flexibility in deploying the more coarse-grained NFs. A more fine-granular design results in higher flexibility in implementing and deploying modules but at the cost of reduced performance in terms of execution time and state management. The introduction of Linked Service Mesh into the 5G Core deployment resulted in an increase of 10.8% in the Procedure Completion Time when using a Service Mesh with proxies to participate in communication between network functions. While these proxies provide valuable information regarding traffic inside the network, for use-cases with more stringent requirements such an increase in PCT may be unacceptable. Emulating disadvantageous networking conditions for network functions has showcased a vulnerability of AMF, UDM and UDR, resulting in a disproportionate increase in Procedure Completion Time compared to other network functions. Fine granular modularisation comes at the cost of an increased overhead in signalling between modules, as well as more complexity in managing more modules.
Requirements	Virtualized NFs and Cloud-Native NFs

5.2.2 Interactions between entities

5.2.2.1 Introduction

This enabler explores on how different entities shall interact in future generations of cellular networks (5G Advanced/6G). The interface of primary interest is the RAN-CN interface, also known as N2 interface. Requirements for a 6G RAN CN to RAN interface should include cloud readiness, load balancing and support in the ecosystem. In this section, the pros of cons of the existing P2P connection are discussed. The usage of SBI has been introduced as part of adopting the SBA in the 5G core. With SBI, the focus is on defining the services provided by a producer, and how a consumer could request such services, rather than designing a different point-to-point interface between each producer-consumer pair. In addition to this design shift, SBI also brought the adoption of HTTP/2 stack. Furthermore, a service framework (registration, discovery, authorization, inter-service communication) has been introduced as part of SBA.

As mentioned in [HEX224-D33], the RAN communicates with CN using the N2 interface and always communicates via the AMF (regardless of the action network function in CN being the “final destination”). In 5G, any CN function accessing RAN needs to go via the AMF (and vice versa). Procedures supported over N2 are for example PDU Session Management, UE Context Management, UE Mobility Management, Paging, Transport of NAS Messages and NG Interface Management.

The N2 interface supports the NG application protocol (NGAP). An NGAP [38.412] packet is encapsulated by IP and Stream Control Transmission Protocol (SCTP), which are more of generic purpose protocols (i.e., are found in other specifications than 3GPP). The existing control plane interactions and interfaces are depicted in Figure 5-10.

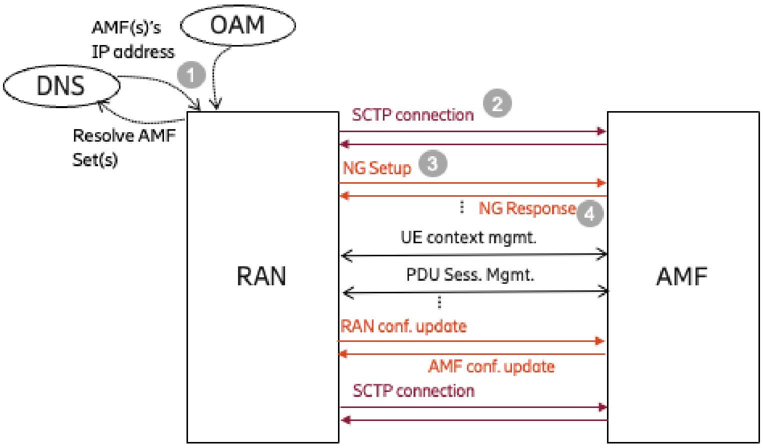


Figure 5-10 Current RAN-CN interface and NGAP procedures.

The NGAP procedure for setup (see Figure 5-10) can be described as follows:

0. (not shown). RAN is provided with AMF Set(s). The AMF Set ID is a part of the Globally Unique AMF ID (GUAMI) that identifies a set of AMFs. All AMFs within the same set share the same AMF Set ID. [23.003]
1. RAN gets the AMF(s)'s IP address(es) via either DNS or OAM.
2. RAN initializes an SCTP connection towards the selected AMF's IP address.
3. RAN initializes NG Setup towards the AMF.
    a. With NG Setup, AMF becomes aware of RAN and of its configuration (RAN ID / Name, List of configured TAs / PLMNs / Slices, etc.).
    b. Sets up a so called Transport Network Layer (TNL) "association" between RAN and AMF.
4. AMF responds with NG Response to the RAN.
    a. With NG Response, RAN becomes aware of AMF configuration (AMF name, Relative capacity, list of GUAMIs / PLMNs / Slices, etc.)

[HEX224-D33] evaluated the cloud friendliness of the NGAP procedures and the possibility to replace the NGAP functionality with using SBA interface. The conclusion is that the NGAP functionality may be handled by SBA by e.g., NRF via Service discovery/registration and status notification. However, there are some procedures that cannot directly be mapped to any SBA functionality, such as error indication or how to include functionality to make RAN operational (done via NG setup earlier).

There are three main tracks in this enabler:

- RAN-CN control plane interactions and interfaces (see section 5.2.2.2)
- Cloud friendliness of N2 interface (see section 5.2.2.3)
- Data centric service-based architectures (see section 5.2.2.4)

The "RAN-CN control plane interactions and interfaces" track addresses how to manage the demands of emerging 6G services and applications, such as immersive communications, sensing, and ambient IoT. It also investigates the migration aspects with 5G. The next section 5.2.2.3 "Cloud friendliness of N2 interface" addresses means to improve or replace SCTP. Finally, in section 5.2.2.4 the "Data centric service-based architecture" track aims to better fulfilment of 6G requirements, emphasizing flexible service routing for distributed resources and supports dynamic stateless and loosely decoupled highly granular NFs.

5.2.2.2 RAN-CN control plane interface evaluation

The key requirements for RAN-CN interactions and interfaces are as follows.

- RAN-CN CP interaction for 6G shall adhere to the 6G system architecture design principles.
- RAN-CN CP interaction for 6G shall not hinder the deployment and uptake of 6G SA.
- RAN-CN CP interaction for 6G shall not break the support of the baseline connectivity services and their requirements.
- RAN-CN CP interaction for 6G shall enable the support of envisioned new services and their requirements.

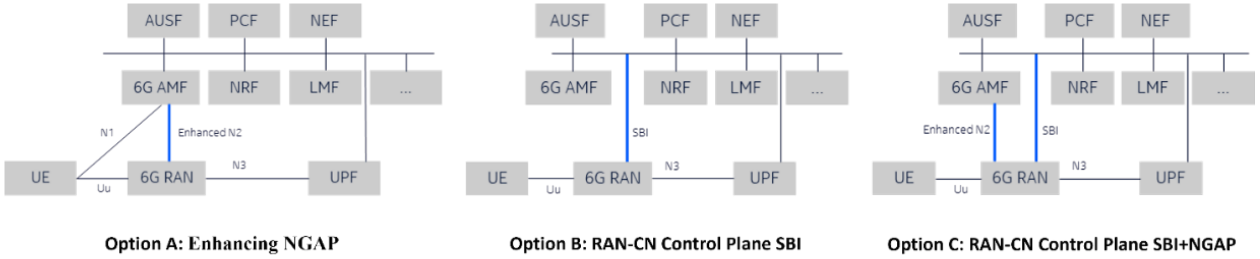
- RAN-CN CP interaction for 6G shall support the various RAN deployments and architectures.
- RAN-CN CP interaction shall be efficient in terms of such as latency, payload size, reliability, availability and resilience.

Considered RAN-CN CP options for 6G:

Option A: 6G RAN communicates via NGAP with the AMF, or a 6G equivalent of it. RAN maintains ASN.1 encoded interfaces and 6G AMF maintains the gateway functionality between RAN and CN NFs.

Option B: the RAN-CN interface is changed by introducing a SBI between 6G CN and RAN. The 6G RAN directly exposes and consumes services towards and from the CN.

Option C: a hybrid approach where, in addition to the NGAP interface, a service-based interaction is introduced with minimal impact on the existing point-to-point interactions and without creating duplication among the two interfaces.



Option A: Enhancing NGAP

Option B: RAN-CN Control Plane SBI

Option C: RAN-CN Control Plane SBI+NGAP

Figure 5-11 Considered RAN-CN CP options for 6G (option A-C)

Evolution of the RAN-CN interface for the new 6G services

In Table 5-2 the implications of the different RAN-CN CP options in Figure 5-11 are evaluated. Although some limitations have been identified with Option A using an enhanced NGAP interface [HEX224-D33], there are several advantages of that option. One important aspect is that since the communication between RAN and CN goes via a proxy (the AMF), RAN and CN are isolated from each other. Staying with the point-to-point RAN - CN interface in option A does not prevent possible protocol enhancements on that interface, e.g. NGAP could be enhanced and/or the underlying SCTP be enhanced or replaced, thereby addressing some of the “limitations” and improving the overall performance on that interface. One additional benefit with option A is that during a change of gNB, the active/idle state change needs to be communicated to and known/managed by only one NF, the AMF. In addition, the UE needs to establish and maintain only one NAS association (including security). Further, the current solution inherently supports (both 5G and) 6G if option A is selected for the future architecture. Overall, despite its issues, the current N2 solutions has many positive characteristics and is a well-proven multi-vendor interface between RAN and CN.

Option B (SBI) could help avoiding the interface specific managements procedures and instead there could be one service framework (registration, discovery, authorization etc.) common to both RAN and CN. However, the gains of such changes, due to needed standardization etc, are difficult to quantify. One possible positive implication of direct interaction between RAN and CN services could be that the RAN-CN procedures will have shorter latency since the (AMF) proxy is removed. On the negative side, due to removal of the AMF as a security point, so called “CN hardening” is necessary. This means that all CN NFs must have improved security. Another negative aspect is that this option requires higher testing and integration burden, due to the many relations to CN NFs (establish, update at mobility, dependencies, etc.), multiple NAS terminations (security risks, maintenance, dependencies, etc.). In option B all CN NFs that communicate with the RAN need to track UE mobility, status and maintain an association. There may also be race conditions, when two CN NFs do not agree on UE location and state - difficult to sort out unless the CN NFs talk to each other - defeating the purpose of removing the proxy. Yet another negative aspect is that the 5G to 6G interworking is not inherently supported.

The third “hybrid interaction” solution Option C would allow having a single UE anchor point in the CN for UE related services, both communication related and non-communication related. The new interface would then carry only non-UE related services. Since 5G only has UE-related services, this solution would also be backwards compatible and support the migration from 5G to 6G, but only for the 5G services. Also, this option may be complex since it requires implementation of two different solutions (side- by side) and it also induces higher testing and integration burden than the “proxy solution” (i.e. option A).

The main conclusion from this analysis is that Option A is currently the preferred option as it has positive valuation results compared to options B and C.

Table 5-2 Evaluation of the CN-RAN interface for the new 6G services

	Option A	**Option B**	**Option C**
Features	Well-proven and tested design based on existing architecture and interfaces already used and defined. 6G RAN communicates with 6G AMF via enhanced RAN-CN interface (enhanced N2). RAN maintains P2P interfaces. 6G AMF maintains the gateway functionality.	Direct RAN-CN NF CP communication where 6G RAN and CN NFs expose services towards each other and communicate directly via RESTful APIs over SBI. Cloud friendly mechanisms and protocols used to offer security framework harmonization and resiliency mechanisms support. Supports native compute/storage separation.	Existing connectivity and communications support maintained over P2P interfaces, extended for 6G radio support. New 6G services (e.g. sensing, AI) may be enabled over SBI, and direct RAN-CN NF communication is enabled for those services.
Implications	Separate domains with positive RAN/CN isolation. Similar implementation and standardization compared to 5G. Possible improvement in cloud-friendliness by using enhanced Transport and Application protocols (see section 5.2.2.3). Security inherently supported due to the clear split between RAN and CN and reusing 5G security. The proxy may add delay to the procedure. Inherently supports 5G to 6G interworking and migration.	Requires higher standardization effort and increased testing, verification, and implementation efforts. Additional efforts for handling coordination and race conditions (such as mobility and UE states). Performance degradation and more processing power requirement due to the overhead introduced by clear-text based JSON encoding and decoding and larger message sizes. No clear domain separation. Limited architectural benefits compared to direct RAN-CN communication as 6G AMF is the consumer to most of the current RAN services. Complicated security handling due to multiple UE anchor points in CN, i.e., there is no natural RAN-CN isolation which means “CN hardening” would be required. It might be possible to cut system procedure delay due to no CN NF proxy hop. Multiple NAS terminations (security risks, additional maintenance, dependencies, …). Interworking with 5G not inherently supported	Requires higher standardization efforts due to need to support both SBI and P2P interfaces. Performance degradation and more processing power requirement is expected for SBI due to clear-text based JSON encoding and decoding and larger message sizes Complex development and implementation efforts due to dual stack requirements in 6G RAN and in the CN. Increased efforts for handling coordination and race conditions for new 6G services using SBI. Additional configurations required to map services and supported service instances across the nodes. AMF (or 6G equivalent) is relieved of proxy task for non-communication service. Supports 5G to 6G interworking for 5G services. Higher testing and integration burden. These additional requirements and efforts are not friendly for smaller footprint RAN deployments (Pico, micro, etc.)

Conclusion	Option A is currently the preferred option as it has positive valuation results compared to options B and C.

5.2.2.3 Cloud friendliness of N2 interface

This section continues the evaluation of the N2 NGAP functionality done in [HEX224-D33]. As mentioned above, the current N2 uses the transport protocol SCTP, a protocol that was designed primarily for the telecommunications industry's needs. SCTP supports features like multi-homing, multi-stream, and semi-permanent associations. Regarding cloud support, SCTP has some limitations, such as lack of load balancers due to the complex state machine and multi-homing, difficulties with Network address translation , etc. As a result, telecommunication vendors must develop their own solutions when using SCTP in cloud deployments. In addition, the community support around SCTP is limited compared to other mainstream solutions, which eventually results in a slower improvement pace.

The NG protocol supports the following capabilities [38.410] [HEX224-D33]:

- procedures to establish, maintain and release NG-RAN part of PDU sessions;
- procedures to perform intra-RAT handover and inter-RAT handover;
- the separation of each UE on the protocol level for user specific signalling management;
- the transfer of NAS signalling messages between UE and AMF;
- mechanisms for resource reservation for packet data streams.

For a cloud friendly protocol, the CP application logic design should be considered E2E. In the following we provide a brief comparison between the current N2/NGAP solution (with SCTP), an SBI solution and an NGAP solution where SCTP is replaced with a modern protocol. For more details on the various alternatives see D3.3.

In the current N2/NGAP the logic design is coupled to SCTP (see Figure 5-12 left). In conclusion, the NGAP/SCTP solution provides a high level of functionality suited for N2, however, the solution is not particularly cloud friendly.

Using SBI instead, the application layer is designed with an E2E approach, decoupled from the transport layer. Furthermore, no binding is needed since HTTP connections and cloud can do load balancing (see Figure 5-12 middle). This alternative is cloud friendly. However, SBI needs to be extended to provide the same level of functionality as the current N2 solution.

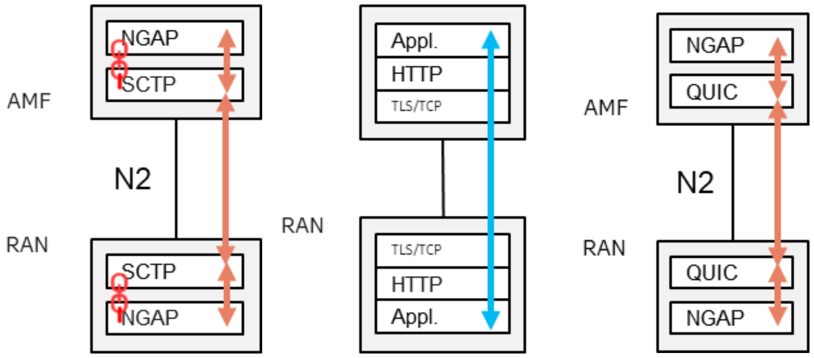


Figure 5-12 Options for the 6G RAN to CN interface protocol

Finally, in the solution where NGAP is used with a more modern protocol, e.g. QUIC, the NGAP logic with E2E features is decoupled from the transport layer (see Figure 5-12 right). There is no TLNA binding/management. The solution handles losses/failures throughout the operation. Thus, with this solution the best parts of NGAP will remain. Further, no binding is required and consequently no changes are needed for the application logic. Also, the NAT friendliness and the inbuild E2E transport layer security with QUIC enables the more dynamic cloud deployments.

5.2.2.4 E2E module interfaces and interaction

As 5G networks evolve beyond the centralized cloud-native architecture of 5G SBA, they are expected to become fully distributed across the entire compute continuum. This shift requires new service routing capabilities for optimal resource utilization. To address this challenge, two key concepts are introduced: Data-Centric Networking (DCN) and Data Flow Programming (DFP). DCN shifts away from traditional IP-style addressing and focuses on routing based on application-specific data, making it more suited for 6G's Edge-Native network requirements. When combined with DFP, which models applications as directed graphs with operators and FIFO data streams, it enables the creation of serverless, dynamic, and highly granular NFs. This approach supports a flexible and scalable architecture, improving efficiency, resiliency, performance, and security across distributed 6G systems.

In this work, we have proposed the application of data-centric paradigms to the SBA. Dataflow-based service composability is proposed as a solution to integrate these approaches seamlessly, fostering innovation and customization for emerging applications [BQG+2024]. It also simplifies development for programmers, operators, and service providers. Nevertheless, this approach will also encompass a higher integration and validation effort sue to the higher number of NFs, albeit simpler. The proposed architecture (Figure 5-13) uses a hierarchical, layered structure, where each layer builds upon the previous one, providing abstractions that ease development and management. This approach ensures flexibility, scalability, and the ability to meet the complex demands of Edge-Native 6G networks and aligns with the Hexa-X-II E2E blueprint, focusing on the network’s functions layer:

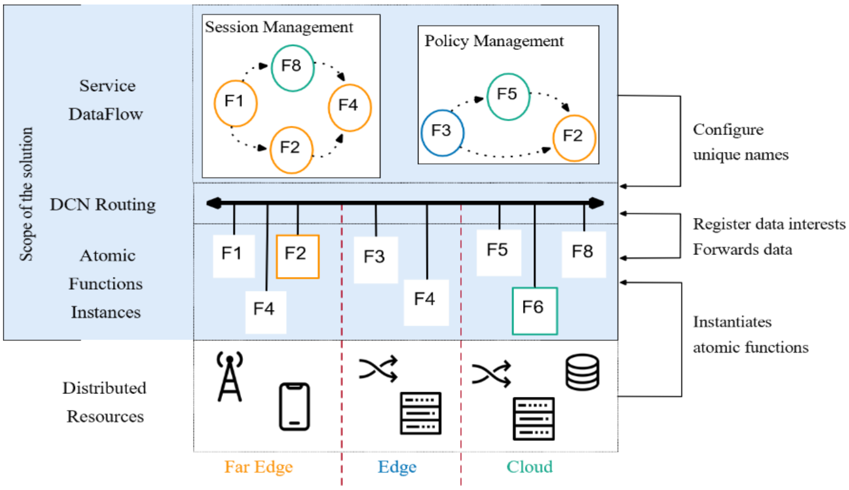


Figure 5-13: Data-centric Service-Based Architecture for Edge-Native 6G Network

1. Service Dataflow: This approach enables the creation of complex services by combining granular components into service chains. It supports serverless and stateless functions that can run in parallel as long as their data dependencies are met. The service dataflow model represents services as a directed graph, with dynamic interconnections and lifecycle management of functions, including tasks like orchestration, scheduling, and resource-efficient placement.

2. Data-centric Service Routing: This layer addresses challenges in routing communication between dynamic, granular functions, offering location-transparent communication. It enables data to be routed based on its name, rather than its origin, supporting efficient, scalable, and robust service routing. It improves performance, reduces overhead, and facilitates the sharing of granular functions across dataflows.

3. Highly Granular/atomic Function Instances: Functions are decomposed into small, serverless units that can operate in parallel and be dynamically allocated across distributed resources [URB+21]. Challenges include integrating these functions across heterogeneous resources and ensuring optimal allocation and scheduling to meet performance goals.

4. Distributed Resources: The resource layer spans a continuum from cloud data centers to edge and far-edge devices. Each tier of infrastructure varies in properties like computing power and network topology. Managing this distributed environment requires abstraction and flexible management tools to handle the complexity of different resource types and administrative domains.

The proposed data-centric SBA for Edge-Native 6G Networks has several architectural impacts and benefits, which are presented below:

1. Efficient Core Network Signalling: The proposed SBA's data-centric and multicast features, along with atomic and composable functions, enhance signalling efficiency by reducing message overhead, speeding up procedures, and enabling parallel signalling. It also eliminates service indirection and reduces network overhead by bundling related functionalities.

2. Streamline Refactoring: The SBA’s location-transparency and data-oriented primitives enable more efficient software updates, supporting rapid innovation cycles. This reduces the Total Cost of Ownership (TCO) by enabling quicker improvements and faster adjustments.

3. Serverless, Stateless, Reusable, and Shareable Functions: Functions are designed as stateless, serverless, and reusable components that can be dynamically replicated across distributed, volatile resources. This flexibility supports sharing across multiple stakeholders, enabling new opportunities for concepts like PNI-NPN.

4. E2E Orchestration over the Continuum: The Service Dataflow introduces unified orchestration for 6G networks, managing heterogeneous software modules and offering granular services. It also provides interfaces for connecting multiple administrative domains, enabling seamless integration from private networks at the Far-Edge to the Cloud.

5. Security and Fault-Tolerance: A multi-layered approach addresses security and fault tolerance at different architectural levels. The Service Dataflow layer handles content-level security, the Data-Centric Service Routing layer secures communication links, and the granular function layer ensures application-specific security, contributing to a comprehensive security model.

A proof-of-concept was implemented to validate the proposed architecture, focusing on its data-centric principles: Data-centric Service Routing and Service Dataflow. Using 5G and SBA as benchmarks for future communication systems, the proof-of-concept incorporated selected 5G workflows representing diverse communication models for meaningful comparison (see Figure 5-14).

Zenoh and Zenoh-Flow [GCF+22] were chosen to implement the Data-centric Service Routing and Service Dataflow layers of the proposed architecture due to their unique capabilities. Zenoh provides decentralized pub/sub/query functionalities with low overhead, dynamic node discovery, and high efficiency, while Zenoh-Flow enables seamless, high-performance dataflow programming across the continuum with features like declarative graph definition and automatic deployment. Open5GS (5G Core) was deployed alongside emulated gNBs and UEs to gather traces of 5G workflows, which were replayed to compare the proposed approach with current solutions (e.g., HTTP, gRPC, Istio, MQTT, and Kafka). Validation was conducted on a high-performance testbed with 100GbE connectivity.

Two 5G workflows were evaluated: (i) PDU Session Establishment [29.502], representing a Request/Reply pattern, and (ii) Charging Policy Updates [29.512], representing a Subscribe/Notify pattern. These workflows were selected to showcase different NF communication models.

For PDU Session Establishment, the proposed solution was compared with HTTP (3GPP’s default protocol), gRPC (a popular SBA protocol), and Istio (a service-mesh implementation). Results demonstrated that the proposed solution is significantly faster—by one or more orders of magnitude—than all alternatives, regardless of whether service discovery was performed or not.

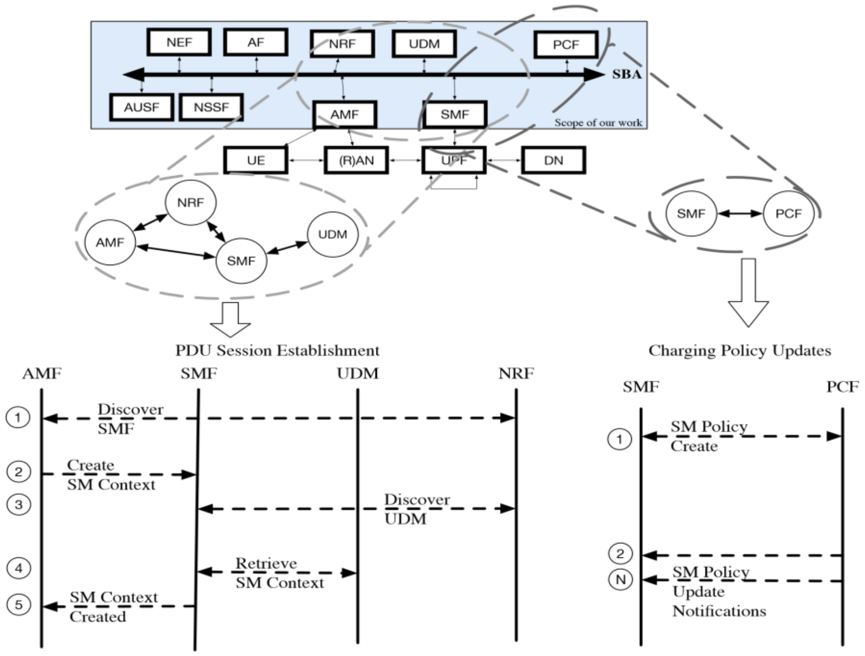


Figure 5-14: 5G SBA architecture, highlighting the workflows used for evaluation.

HTTP and gRPC: Delivered similar performance due to efficient data exchange, faster serialization, and lower transmission overhead. Istio: Showed higher latency due to its reliance on proxy chains to achieve location transparency. Zenoh: Reduced “wire overhead” by mapping topic names (subscription groups) to integers, unlike the other protocols. These results emphasize the performance benefits of the proposed solution for time-sensitive 5G workflows.

The Charging Policy Updates workflow, which uses a Subscribe/Notify pattern (that may be implemented through publish/subscribe protocols), was evaluated by comparing the proposed solution with MQTT and Kafka, two widely used publish/subscribe protocols. Results (Figure 5-15, right subplot) reveal that the proposed solution outperforms both MQTT and Kafka in terms of the time required to push notifications to subscribed entities. The performance advantage stems from Zenoh’s peer-to-peer communication model, which enables direct communication between consuming and producing NFs, avoiding the extra hop introduced by the brokered communication models of MQTT and Kafka. Additionally, Zenoh achieves lower wire overhead compared to MQTT and Kafka, further contributing to its efficiency (Table 5-3).

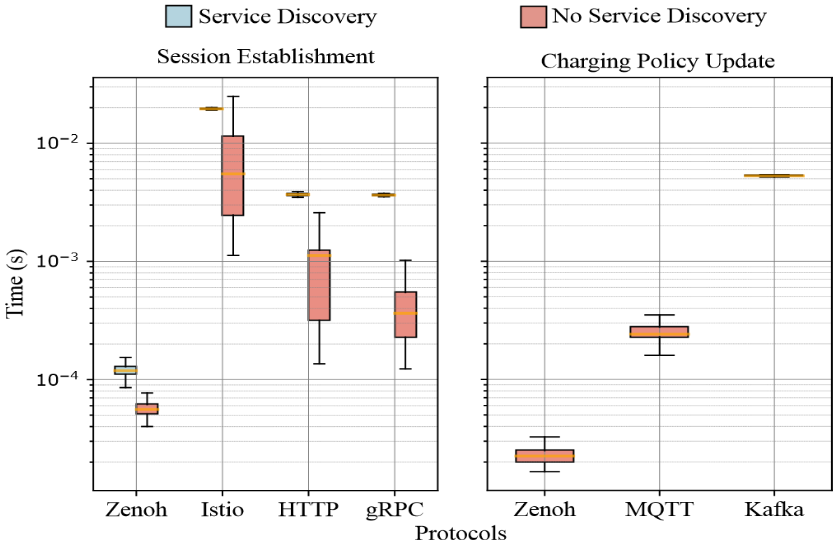


Figure 5-15: Workflow completion time for different protocols (lower is better).

Table 5-3: Analysis of wire overhead, and lines of code, for the different protocols.

	PDU Session Establishment				Charging Policy Update		
	Proposed Solution	HTTP	Istio	gRPC	Proposed Solution	Kafka	MQTT
Overhead (bytes)	365	738	1130	963	11	128	36
Overhead (%)	8.9	18.0	21.1	63.2	20	232	65
CLOC Count lines of code	408	1664	1960	548	114	198	93

Table 5-4 highlights the features of host-centric (e.g., HTTP, gRPC), service-centric (e.g., Istio), and data-centric (e.g., Zenoh, Kafka, MQTT) protocols. Host-centric protocols struggle in decentralized environments, while service-centric ones face challenges in multi-cluster setups. Data-centric protocols, like the proposed solution, abstract data location and optimize decentralized communication using publish/subscribe and request/response models.

The dataflow programming model simplifies Service Function development by breaking applications into atomic components, reducing coding effort and enabling declarative definitions. This approach streamlines core network function design, making interfaces simpler and more efficient for future systems.

Table 5-4: Qualitative analysis of the different protocols.

	Proposed solution	HTTP	gRPC	Istio	Kafka	MQTT
Paradigm	Data-centric	Host-centric	Host-centric	Service-Centric (Intra-Cluster) or Host-centric (Inter-Cluster)	Data-centric	Data-centric
Topology	Peer-to-Peer, Brokered, Routed	Peer-to-Peer	Peer-to-peer	Routed	Brokered	Brokered
Communication Model	Query (Request/Reply) Publish/Subscribe (Push / Pull)	Request/Reply	Request / Reply	Request / Reply	Publish/Subscribe (Pull)	Publish/Subscribe (Push)
Types Aware	Yes	Blob	Yes	Blob	Blob	Blob
Composability	Declarative Application Definition	No	No	Declarative Service Definition	No	No

This work proposes using data-centric and dataflow mechanisms to enhance 5G's service routing and discovery, aiming for a flexible, distributed, Edge-native 6G architecture. The solution, implemented with Zenoh and Zenoh-Flow, was evaluated with 5G workflows, showing significant improvements: faster procedure completion (10-100x), reduced wire overhead (2-11x smaller), and fewer lines of code (up to 4x less). These results suggest increased flexibility, efficiency, and scalability for future 6G networks. Future work will validate the approach in testbeds like 5TONIC and 5GAIner and explore its application in 3GPP's Edge Applications Service Discovery Function (EASDF) and service-mesh.

5.2.2.5 Summary

Table 5-5 shows the summary and key take-aways for the “Interaction between entities” enabler.

Table 5-5 Summary of the “Interaction between entities” enabler.

Description	**RAN-CN control plane interactions and interfaces** There is a need to improve the cloud friendliness of the interface between RAN and the CN (NGAP/N2 interface). For this to happen, we need to evolve or replace the SCTP protocol used in 5G to support better decoupling between the different layers and avoid so called transport bindings, etc. **Data-Centric Service-Based Architecture for Edge-Native 6G Network** Transition to a fully distributed 6G system with Data-Centric Networking, enabling enhanced scalability and flexibility through dynamic stateless NFs and simplified architecture with efficient resource management.
Key take-aways	For the 6G architecture, the analysis shows the main option to be that the RAN-CN interface goes via the AMF, or a 6G equivalent of it, and a major reason for this is to inherently support 5G to 6G migration and for security reasons. QUIC may be one option to use instead of SCTP for a more cloud friendly N2. An evolved N2 with AMF is also seen as the most reliable solution for supporting the new 6G services, mainly for the same reason (security and migration). Data Centric SBA is validated through proof-of-concept prototype using 5G workflows and shows significant architectural benefits in terms of automation and service composability. Procedure completion times (between 10 to 100 times faster) and wire overhead is between two and two-and-a-half times smaller, for PDU session establishment, and between three-and-a-half and 11 times smaller in charging policy update.
Standard relations & regulations	The network composition changes as well as the relevant interfaces and interactions would mainly impact [23.501], [23.502], [38.413].

5.2.3 Modularisation examples

5.2.3.1 Introduction

Modularisation allows for streamlining network modules and functions according to the respective locations and the respective KPIs/KVIs. This section demonstrates how network modules are designed to achieve flexibility and a better adaptability to the needs of the network. Concretely, examples of modularisation from two network domains are described: UPF and RAN.

5.2.3.2 Modular UPF

Although the control plane of 5G Core networks is designed to adhere to the SBA, the user plane, specifically the UPF, is managed as a single large monolithic entity with numerous functions and capabilities to provide support (around 22 in accordance with 3GPP TS 23.501 [23.501] Release 18). Large, monolithic function design can impede the flexibility to adjust resources for subfunctions at a detailed level, leading to slower deployment and debugging processes, as well as limiting the ability to use varied technologies for different design components [AAE16]. For instance, the UPF might manage an uneven amount of uplink/downlink traffic that needs to be dealt with at the Core Network. In this scenario, a modular UPF design would allow for adjusting processing resources for each traffic direction separately depending on the need for scalability. Another scenario occurs when the UPF is anticipated to execute certain functions, like Lawful Intercept, across various locations for a set duration. In this scenario, the module handling Lawful Intercept functions could have its own processing resources separate from the UPF to handle the demand effectively, resulting in more efficient and sustainable utilization of processing resources. In order to achieve this goal, we suggest a modular design for the UPF that could be realized at implementation level in the upcoming Core Networks as shown in Figure 5-16. The entities suggested are the following [Har22]:

- **Ingress Steering Module (ISM):** This module manages the arrival of incoming packets at the UPF. Initially, it verifies the eligibility of the incoming packets according to specified admission control rules in order to discard the ineligible packets. Next, it directs the packets towards either the uplink or downlink modules. This module is capable of implementing load balancing strategies to equalize the processing workload in scenarios where there are multiple replicas of the uplink function or downlink function.
- **Downlink Module (DLM):** It manages the packets from the Data Network to the gNB and then to the UE. It carries out all the packet processing needed to complete the fundamental tasks given to the UPF for managing downlink data traffic.
- **Uplink Module (ULM):** It manages the packets sent from the UE to the Data Network through the gNB. It carries out the necessary packet processing for completing the fundamental duties assigned to the UPF in managing uplink data traffic.
- **On-Demand Module (ODM):** This module list consists of optional UPF functionalities that can be activated as needed, separate from the basic packet processing, such as Lawful Interception, ATSSS, etc. These ODMs can be flexibly activated/deactivated on demand and can also be scaled in/out over time as needed.

The connection between modular UPFs (mUPFs) is based on Service Function Chaining (SFC) principles where certain metadata can be exchanged between mUPFs to fulfill the functional requirements of the created service.

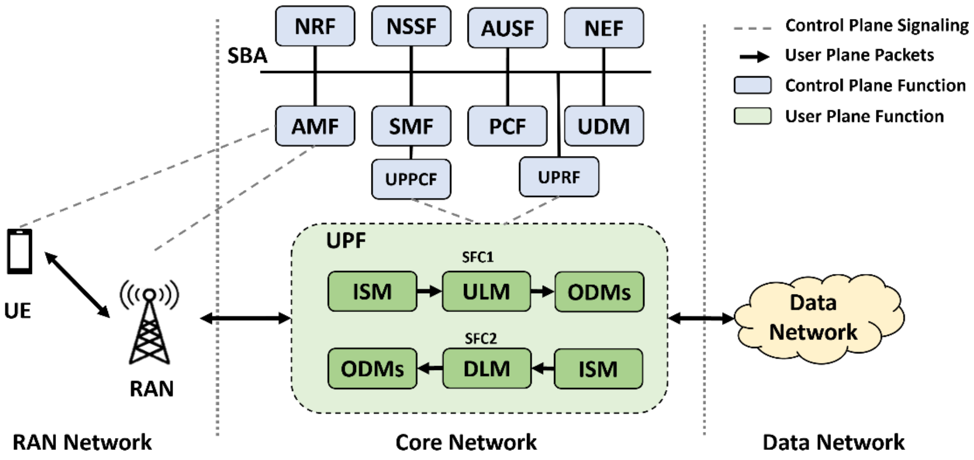


Figure 5-16. Modular UPF design integrated in the E2E mobile network system.

On the control plane side, each mUPF needs to register itself with a profile to a User Plane Registry Function (UPRF). This function can be an independent entity or the 5GC NRF. In addition to the profile's Information Elements sent by the UPF to NRF for registration in 5GS (as specified in TS 29.510 [29.510]) the Registration Request should also include the following information in the case of registering mUPFs:

- mUPF instance identifier. This could be the URI of the mUPF instance which is unique universally in the PLMN. It is used to identify and discover the mUPFs.
- A pre-defined mUPF type is provided from which the mUPF capabilities are derived. These derived capabilities tell about the processing the mUPF is capable of doing. For example, this could be any mUPF type listed before.
- A performance level indicator that can take concrete levels such as high, moderate, and low. It maps to the infrastructure on which the mUPF instance is executed as it relates to the performance of the mUPF. It depends on the processing platform where the mUPF is running, i.e., as a software instance

on commodity off-the-shelf server or on a Hardware Accelerator, or in a middlebox). This information is useful given the heterogeneity of deployments in the cloud as it helps in provisioning distinct performance levels for different network slices.

- The network domain to which the mUPF belongs to. Based on this information the control plane can identify which mUPFs belong to the same domain and thus have physical connectivity to each other and can be chained.
- Unique address identifier(s) for the mUPF instance where the mUPF can receive control and/or user plane information.

A consumer NF sends a “GET” request to UPRF specifying the mUPF URI (which could be the mUPF type name). In response, UPRF sends the mUPF profiles that match the requested mUPF URI. If the consuming NF is not authorized or there is no matching response to the request, an error message is sent back from UPRF with the problem details.

Furthermore, a User Plane Path Control Function (UPPCF) is introduced as an intermediary function between SMF and the mUPFs. This option could be adopted to minimize the required changes needed in SMF when the 5GS supports the modular design of UPFs. Alternatively, UPPCF can be integrated within SMF as a deployment option. In the PDU Session Establishment Procedure, defined in TS 23.502 [23.502], the UPPCF receives the N4 Session Establishment/Modification Request with the requested optional capabilities (e.g., Lawful Interception) and performance level from SMF. Then, UPPCF takes care of selecting the mUPFs that should be involved in creating the UP path for the PDU session based on the requested optional capabilities and performance level. UPPCF interacts with UPRF to discover the registered mUPFs and to select the ones needed for establishing the requested UP path. For example, it will select ISM, ULM, and Lawful Intercept ODM that run as software instances when requested to set a UL traffic path with lawful interception functionality with low-performance level.

In conclusion, modularizing UPF brings a lot of benefits such as scaling up/down processing resources for uplink/downlink traffic independently (especially useful when asymmetric traffic is expected), on-demand activation/deactivation of optional functionalities in UPF, and more flexible placement of modular functions. However, the penalty in terms of the management complexity of the modularised functions as well as the extended execution delay because of the intercommunication between mUPFs should not be neglected. In the end, this modular design adheres to the same principles discussed earlier on the trade-off between flexibility and performance when deciding on the level of granularity to be adopted when modularising any NF.

5.2.3.3 Disaggregated RAN to enable cell-free massive MIMO

Cell-free massive MIMO is a physical layer technology intended to solve one of the main limiting factors of cellular networks. That is, improve the performance of cell-edge users, thus providing reliable service ubiquitously in the network.

A disaggregated RAN, in which the functionalities are separated into different entities, is essential for the practical implementation of cell-free massive MIMO networks. Joint processing of signals requires a certain level of centralization between radio units. Cell-free massive MIMO benefits from the flexible and scalable solutions brought by RAN disaggregation. A suitable functional split between entities, enabling the required levels of coordination, is required to make cell-free a practical solution. The proposed choice for cell-free massive MIMO in a disaggregated RAN involves two entities, namely RUs and DUs. The physical layer functionalities are split between the RU and the DU. In particular, as defined by 3GPP, the RU-DU split considered corresponds to option 7-2 [38.801].

The canonical form of cell-free massive MIMO, where all users are served from all RUs [NAY+17], is unscalable and impractical, due to the enormous computation and coordination complexity that it requires [BS20]. These limitations can be overcome with user-centric approaches, consisting of creating overlapping serving sets of RUs to serve each user [BD17].

Aiming at achieving user-centric operation, a framework consisting of multiple orthogonal partitions of the system bandwidth is suitable for improving the situation of the cell-edge users. This is done by making DUs manage multiple RU clusters, each in a different partition [GAT24]. This framework is depicted in Figure 5-17 [GAT24b]. The cellular D-MIMO clusters are shown in two different colors. To solve the poor performance of cell-edge users in such a scenario, new RU clusters are created, after extending the fronthaul network by connecting the RUs near the DU borders to multiple DUs. In this example, the RUs in the gray area are connected to both shown DUs. These RUs divide their resources between the two DUs. In turn, the DUs now manage multiple clusters, each in a different bandwidth partition. Even though the two DUs in the figure share the gray cluster, they do it in different partitions, creating a structured and regular cluster structure in which, ultimately, the whole bandwidth is available at all RUs, but a number of them divide these resources to multiple DUs. The weights w_i denote the fraction of bandwidth allocated to each cluster, where w_i is orthogonal to $w_j,$ for $i \neq j$.

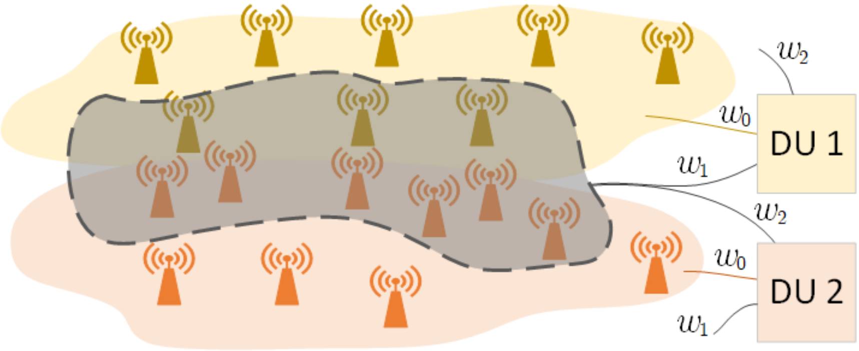


Figure 5-17 Disaggregated RAN architecture to serve users in a user-centric manner [GAT24b].

After the extension of the fronthaul network to accommodate new clusters in which to serve users, a distributed optimization finds the optimal scheduling of users [GAT24]. Ideally, each user will be served, at least, by the cluster for which it can obtain better rates, provided that such cluster has available resources. Due to the increase of inter-cell interference as one moves towards the cluster edge, an efficient solution means that users have at least one cluster whose border is relatively far away. In other words, guaranteeing a minimum rate for each user is equivalent to defining a minimum distance between users and the edges of their serving clusters, or, equivalently, a maximum distance between users and the centers of their serving clusters.

The problem can therefore be formulated as [GAT24b]:

Create new clusters so that each user has at least one cluster center at a maximum distance of r, while:

1. Minimizing the number of clusters needed (to reduce computational complexity and coordination overhead in the backhaul when optimizing the partition bandwidth allocations).
2. After 1., minimize the number of new fronthaul connections.

The clusters are approximated as having circular shape, with a radius $R<r$. Before the creation of the new clusters, the network is composed of RU clusters, each of them composed of two parts: a circle of radius r, where the users inside already satisfy the distance requirement, and an annulus of inner radius r and outer radius R, where users inside do not satisfy the requirement. Upon creating the new clusters, each of the new clusters will follow the same pattern: an area of good performance and an area of poor performance, starting at a distance r from the center. This is shown in Figure 5-18.

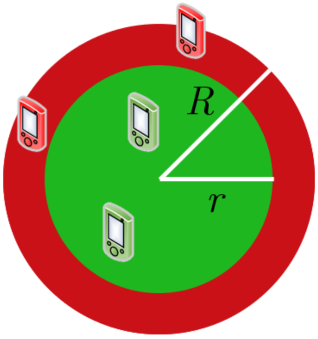


Figure 5-18 Good and bad performance areas of a DU area, in green and red, respectively.

The problem, thus, reduces to finding the minimum number of clusters that fully cover, with their good performance areas, the annulus of the initial cluster. A further assumption is made, which considers that the new clusters are all shifted the same distance s from the original clusters. A geometry problem is formulated, the details of which are in Section A.9 of this document.

The performance of the clustering solution is evaluated, considering a deployment in which the RUs are distributed at random in the network area. The downlink ergodic rates are computed before and after the extra clusters are created. Users are connected to the DU containing the RU with minimum path-loss, based on the initial clusters. After the creation of the new clusters with the method described, users are served from the cluster, managed by the same DU, that provides them the highest ergodic rate. The cluster radius R is selected as the maximum distance from a cluster centroid to its furthest RU. The value of r is approximately selected as a measure of how large cluster borders should be considered. The new clusters are formed by moving the original RU cluster centroids a distance $s_{\min}$ from their original positions, in directions equally spaced angularly.

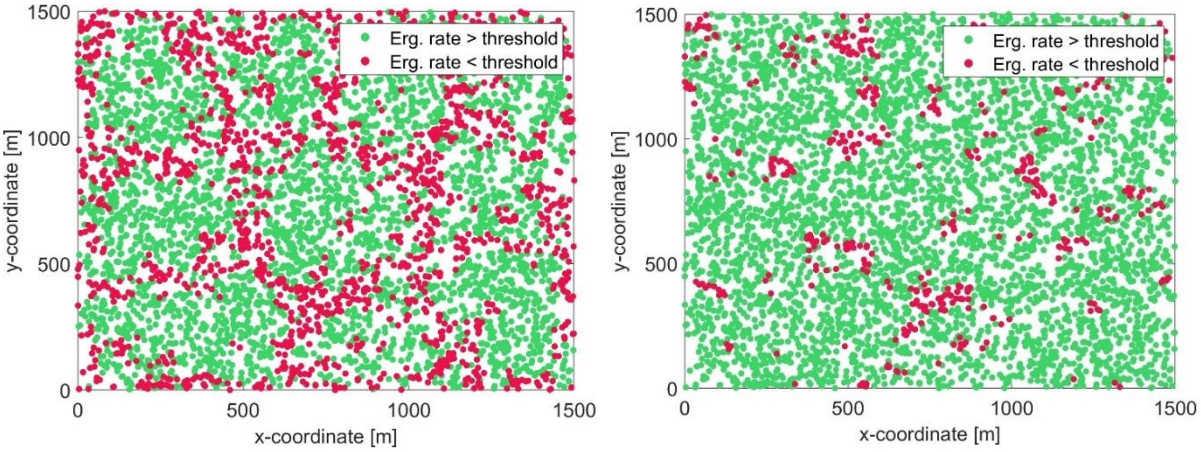


Figure 5-19 Users below and above a certain ergodic rate threshold, before the creation of new clusters (left) and after (right) [GAT24b].

Considering a deployment in an area of 1.5 x 1.5 km, with 5000 single-antenna users and 900 4-antenna RUs, initially divided into 10 clusters (DUs), Figure 5-19 shows the downlink ergodic rates before and after the new clusters are created, with respect to a certain threshold [GAT24b]. The cluster borders are clearly visible in the left figure, where users near borders have lower ergodic rates. With the same ergodic rate threshold, after creating new clusters with the described method and serving users accordingly, the number of users that are still below the threshold is significantly reduced. As creating new clusters is intended to solve the poor performance of the edge users, they are the ones that experience a greater benefit. Furthermore, due to the way users are served, no user sees a degraded rate compared to the single clustering network.

Figure 5-20 shows the CDF of the user ergodic rates in the two cases. Focusing on the 5th or 10th percentile, that is where the users are “suffering”, the ergodic rate gain is significant.

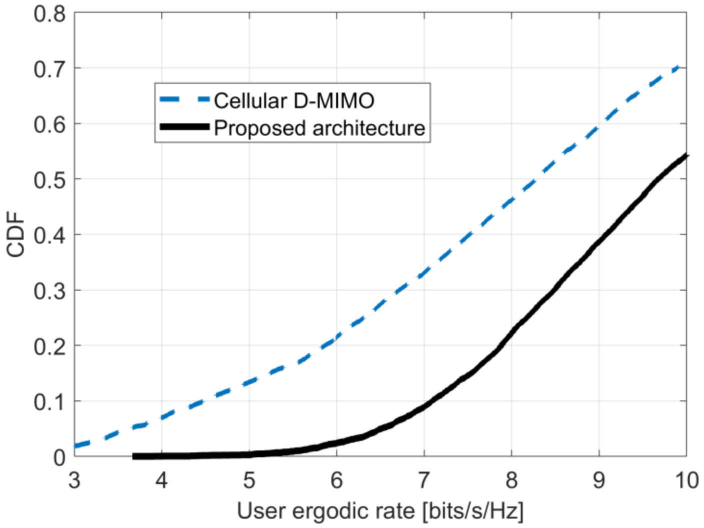


Figure 5-20 Empirical CDFs of the ergodic rates of users in both cases studied.

5.2.3.4 Summary of the modularization examples

Table 5-6 shows a summary of the key take-aways for the modularization examples.

Table 5-6 Summary of the modularization examples

Description	Modularization gives the advantage of streamlining network modules and functions according to the deployment locations and the respective KPIs/KVIs. In this enabler, the focus is to demonstrate how the new modules can be designed at the different network domains (e.g., RAN or UP).
Key take-aways	UPF is designed in 5G as one monolithic function with multiple functions and features (i.e., approximately 20 [23.501]) hindering the scalability of different sub-functions. Modularized UPF aim at enabling flexible scaling and activation of functionalities on demand resulting in efficient resource utilization and energy efficiency. Different RAN architectures have an impact on the performance for D-MIMO. The cell-free architecture provides better user rates compared to cellular networks. According to simulations, users in the fifth percentile may see ergodic rate improvements of more than 75%. Modularisation offers a higher adaptability to the needs of the network, as it scales the different functions as needed. (E.g., modular UPF better handles asymmetric uplink-downlink traffic, and modular RAN allows for flexible functional splits in the RAN stack).
Standard relations & regulations	The modularisation examples at this enabler impact: Modular UPF: TS 23.501 and TS 23.502. Disaggregated RAN: TS 38.401 and O-RAN.

5.3 Novel cloud functions

5.3.1 Introduction

As 6G networks continue to develop, the need for efficient coordination across multiple domains and cloud environments becomes critical. Section 5.3.2 delves into the concept of multidomain and multi-cloud federation, a pivotal enabler for the highly distributed and interconnected nature of 6G. This federation approach addresses the inherent complexity of managing resources and services across diverse domains, spanning core networks, edge infrastructures, and extreme-edge environments.

The multi-cloud paradigm is central to 6G's ability to deliver seamless and scalable services. By federating cloud resources from various providers, the network gains flexibility and resilience, ensuring that workloads are dynamically allocated based on performance requirements, resource availability, and cost optimization. This approach enables the integration of edge-native applications and distributed storage solutions, enhancing support for latency-sensitive and data-intensive services. Federated cloud systems also play a crucial role in extending the capabilities of edge and far-edge nodes, allowing efficient workload offloading and ensuring seamless application lifecycle management across a unified ecosystem.

However, achieving this level of interoperability poses significant challenges. Differences in cloud architectures, APIs, service-level agreements (SLAs), and security protocols create barriers to integration. Multi-cloud federation addresses these issues by establishing common frameworks and protocols that enable collaboration between cloud service providers and on-premises infrastructure. This includes developing shared data models, harmonizing security standards, and adopting decentralized orchestration techniques to facilitate resource and service coordination across disparate domains.

Incorporating these principles, the multi-domain federation model in 6G enables the seamless integration of edge, core, and cloud resources, ensuring optimal resource utilization and service delivery. This collaborative framework fosters innovation, promotes operational efficiency, and lays the foundation for a more sustainable and flexible network ecosystem. By embracing these advancements, 6G networks are well-positioned to address the demands of emerging applications and ensure a future-proof infrastructure for diverse stakeholders.

5.3.2 Multidomain – Multi-cloud federation

5.3.2.1 Multi-domain Federation at Data centres

While IoT allows for the collection of data at a much wider radius along the edge and extreme-edge, the overall volume of data stored and processed by this increase in coverage also grows. In order to support 6G applications and services, service providers must find a solution towards this growing demand of data processing and data storage requirements, beyond what physical servers and IoT devices can supply. This increase in network coverage also raises problems regarding the sustainability of the 6G network, as a higher number of devices have associated deployment and maintenance costs. One way to mitigate this issue is to offload these computational and storage requirements towards the cloud; this paradigm shift away from physical resources is associated with the growing adoption of Cloud Computing technologies.

According to IBM, Cloud Computing encompasses the "on-demand access of computing resources—physical servers or virtual servers, data storage, networking capabilities, application development tools, software, AI-powered analytic tools and more—over the internet with pay-per-use pricing"[4]. Companies like Microsoft, Amazon and Google dominate the market share with over 60% of total usage of cloud infrastructure services[5] with their Azure, AWS and GCP solutions respectively. Demand for Cloud Computing is also rising applications and services with heavy data collection and processing requirements are moved more and more towards edge and cloud servers, with cloud servers taking the bigger slice as the solution to adopt towards this trend.[6]

One of the services that Cloud Computing can offer is Federated Learning. As explored in previous deliverables, Federated Learning is a decentralized approach to model training with privacy-preserving features. Usual Machine Learning (ML) model training involves using an ML algorithm on training data that is properly processed for the algorithm to learn from it and produce a model that is able to predict some information from new data of the same type. This can be used in several ways, such as image recognition, survey analysis, medical prediction, and many others. Federated Learning expands this capability in a decentralized setting: an ML model is chosen and uploaded (usually by a central server) towards nodes across an edge network; this model will then use data collected on those edge nodes for local training; once the training is finished, the model results will be uploaded back to the central server for aggregation, creating a global model of all of the edge nodes. A benefit from training the model on local edge node data is that only model results are uploaded to the central server for aggregation; this means that data collected from the nodes will not be accessed, thus ensuring the privacy of said data.

[4] https://www.ibm.com/topics/cloud-computing, 2024

[5] https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/, 2024

[6] https://data.gsmaintelligence.com/api-web/v2/research-file-download?id=79791138&file=260224-Global-Mobile-Trends-2024.pdf, 2024

In a future paradigm where Federated Learning is widely used by applications to handle model training on edge nodes, Multi-cloud Federation is a way to establish interoperability between different cloud domains coexisting in the 6G network. Access to different cloud domains is also a way to minimize infrastructure costs while managing edge resources and applications: by merging cloud and edge technologies, we can have distributed storage capabilities on our network, supporting Edge Native Applications while also having the processing flexibility of Cloud Computing across multiple vendors [HEX223-D32].

By pooling cloud vendor resources with our federated edge and extreme-edge capabilities, the 6G network becomes more flexible when dealing with workloads: not only does it increase the reliability of running workloads on the uncertain distributed nature of nodes, but it also allows the offload of these workloads towards reliable cloud environments, handling the computation and storage requirements that the physical network cannot supply. This flexibility also increases the resource efficiency across the network as a whole. Services like Lifecycle Management (LCM) and Workload handover are supported by cloud vendors, which provide important support to the applications running on the network.

The coordination between different Cloud Service Providers (CSPs) does not come without its sets of challenges: differences between architectures, Service Level Agreements (SLAs), API's, abstractions and storage structures influence how data is fed from the customer towards the vendor, with different formats, resources and topologies have to be interpreted; security also poses a challenge, as different CSPs have different security protocols, SLAs and rules, which could expose user data towards uncertain security risks. In a sense, interoperability between CSPs is the biggest challenge in the Multi-Cloud paradigm: be it vertical interoperability, along the cloud stack with regards to specific platform standardization; or horizontal interoperability, dealing with the communication between CSPs for cooperation in the Multi-Cloud system [SGS21].

A way towards establishing a bridge between different Cloud providers and local Data centres is the development of a Custom OS for Edge Nodes on the 6G network. Development of this solution would allow for:

- Edge Node Federation: host nodes collaborate on model training and workload splitting, as well as nodes outside of the federation being able to register in with the host nodes, sharing the federation configuration to recent member nodes;
- Node Clustering: the discovery and configuration of local node clusters in the federation, allowing for specificity among certain edge and far edge locations running similar location or data applications;
- Edge Native Apps: this custom OS allows for the integration of edge-native applications, designed to run and communicate most efficiently on edge nodes, supporting their specific hosting and LCM;
- Cloud Offloading: given the distributed nature of the edge nodes of the 6G network, offloading workloads running on the edge towards the cloud can be more efficient, as well as being able to choose from multiple cloud vendors to benefit from the most optimal cloud services and LCM for the current workload;
- Smart Storage Layer: to overcome the challenge of the limited storage capabilities of the edge nodes, a Distributed Storage Layer (Smart Storage) is designed with the purpose of supporting edge native applications.

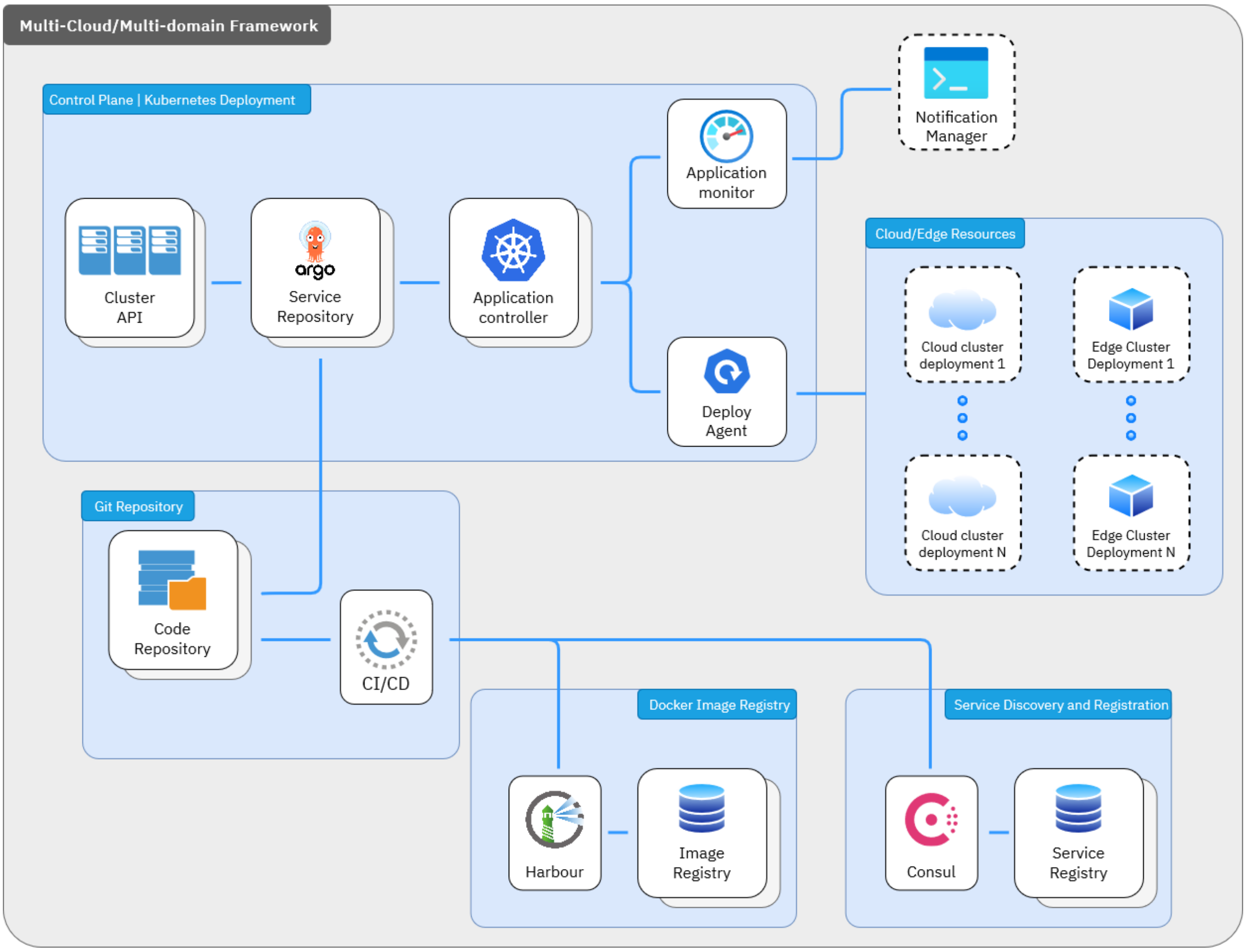


Figure 5-21: Proof of Concept of our Custom OS for Multi-domain Federation on data centres

This Custom OS solution illustrated by Figure 5-21 supports Edge Native Applications on the Edge and Far-Edge. The advent of Network Function Virtualization (NFV) [NFV006] enables the deployment of these applications through software instead of hardware, allowing for Commercial-Off-the-Shelf (COTS) devices and servers to fulfil the purpose of hosting the required services and applications; together with Software Defined Networking (SDN), a network architecture approach driven by APIs, it allows for a software-controller platform to directly communicate with the infrastructure as a service (IaaS), directing the relevant network traffic [ONF24].

In a MEC framework, it is possible to move from centralised servers towards providing services at the edge [MEC003]. These edge servers, although more limited in resources than their centralised counterparts, are physically closer to their clients, enabling lower latency applications. Using NFV and SDN empowers MEC with more flexibility on the way the network is designed. One problem with MEC is the scarcity of edge devices in network scenarios, given that a higher density of edge servers running is highly expensive; this can impact clients, as they can be subject to latency issues when trying to connect on different geographical locations, such as the case of smart city applications. Our hybrid setup complements this current paradigm, being capable of managing the available Edge resources and applications, while offloading the necessary computational effort required to run them towards the Cloud, preserving the distributed nature of MEC.

This PoC is composed of a central control plane that supports the multiple Kubernetes [K8S] control planes managing edge servers. A declarative configuration is exposed from every edge server, processed by a Cluster API, that provides the control panel with tools that simplify provisioning, upgrading, and operating multiple Kubernetes clusters. ArgoCD is a declarative GitOps tool for Kubernetes deployments [Arg24], enabling domain specific service bootstrapping and supporting the Continuous Integration and Continuous Delivery (CI/CD) of the deployed applications and services. Information related to Kubernetes deployments on edge servers and cloud is securely managed and stored through Harbour, an open source registry for Kubernetes images [Har24]. Discovery and registration of new edge devices and cloud resources is managed using Consul, which updates the network topology according to the registration and elimination of available network resources, while preserving the securityof the devices and the network [Con24].

Using the built-in Kubernetes controller, the PoC can automatically assign the necessary resources to deploy the services and applications on either edge or cloud clusters, through a vanilla Kubernetes installation on each of them. The Lifecycle Management (LCM) of the applications and services through CI/CD mechanisms is done by ArgoCD, through the internal exposure of APIs. Cluster performance monitoring information is exposed to the control plane, natively by Kubernetes; a Notification Manager is in place to highlight important statuses and updates on the deployed applications and available/unavailable services/resources.

5.3.2.2 Multi-provider in the cloud continuum concept - Data sourcing for Data Centre Monitoring Engine

Multi-provider-based Cloud Continuum approach was described in [HEX224-D33]. An important element of this approach is the resource layer, which contains resource-oriented functions. Among these functions is DCME (Data Centre Monitoring Engine), which evaluates in real-time each data center status, i.e., energy consumption, performance, and reliability. In order to fulfill the multi-cloud capability requirements for multi-provider environment, the Kubernetes-based cloud native approach was chosen, which enables to organize resources in the form of nodes into Kubernetes Clusters, capable of efficient orchestration and management of distributed systems, such as the CNF-based 5G Core networks. Following the OCI (Open container initiative) standards [OCI24] enables plug and play flexibility of integrating different container storage, networking and runtime solutions. These standards allow us to manage resources despite the heterogeneity of the infrastructure, allowing the Kubernetes to manage the cluster.

This contribution is focused on monitoring resources (CPU, Memory) and energy usage of cloud-native network functions, providing valuable insights into the efficiency of different parts of the core network. These insights can be further analyzed to pinpoint parts of the core network that use disproportionate amounts of energy and improve their efficiency.

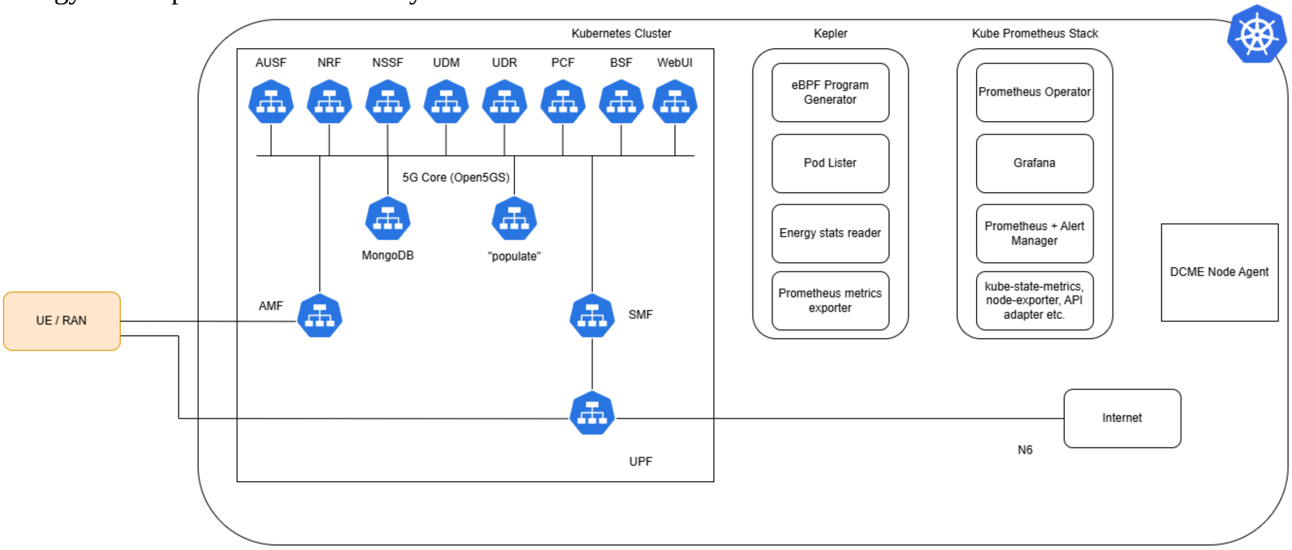


Figure 5-22: Experimental TestBed

The Open5GS Open-Source project was selected as the 5G Core implementation in the Experimental TestBed (Figure 5-22). Additionally, the Kube Prometheus Stack was deployed to collect the data from Kepler tool [Kepler] and monitor the resource usage and "health" of the Network Functions. The solution is capable of supplying information to the DCME thanks to the DCME Node Agent.

For the Experimental TestBed, a Kubernetes Cluster (version 1.31) was created on a server with the following specifications: 128 CPU Cores, 1000GB RAM, Operating system: Ubuntu 22.04.4 LTS.

Kepler was chosen, which utilizes a variety of data sources, such as eBPF programs [EBP24] to provide an estimate of the energy usage (in Joules), which is then exported and stored in Prometheus. Depending on the environment, when possible, the tool can provide power usage statistics of CPU, Memory and GPU usage. Kepler can then calculate the energy consumption per process, which is assigned to a Kubernetes container in a pod. Compatibility with Kubernetes allows for energy usage measurements of different implementations of

5G Core network, making it easier to compare and improve existing projects. As a traffic generator, my5G-RANTester [MY5G] was deployed. It is an open-sourced gNB/UE emulator, configured to allow communication with the 5G Core. The registration and PDU session establishment procedures were conducted for a single user in a loop – as soon as the procedures were completed, the user was de-registered, and the gNB and UE were terminated. After that, the procedures were once again carried out. Kube-prometheus-stack was chosen as the tool to provide CPU and Memory measurements for all parts of the 5G Core Network. These measurements were taken during both the idle state of the Core network (meaning no procedure was being carried out) and during the registration and PDU session procedures.

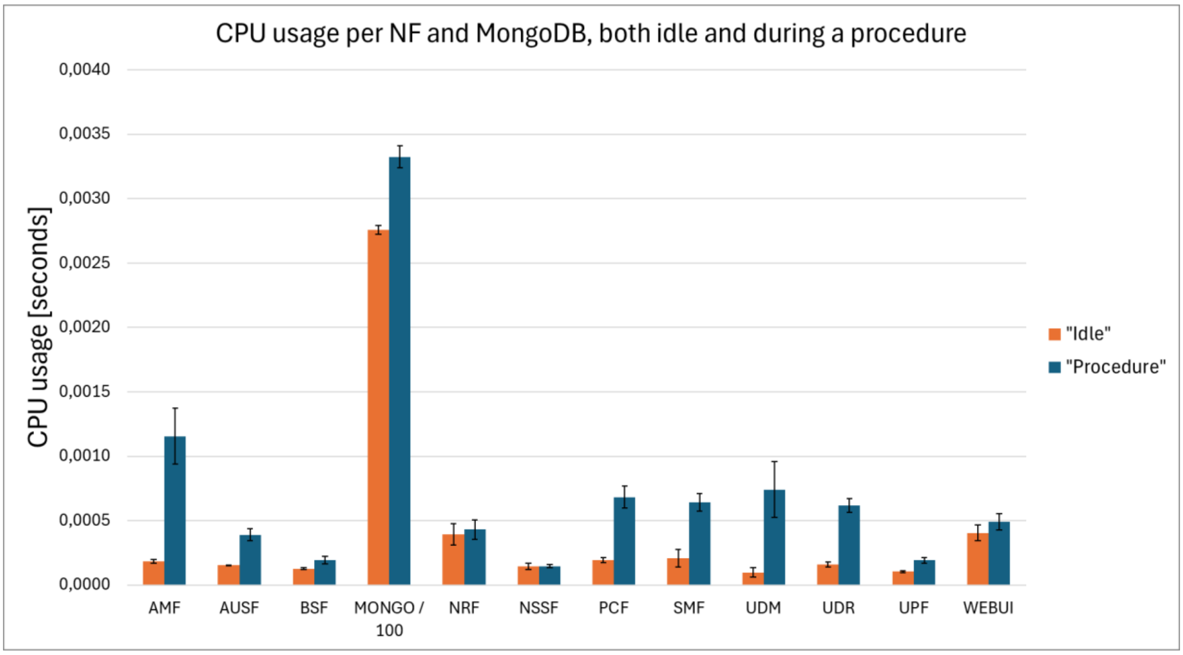


Figure 5-23: CPU usage per NF, in both idle state and during a procedure

The obtained results can be seen in Figure 5-23. In the considered case, the MongoDB database [Mon24] (scaled by 1/100 in the figure) is using a disproportionate amount of CPU resources compared to the 5G NFs. The increase of CPU usage during the procedure can be seen in almost every network function, with the biggest increases occurring for the AMF, AUSF, PCF, SMF and UDM.

Comparing the total usage of CPU resources by all the network functions and the MongoDB database (see Figure 5-24), it can be observed that the cumulative CPU resource usage of all network functions is a fraction of CPU resource that the MongoDB database consumes. The MongoDB database consumed over 58 times as much CPU resources during the procedure as all the network functions combined, and over 126 times as much when the network was in the idle state (no procedures performed).

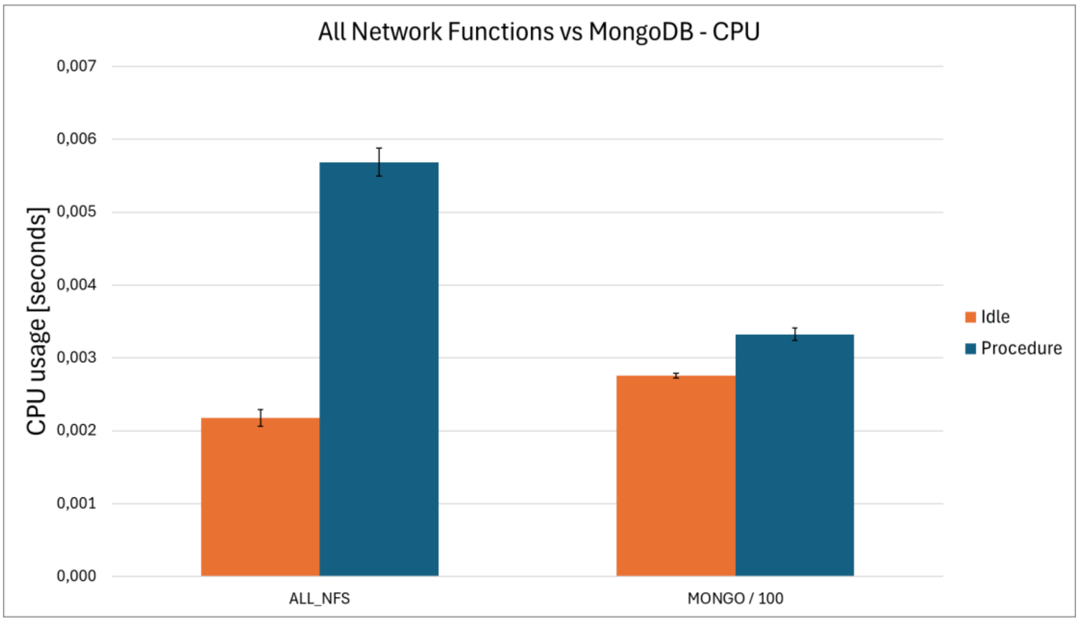
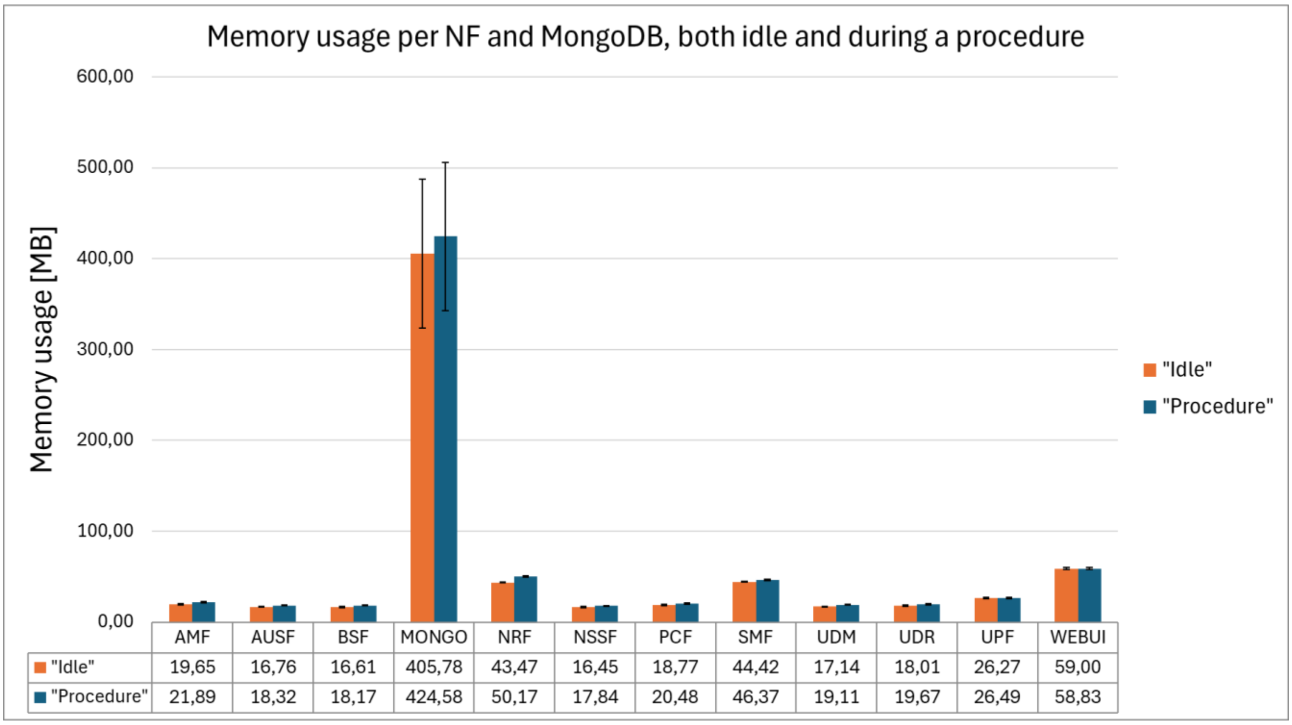


Figure 5-24: Cumulative CPU usage of network functions and MongoDB database, both in idle state and during a procedure

Similar results can be observed in Figure 5-25 where we can see the memory utilization of network functions and the database.



	AMF	AUSF	BSF	MONGO	NRF	NSSF	PCF	SMF	UDM	UDR	UPF	WEBUI
"Idle"	19,65	16,76	16,61	405,78	43,47	16,45	18,77	44,42	17,14	18,01	26,27	59,00
"Procedure"	21,89	18,32	18,17	424,58	50,17	17,84	20,48	46,37	19,11	19,67	26,49	58,83

Figure 5-25: Memory usage per NF and the MongoDB database, both in idle state and during a procedure

When it comes to the memory usage, the cumulative usage by all network functions (which can be seen in Figure 5-26) is surpassed by the MongoDB memory usage by 36.84% in idle state, and by 28.16% when the 5G Core network is carrying out a procedure.

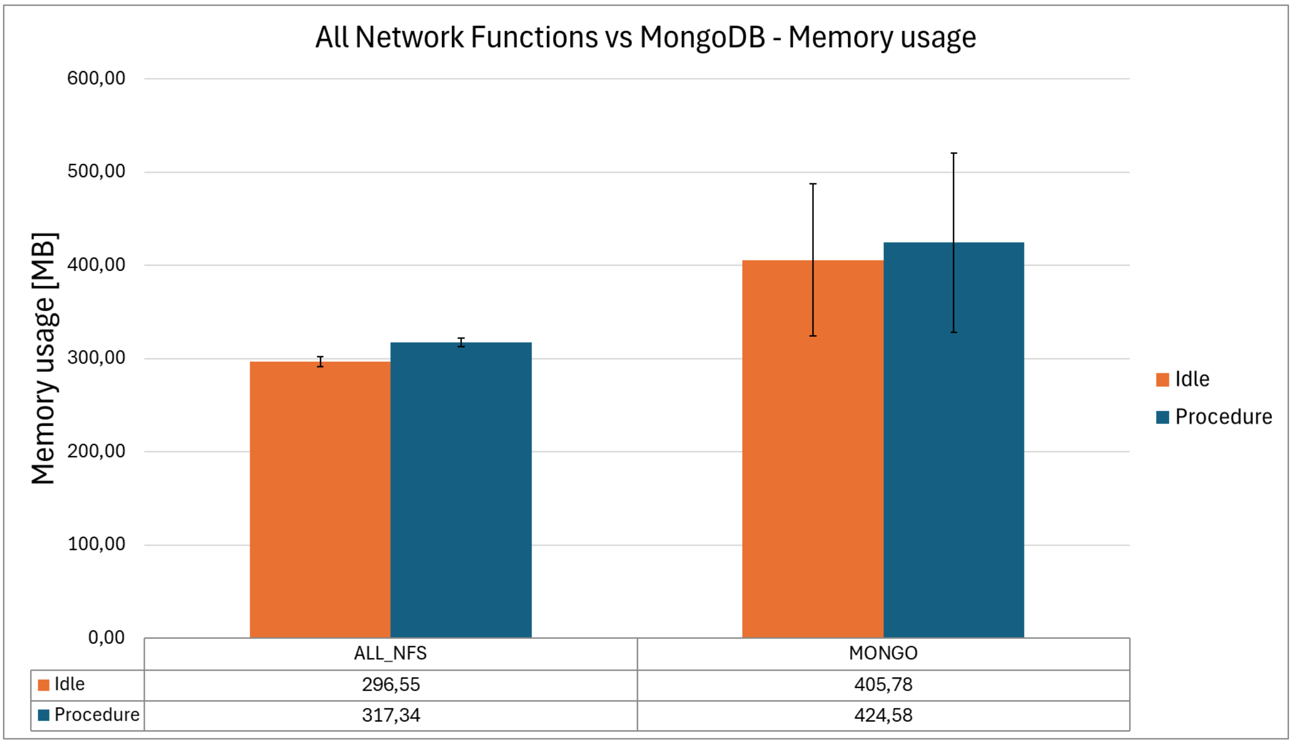


Figure 5-26: Memory usage of network functions and MongoDB database, both in idle state and during a procedure

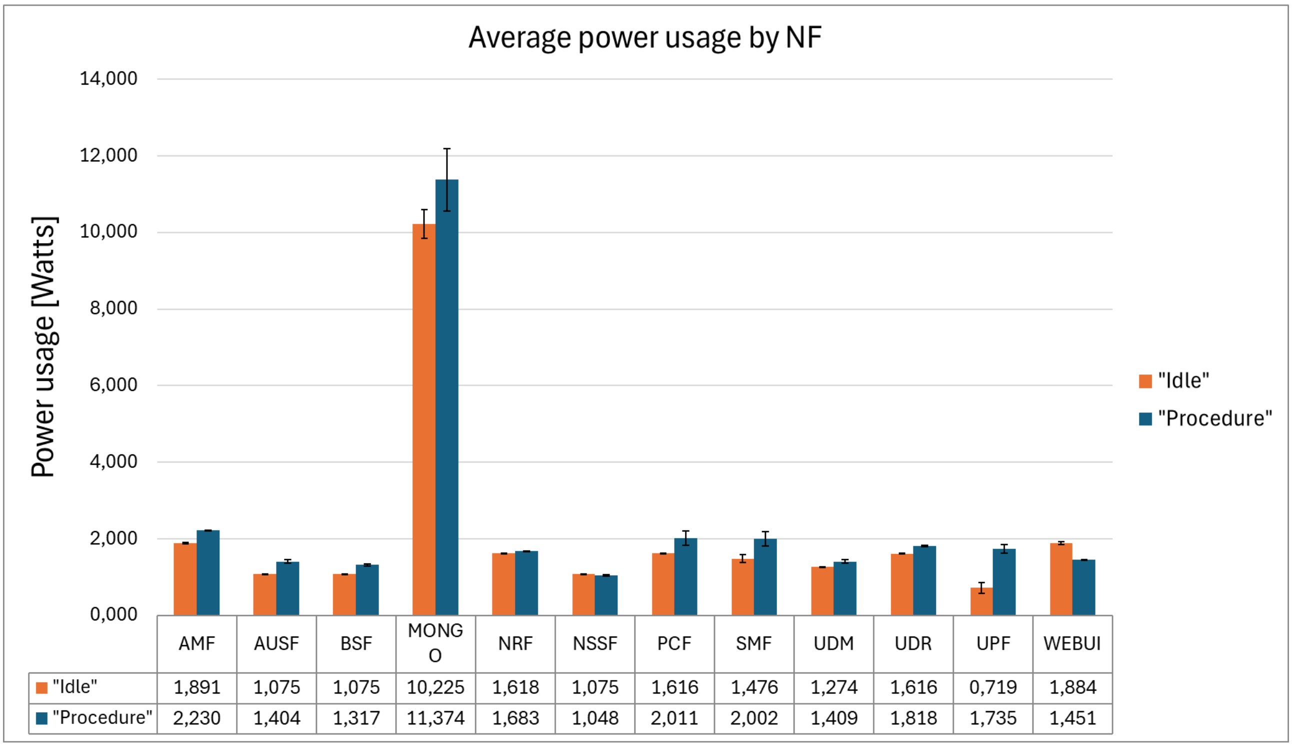
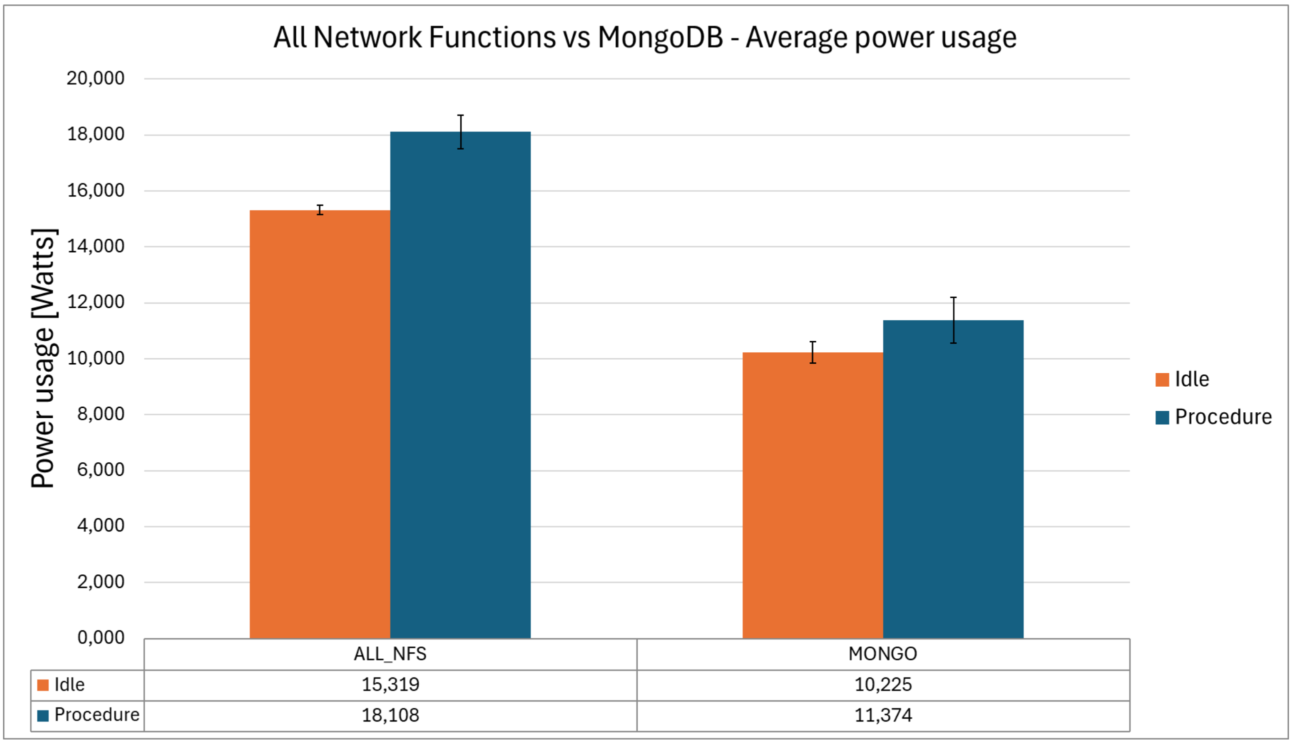


Figure 5-27: Average power usage by network function and the MongoDB database, both in idle state and during a procedure

Power usage, which can be seen in Figure 5-27, differs from the resource usage. Once again, the MongoDB database uses more resources, up to 14 times as much power as the UPF network function uses in the idle state.



	ALL_NFS	MONGO
■ Idle	15,319	10,225
■ Procedure	18,108	11,374

Figure 5-28: Power usage – all network functions vs MongoDB database, both in idle state and during a procedure

This time however, the average power usage of all network functions (see Figure 5-28) surpasses the average power used by the MongoDB database both in idle state, and when carrying out a procedure by 49.8% and 59.28%, respectively.

In conclusion, data regarding resource usage of CPU and Memory shows a staggering difference, with resources used by the MongoDB database surpassing the combined usage of all network functions by a factor of 58 to 126, depending on whether the network is in idle state or not. A difference this big warrants an investigation into alternative databases, ones which do not require as many resources, especially when the network is in the idle state. However, more sophisticated metrics, such as average power usage of individual containers provided by Kepler are more lenient towards the MongoDB database. Power usage still indicates that the database consumes a disproportionate number of resources, surpassing some network functions usage by a factor of 14 in the idle state. Based on all of the aforementioned results, an inquiry into more lightweight databases capable of replacing MongoDB for the 5G Core networks is needed. The same principles and measurements can be carried out for other 5G Core implementations, allowing for efficiency comparison, especially when the compared 5G Core implementations both follow the same 3GPP release.

5.3.2.3 Multi-domain/multi-cloud federation

The 6G era brings an unprecedented level of connectivity and innovation, with applications ranging from real-time holographic communications to massive IoT ecosystems and beyond. These applications place demanding requirements on network infrastructure, including ultra-low latency, high reliability, and seamless scalability. As 6G evolves, so does the need for network operators to efficiently manage and expand infrastructure, particularly across national boundaries, to meet user demands and economic goals. A federation module, leveraging a cloud continuum and peer-to-peer architecture, is essential to this evolution. Such a module would enable network operators to share resources, streamline infrastructure utilization, facilitate service deployment across regions, and support emerging 6G network capabilities.

The transition to 6G and the integration of a federation module expose specific limitations within the 5G architecture. 5G networks were primarily designed with a single-operator model, where each operator managed its own infrastructure. Although 5G allowed for some level of resource sharing, such as network slicing, these capabilities were limited to individual operators' domains and could not be extended seamlessly across different operators.

In this section, it is outlined the structure and components required for a 6G federation module that allows network operators to collaboratively extend their networks and services through a peer-to-peer federation model. This approach fosters cross-operator cooperation, allowing each operator to seamlessly connect and operate with shared infrastructure, respecting local regulatory requirements, while maximizing resource efficiency and service quality for end users.

The federation module aims to achieve the following objectives:

1. **Infrastructure Sharing Across Operators**: Facilitate secure sharing of infrastructure, including network functions, compute, and storage resources among operators.
2. **Cross-Border Service Deployment**: Enable seamless service deployment and roaming across different geographical regions and national boundaries.
3. **Integration of Network Capabilities**: Support essential 6G functionalities, such as local breakout, network slicing, and dynamic resource allocation, within a federated framework.
4. **Decentralized Control**: Adopt a peer-to-peer model where operators maintain autonomy over their resources while participating in a collaborative federation.

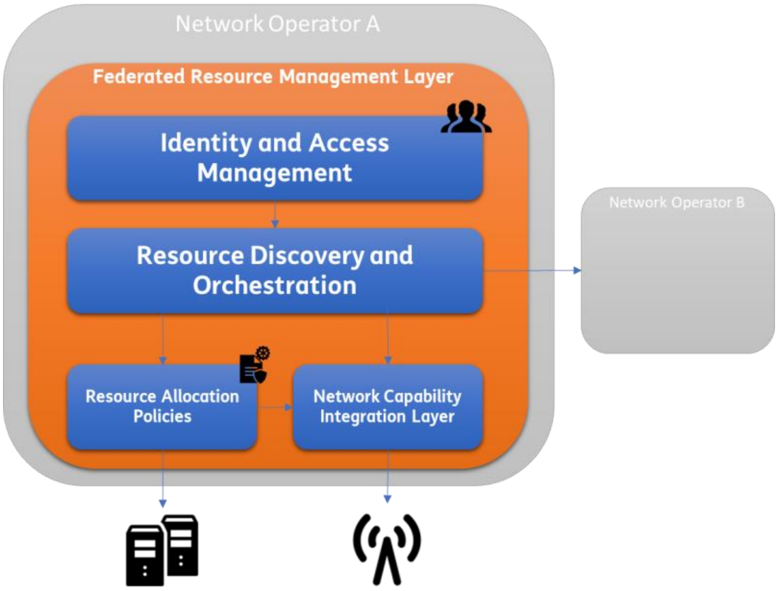

Network Operator A
Federated Resource Management Layer
Identity and Access Management
Resource Discovery and Orchestration
Resource Allocation Policies
Network Capability Integration Layer
Network Operator B

Figure 5-29: Federated Resource Management Layer high level architecture

To achieve the objectives outlined, the federation module requires a well-defined architecture with several interdependent components. Each key component and its role within the federation framework is described below.

The **Federated Resource Management Layer** is the core of the federation module, enabling operators to coordinate and share network resources. As show in Figure 5-29, this layer includes four subcomponents:

- **Resource Discovery and Orchestration**: This subcomponent allows operators to discover available resources from other network operators within the federation. An automated orchestration system manages the allocation and release of resources across networks, enabling efficient use of shared infrastructure.
- **Resource Allocation Policies**: To ensure fair and efficient use of shared resources, the module incorporates customizable policies for allocating network and compute resources. These policies can be dynamically adapted based on traffic demands, regulatory constraints, and business agreements. This module also ensures that all shared resources adhere to predefined SLAs. It monitors resource usage and performance, ensuring compliance with a set of predefined constraints (e.g., latency, bandwidth, and reliability standards agreed upon within the federation).

- **Identity and Access Management (IAM)**: to ensure secure access control within the federated environment. In a peer-to-peer federation, trust and security are paramount to prevent unauthorized access and maintain the integrity of each operator's resources.
- **Network Capability Integration Layer** brings together essential network functions to enable complex 6G services across the federated infrastructure. This layer is especially crucial in enabling high-performance services with specific local network capabilities, such as local breakout and network slicing.

The diagram in Figure 5-30 demonstrates the capabilities of the federation module in a realistic 6G scenario, highlighting how infrastructure and service sharing between different network operators can support seamless application mobility. This capability is critical for enabling devices to access services efficiently as they move across different operator networks.

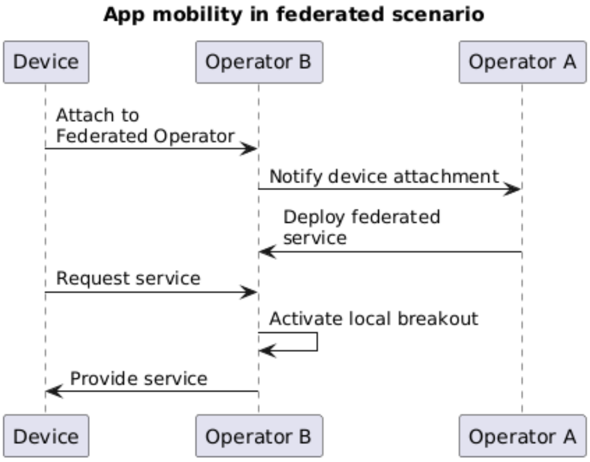


Figure 5-30: Sequence diagram for app mobility

In this example, the federation module coordinates interactions between two operators, Operator A and Operator B, to deliver a continuous service experience, despite the device's attachment to a different network than the one originating the service. The diagram outlines a step-by-step sequence in which operators communicate and deploy necessary services in real-time, demonstrating the essential processes involved in federated 6G service provisioning. This approach ultimately enhances user experience, reduces latency, and optimizes network resources by enabling devices to maintain access to federated services without disruption.

In this scenario, two network operators, specifically "Operator A" and "Operator B," collaborate to facilitate seamless service provision and device mobility across their networks. The steps outlined in the sequence diagram represent the interactions among 3 entities (i.e., the device, Operator B, and Operator A) when a device moves from the infrastructure of an operator to another.

- **Device Attachment to Federated Operator**: The process begins when a user device attaches to "Operator B," a federated operator within the 6G network framework. This attachment signifies the device's initial connection to the network, establishing Operator B as the primary serving operator in this context. This step leverages current network procedures.
- **Federation Notification and Service Deployment**: Upon the device's attachment, Operator B notifies Operator A about this connection event, indicating the device's need to access federated services. This request also specifies that the service needs the local breakout configuration. Operator A, as a federated partner, responds by deploying the requested federated service, ensuring that the device can access resources or applications originating from Operator A's network while connected through Operator B.
- **Service Request and Local Breakout Activation**: The device then initiates a service request to Operator B. This triggers the activation of a "local breakout," a feature that allows Operator B to handle local data traffic directly rather than routing it back through Operator A. This local breakout activation optimizes the data path, reducing latency and enhancing service performance for the end user.

- **Service Provisioning**: Following the local breakout activation, Operator B provides the requested service to the device. At this stage, the device can seamlessly access federated services originally deployed by Operator A, despite being attached to Operator B's network infrastructure. This federated approach to service delivery not only enhances user experience but also maximizes resource utilization across multiple network domains.

This sequence demonstrates how 6G federated networks enable multiple operators to dynamically share infrastructure and resources. By facilitating device mobility and service continuity, federated 6G systems can support advanced applications, improve network efficiency, and provide consistent service quality across operator boundaries. The federation module represents a transformative solution for 6G networks, enabling operators to collaborate while preserving their autonomy, addressing regional regulatory requirements, and expanding services globally. This module is pivotal to meeting the unique demands of 6G applications, bridging the gap between isolated network operators and fostering an interconnected 6G network ecosystem reaching the cloud continuum vision.

5.3.3 Summary of the multi-cloud federation enabler

Table 5-7 summarized the multi-cloud federation enabler.

Table 5-7 Summary of the multi-cloud federation enabler

Description	The main aim of the multi-cloud federation enabler is to enhance the efficiency, reliability, and flexibility of computing infrastructure by fostering interconnectivity among diverse infrastructures. High level concept design and placement algorithms are studied to allow organizations to leverage a distributed network of computing power, storage, and services, transcending the limitations of individual cloud providers.
Key take-aways	New functionalities are needed to embed the federation in an Operator network. The new modules require automated resource discovery and orchestration in order to create advanced functionalities by combining essential 6G services. New procedure should allow one Operator to extend its network capabilities across borders (e.g., request a network slice or activate the local breakout). The result is a seamless user experience in all EU territory, applications work as in the home country even for edge applications (e.g, Immersive Reality App). Access to more cloud resources, raising the flexibility of services that can be made available, the devices that can be reached and allow for cross-domain deployment. Higher resource availability and flexibility also lowers the total network load and latency times in service deployment, as well the ability to leverage computing power and storage that is superior to current solutions. Monitoring resources (CPU, Memory) and energy usage of cloud-native network functions provides valuable insights into the efficiency of different parts of the 5G Core. These insights can be further analyzed to pinpoint parts of the Core network that use disproportionate amounts of energy and improve their efficiency.
Requirements	Power usage still indicates that the database consumes a disproportionate number of resources. An inquiry into more lightweight databases capable of replacing MongoDB for the 5G Core is needed.
Standard relations & regulations	Procedures and scenarios described in this enabler affects TS 23.501, TS 23.502 of the 3GPP specification.

5.4 Orchestration Transformation

5.4.1 Orchestration of the cloud continuum

5.4.1.1 Introduction

The cloud continuum presents a paradigm shift, extending beyond traditional centralized models to include highly distributed, resource-constrained environments. This transition is characterized by the increasing prominence of extreme-edge devices equipped with embedded computing capabilities. These devices enable local data processing, reduce latency, and support context-aware applications. However, their inherent heterogeneity and volatility necessitate advanced orchestration strategies to ensure seamless integration and optimized performance. A cornerstone of this orchestration strategy is the development and implementation of a multi-cluster resource management platform. This platform provides a unified interface for managing compute resources across the continuum, including capabilities for resource discovery, dynamic allocation, and migration. By leveraging platform-agnostic designs, this approach ensures compatibility across diverse virtualization technologies and aligns with the principles of cloud-native architectures.

The orchestration framework also integrates decentralized management models, enabling collaboration across multiple stakeholders while maintaining autonomy over their respective domains. This decentralized approach promotes the disaggregation of management and orchestration components, facilitating interoperability and fostering innovation through the integration of diverse assets.

Moreover, the framework incorporates the extension of established edge computing paradigms, such as the ETSI MEC framework, to include constrained devices. This adaptation recognizes the unique challenges of resource-limited environments and tailor orchestration functionalities, accordingly, ensuring efficient operation while minimizing overhead.

5.4.1.2 Multi-cluster resource management

The programmability of compute resources across the cloud continuum, and the increasing proliferation of extreme-edge devices (even volatile) with embedded computing resources, are becoming key assets for 6G. These can be leveraged to efficiently deploy and distribute applications by adopting dedicated continuum resources discovery, resource allocation and migration strategies based on continuum nodes' characteristics and constraints (e.g. in terms of compute resource scarcity, battery availability for mobile nodes, compute acceleration options, etc).

For this, the multi-cluster resource management platform has been introduced with its high-level design in deliverable [HEX224-D33], to provide a unified interface for compute continuum resource management (inventory, provision, operate), as well as enhanced placement mechanisms to distribute network functions and applications in the cloud-continuum. The goal is to address resource and proximity constraints through automated discovery mechanisms for virtualization platforms and extreme edge devices. The multi-cluster resource management platform is designed to be agnostic of the underlying virtualization technology, and extensible depending on the scenario and on the involved platforms and devices.

Starting from the high-level design and the operational workflows reported in [HEX224-D33] for continuum resources dynamic discovery, continuous monitoring, and inventory, and platform-agnostic service applications orchestration, a software prototype of the multi-cluster resource management platform has been implemented and already reported in [HEX224-D63], as part of the joint activities on inter-domain and inter-computing resource management and orchestration. For the sake of completeness of this section on the orchestration of the cloud continuum, the rest of this section provides a brief summary of the multi-cluster resource management prototype and its supported features, along with its validation and use in PoCs. Full details can be found in deliverable [HEX224-D63].

The software architecture of the multi-cluster resource management prototype is presented in Figure 5-31. It is a resource orchestration platform which adopts a microservice approach, with different modules developed as REST Java Spring Boot Applications which are integrated to realize the required flexible and dynamic cloud continuum resource management functionalities.

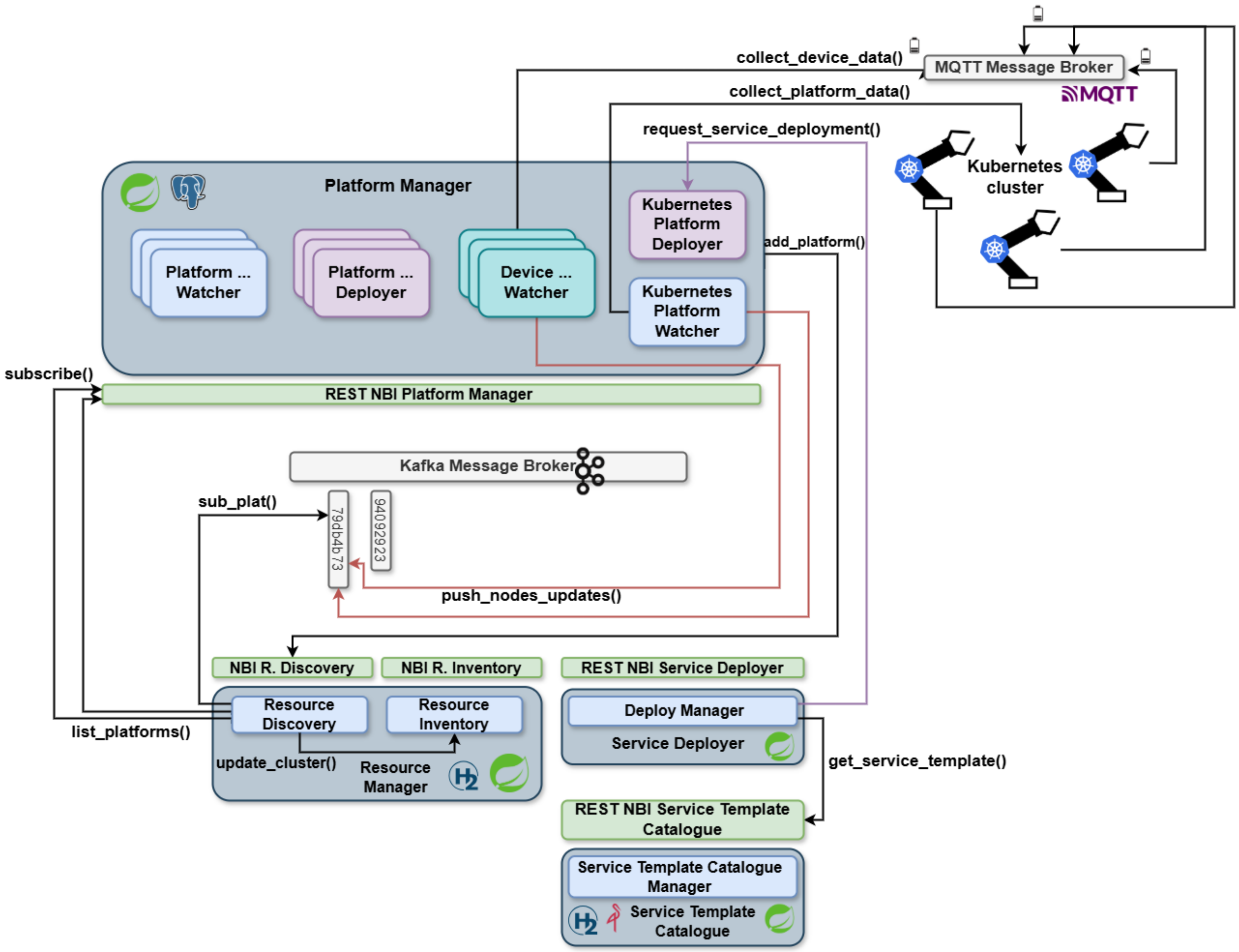


Figure 5-31: Multi-cluster resource management prototype software architecture [HEX224-D63]

A Platform Manager software component implements the dynamic discovery and continuous monitoring of cloud continuum resources (i.e., computing and device characteristics), and it is equipped with multiple drivers to interface with heterogenous virtualization platforms as well as devices. According to the scenario, the drivers can be dynamically activated or de-activated. A PostgreSQL database is used to maintain the characteristics and status of the available virtualization platform. It exposes REST APIs to onboard new platforms (e.g., Kubernetes clusters) to spin up the processes that carry on the discovery and monitory features. A Resource Manager software component creates an inventory of the computing resources and device characteristics of the platforms discovered and monitored by Platform Manager and provides APIs to retrieve the information stored in the internal catalogue. It uses an internal Kafka message broker to communicate with the Platform Manager. The Service Deployer software module provides platform-agnostic REST APIs for service applications orchestration operations (i.e., deploy, delete, update, retrieve status), and translates the platform-agnostic operations requests into platform-specific ones (managed by the Platform Manager). It is tightly integrated with the Service Template Catalogue software module, which provides CRUD REST APIs for managing service templates (e.g., Helm Charts, Heat Templates, Kubernetes Manifests) stored in a Minio Object Storage, which can be retrieved for the proper deployment and orchestration of service applications. The various REST APIs exposed by the multi-cluster resource management platform leverage the cloud-continuum resource data model depicted in Figure 5-32.

The multi-cluster resource management platform prototype presented in Figure 5-31 has been validated in the context of the PoC-B, where it has been integrated to discover the resources and the device characteristics of a Kubernetes cluster hosted in multiple cobots, and to perform the deployment and migration of a cobot application based on its monitored battery level. The multi-cluster resource management operational workflows have been validated together with other management and orchestration enablers and components, namely Monitoring Platform, Closed Loop Governance and Service Orchestrator. Full details can be found in deliverable [HEX224-D63].

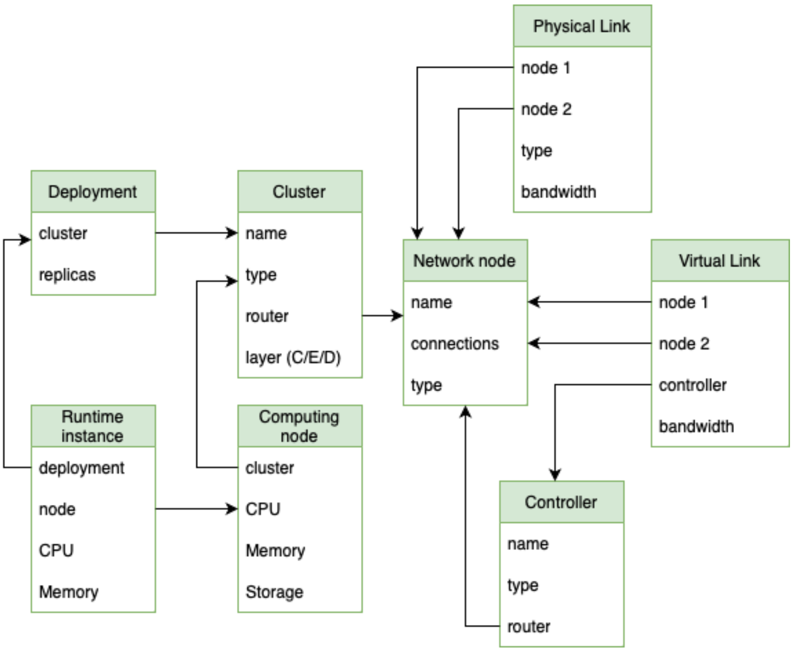


Figure 5-32: Data model for continuum resources [HEX224-D63]

In summary, the multi-cluster resource management platform has been proved to provide a unified and abstract interface for compute continuum resource management (inventory, provision, operate), with enhanced placement mechanisms to efficiently deploy and migrate network functions and service applications that have proximity constraints. Automated discovery mechanisms for virtualization platforms and extreme edge devices capabilities allow to maintain up-to-date cloud-continuum resource awareness, while tailored monitoring jobs for extreme-edge devices attributes, status and behaviour enables the implementation of zero-touch closed-loop automation mechanisms.

5.4.1.3 Decentralised orchestration

In previous generations of mobile networks (5G and earlier), the elements dedicated to network services and resources M&O are generally deployed within the domain of a single provider, the MNO, which uses them to manage and orchestrate resources and services within its own domain. The decentralized orchestration approach described here promotes the disaggregation of these components, allowing different providers to offer specific parts of their infrastructure and network service components, with the aim of facilitating interoperability integrating assets from different providers, which may extend beyond the domain of a single operator, into what is called its extreme-edge.

This decentralised M&O approach was conceptually introduced in this WP3 from the very beginning of the project ([HEX223-D32] and [HEX224-D33] in sections 8.1.3 and 11.4, respectively), and, further elaborated and developed in WP6 with certain practical implementations ([HEX223-D63], section 3.5.2).

What this decentralised M&O approach proposes is well in line with the cloud-native principles[KP17] [PER09], targeting both: the M&O of the network multi-domain infrastructure resources, and the network services that would be deployed on them. The objective is both technical and business-oriented. Technically, it aims to address the challenge of orchestrating resources and services across the entire network continuum. From a business perspective, it offers network operators and other stakeholders within the continuum opportunities to implement new business models to leverage the deployment of profitable network services on resources extending beyond their own network domains [6GW24].

As a whole, the decentralised M&O approach proposes to deploy on the whole network continuum multiple instances of four kind of M&O-related nodes, in a fully distributed fashion, and that would be mainly intended to support the network services provisioning and certain infrastructure resources management mechanisms. These nodes are:

- The DN (Deployment Node), which would act as a front-end to the stakeholders for them to deploy their network services on the 6G network, which could be accessed as a platform for them to both: to deploy their own network service components and to share and monetize them with other stakeholders (including end-users) in a sort of marketplace.
- The IRN (Infrastructure Registry Node), used to build the so-called Infrastructure Registry Service (as an aggregation of these nodes), to form a wide scale distributed database in charge of keeping an updated registry of the infrastructure resources available in the network continuum. Inspired on the well-known distributed hierarchical DNS system [RFC-1034] but enabled with dynamic infrastructure discovery mechanisms and adapted to the volatility and diversity of the extreme-edge domain. Supports the DN for the services deployment.
- The SRN (Services Registry Node), like the IRN in what regards its implementation, but focusing on the deployment status of the network services deployed on the network. Due the volatility of the extreme-edge domain, the deployment nodes on which service components may be running can change over time. Considering that, the Services Registry Service (made as an aggregation of SRNs) provides updated information on which specific nodes the network service components may be actually deployed.
- The ISPN (Infrastructure Status Prediction Node), which complements the IRN providing predictions on the status of the infrastructure devices. Since infrastructure status and network user behaviours can change according certain regular patterns, these nodes can provide predictions based on data analytics, which could be performed relying on AI/ML techniques.

As mentioned, these four kinds of nodes would be spread through the whole network continuum, in a highly decentralised approach. However, they would indeed be hosted by specific stakeholders in the network continuum with the specific capabilities, permissions, and resources to deploy network service components on the network (e.g., different mobile network operators, but also, other stakeholders part of the continuum, such as vertical industries, neutral host providers, or hyper-scalers, among others).

Besides these four kind of nodes, the decentralised M&O approach also proposes the innovative concept of embedding the network services assurance mechanisms as part of the network services themselves, i.e., instead of relying on a centralised services assurance system (e.g., part of a centralised M&O framework deployed within a specific stakeholder scope) the service assurance mechanisms would be deployed within each service itself, and with a scale and purpose tailor-made for each service. So, in practice, those M&O assurance mechanisms would be also distributed through the network, together with the network services to which they belong, in a cloud-native way [HEX223-D32] [HEX224-D33]. This approach considers the massive (cloud-native in scale) amount of network service components that could be deployed on the continuum, which would make a centralized approach impractical and hardly scalable, although this approach is of course subject of concerns as different services may have contradicting assurance mechanisms. It also considers that network services can be quite different in practice, with very different requirements in what regards their needed assurance mechanisms (e.g., some of them could just require to monitor a small set of metrics and implement simple automation mechanisms, while others could require more sophisticated approaches relying on AI/ML techniques and massive data processing). These tailor-made service assurance mechanisms could be implemented relying on already available SotA technologies, such as Kubernetes [K8S] and derivatives (e.g., [K3S] [MIC][Min24a]), other containers orchestration frameworks [SWA][SWI][MES24][Nom24], or microservices choreographies [CDT18]. However, it is considered that, from a conceptual perspective, the decentralised approach presented here can be flexible enough to also integrate other technologies that could be available in the future, which also contributes to promote a diverse and dynamic ecosystem in terms of services development, in line with the cloud-native practices.

In fact, for both, network services implementation and the four M&O nodes described above, the adoption of the cloud-native design and development practices are strongly encouraged to enable stakeholders to deploy and operate the network services across domains, and integrating with third-party infrastructure in addition to their own. Specifically, an enabling technology for this vision would include a wide adoption of APIs and microservices for building a Services Oriented Architecture (SOA). E.g., APIs like CAMARA [Cam24] or CAPIF [29.222] could be used to enable the communication with and among the above-mentioned nodes. This would facilitate secure access to the different stakeholders' network capabilities by different parties. However, beyond these specific API technologies, the well-known API manifesto for services externalization [Yeg11] should be extensively adopted. This, along with the microservices federation concept [GKV+19] [FSP+20] could help to integrate resources and deploy and orchestrate network service components across the various network domains in a highly disaggregated way, enabling a “multi-domain by design” approach.

The implementation of some of the concepts associated to this Decentralised M&O approach has been addressed in the context of WP6 and the project PoCs. Specifically, two components have been developed: The first is the so-called Infrastructure Layer Emulator (ILE), designed to replicate the infrastructure layer envisioned for future 6G networks in a manageable and cost-effective manner. Developing this emulator is deemed essential to create a realistic environment for testing and demonstrating the design M&O-related concepts of future 6G systems. Its key features include:

- The emulation of the different network domains under consideration, i.e., core, edge, and extreme-edge domains.
- The possibility to deploy a large number of computing nodes (the extreme-edge is envisaged to be massive in scale), which can be of different types (besides large, the network continuum is highly heterogeneous).
- The emulation of different stakeholders.
- For the extreme-edge resources, the emulation of its inherent high-volatility, with devices unexpectedly connecting/disconnecting and/or changing certain properties (e.g., memory or CPU occupancy, or the battery level in the case of battery power devices).
- And of course, with the possibility to deploy realistic network services on such emulator.

The ILE implementation leverages LXD [LXD24], a system container and virtual machine manager that offers a unified platform for running and managing complete Linux systems within containers or virtual machines. It supports a wide range of Linux distribution images, from lightweight options (ideal for deploying numerous Linux nodes on resource-constrained equipment) to full-featured distributions capable of hosting and running complex network services. LXD can scale from a single instance on one machine to a cluster emulating an entire data centre, making it well-suited for simulating various scenarios. This ILE component has been released to the open-source software community as one of the outcomes of the project [ILE24].

The second is an implementation of the ISPN described above, which has been used to implement a proactive orchestration mechanism on a network service deployed on the ILE. This implementation, relying on AI/ML techniques, is used to predict the connectivity status of the extreme edge devices in the ILE, and based on that, trigger orchestration signals to proactively live-migrate the network service components when the ISPN forecasts that the nodes on which they are deployed could disconnect in a brief span of time. The target is of course to keep the network service continuity triggering service relocation actions to adapt the deployment status to the dynamically changing connectivity status of the extreme-edge nodes. Figure 5-33 shows an execution example where a network media streaming service consisting of three service components (blue line) is deployed on three extreme edge nodes, emulated by the ILE. As it can be appreciated, when the ISPN detects that these nodes could be disconnected (available nodes are represented in green, while unavailable ones are in red) the service components are proactively migrated to new nodes (right side of the figure). For the experiments, a video streaming media service was used, observing that there were no service disruptions (discontinuities in the video playback) during the service migration process. More details on this implementation are provided in Deliverable D6.5 [HEX224-D65].

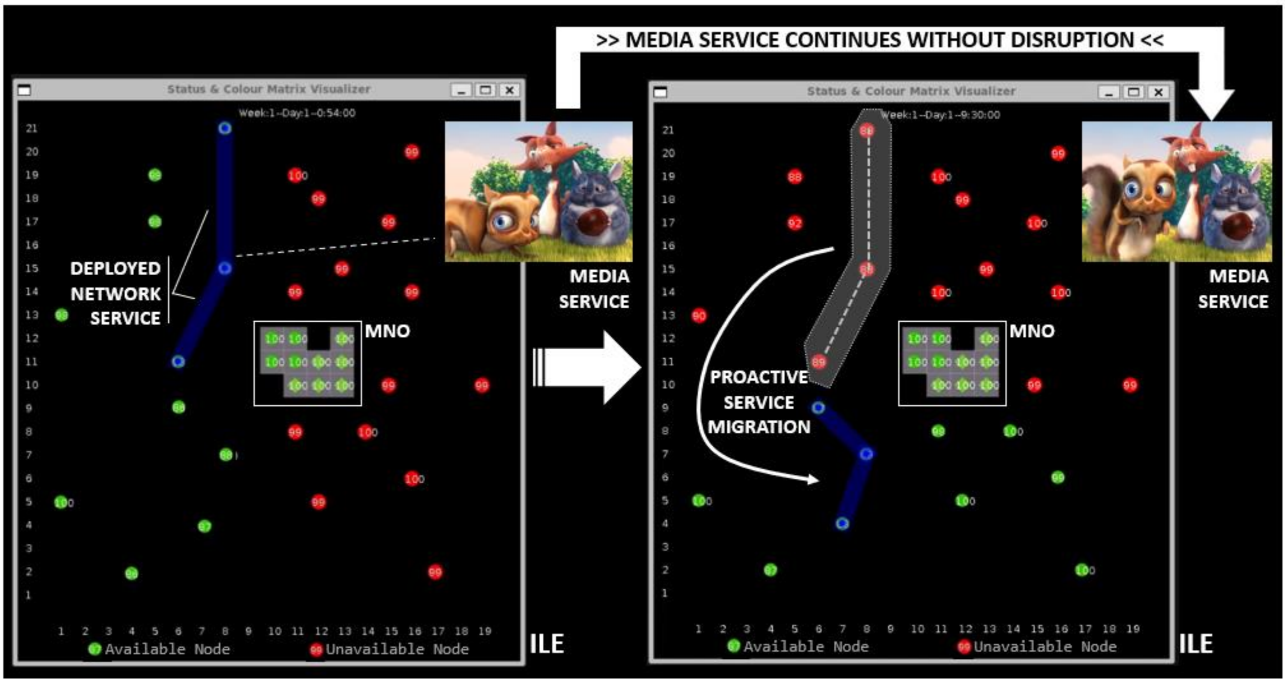


Figure 5-33: Proactive service migration triggered by the ISPN.

5.4.1.4 *Orchestration of the extreme edge*

In [RGO+24] a constrained MEC (cMEC) architecture is presented. A constrained MEC (cMEC) architecture is needed to define a streamlined schema of MEC functionalities obtained by extending the cloud-edge-user-layer architectural model, incorporating a novel layer to represent the devices beyond the RAN. The inclusion of this new layer is abstracted away from developers and users, allowing them to use the complex MEC system and its APIs. The cMEC departs from the traditional MEC (tMEC) framework and exhibits attributes customized and particular to devices with limitations:

- **Efficient Features:** The cMEC is able to function as a complete MEC system, encompassing all its elements if the computational resources (e.g., industrial PC) in the extreme edge are powerful enough to handle the complete framework. However, due to the limited resources in extreme edge devices, the cMEC might support only a subset of MEC functionalities. For example, the MEC Orchestrator, with its resource-demanding functions, may be opted out in particularly restrictive circumstances only if a less demanding action is feasible.
- **A Layered design:** given that the tMEC depends on the edge for computational for offloading, content fetching, user authentication, and context, cMEC depends on tMEC for similar functions. This layered strategy necessitates an interconnected relationship between cMEC and tMEC, without involving the implementation of federation concepts that require explicit business agreements and reliance on orchestrators.

Figure 5-34 details the architectural implications to interconnect the cMEC with the tMEC, without the MEO being present in the cMEC system. The architectural implications of interconnecting the cMEC and the tMEC include a cross-system reference points inter Mm2 and inter-Mm3 that are primarily introduced to facilitate the setup of the cMEC-tMEC interconnection. The Mx2 reference point is expanded to enable users to initiate lifecycle management actions (such as instantiation, deletion, or updates) of MEC applications within a cMEC or even a tMEC. Consequently, the inter-Mx2 interface, linking the cMEC application proxy to that of the tMEC, can ensure a certain level of consistency between cross-system applications (i.e., those spanning multiple layers) and enable any request to be propagated from cMEC to tMEC. Lastly, the Mp1 reference point, connecting MEC applications and services with their respective platforms, should be extended as an inter-Mp1 reference point for service consumption and app-to-app communications between different systems.

The Operational Support System (OSS) is a tool (from the service provider) working at the MEC level and may not always be associated with a subordinate local cMEC for application onboarding and instantiation. These tasks, typically carried out by a network manager working within the MEC through the OSS, might require initiation by the end user (for example, requesting a specific application for their home or vehicle) and managed by the cloud and the remote OSS and MEO of the tMEC, utilizing alternative workflows that support a new range of cross-system MEC interfaces. This includes interfaces such as Mx2 and Mm8 from the traditional MEC framework should be enhanced to allow users to trigger new instantiations.

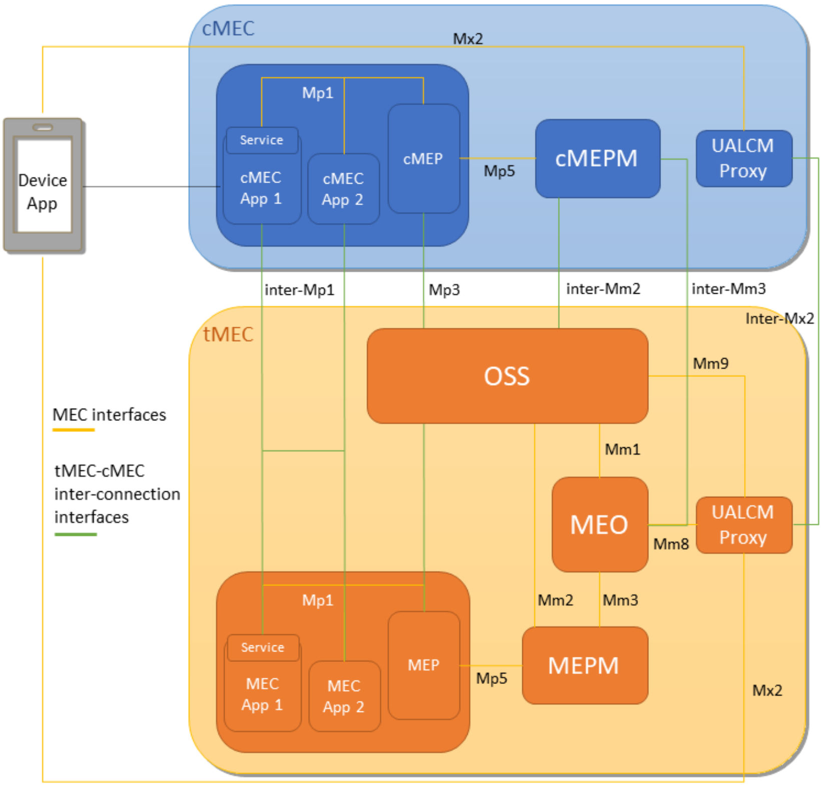


Figure 5-34 Architectural scheme of constrained MEC together with traditional MEC [RGO+24]

High-level cMEC workflows [RGO+24].

The integration of cMEC with a tMEC requires additional workflows. In the following, three key operations are described: i) discovery and interconnection; ii) application on-boarding and instantiation; and iii) service availability and consumption.

Discovery and Interconnection

The cMEC discovery by the tMEC or by other cMEC systems is a necessary step to their interconnection. It consists of either i) making a tMEC system aware of a the cMEC; or ii) discovering peer cMECs. Moreover, the cMEC does not support orchestration (i.e., it does not comprise a MEO element). At the same time, cMEC can be co-located in an end-user device. The challenge for an end-user device to discover a nearby standalone cMEC is left outside of the scope of this work, since multiple protocols (not directly related to ETSI MEC) can serve this purpose.

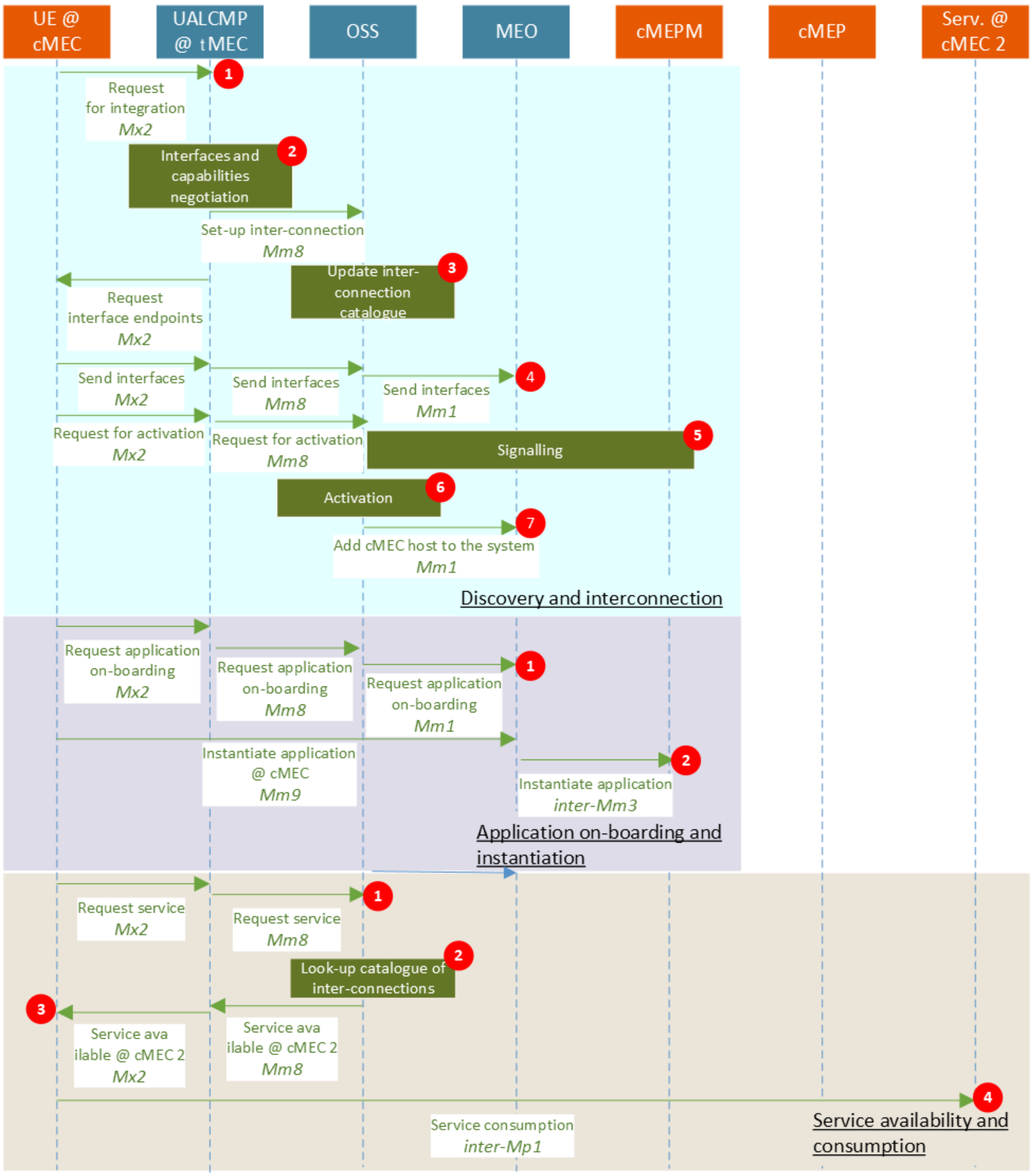

UE @ cMEC
UALCMP @ tMEC
OSS
MEO
cMEPM
cMEP
Serv. @ cMEC 2
1
Request for integration Mx2
2
Interfaces and capabilities negotiation
Set-up inter-connection Mm8
3
Update inter-connection catalogue
Request interface endpoints Mx2
Send interfaces Mx2
Send interfaces Mm8
Send interfaces Mm1
4
Request for activation Mx2
Request for activation Mm8
Signalling
5
Activation
6
Add cMEC host to the system Mm1
7
Discovery and interconnection
Request application on-boarding Mx2
Request application on-boarding Mm8
Request application on-boarding Mm1
1
Instantiate application @ cMEC Mm9
Instantiate application inter-Mm3
2
Application on-boarding and instantiation
Request service Mx2
Request service Mm8
1
Look-up catalogue of inter-connections
2
3
Service ava ilable @ cMEC 2 Mx2
Service ava ilable @ cMEC 2 Mm8
Service consumption inter-Mp1
4
Service availability and consumption

Figure 5-35: Architectural scheme of cMEC together with tMEC

The message workflow for a cMEC to advertise itself to a tMEC is presented within the box titled *Discovery and Interconnection* of Figure 5-35. The cMEC reaches out to the UALCMP of the tMEC it wants to integrate with, by issuing a *Request for Integration* message (step 1), through which the cMEC advertises the interfaces and the computational capabilities (i.e., computing, storage and network resources, MEC services, etc.) to be shared (step 2). The OSS can then update the catalogue of interconnections with this new information (step 3), as cMECs rely on tMEC MEO for coordination. Afterwards, the cMEC can proceed to send the agreed interface addresses (step 4).

After signalling between the cross-system interfaces (OSS contacting cMEPM through *inter-Mm2* and MEO contacting cMEP through *inter-Mm3*) to check interconnectivity (step 5), the process is finalized when the interconnection is activated in the OSS (step 6). An activation step is necessary as the cMEC can move away from the tMEC during the procedure. Finally, the MEO module adds the cMEC to its host list (step 7), so that, if granted permissions, the cMEC host can be selected by the orchestrator for application on-boarding and instantiation.

Application On-boarding and Instantiation

In standard ETSI MEC, the package on-boarding request for an application is initiated by the operator interacting with the OSS. Then, the actual application instantiation is subject to the MEO's decision, which normally evaluates the application requirements and performs host selection accordingly. As previously mentioned, a cMEC host connected to a tMEC, becomes a host of the tMEC system, so that it can be automatically selected by the MEO for application deployments when needed. As illustrated in the box titled *Application On-boarding and Instantiation* of Figure 5-35, the cMEC Device App contacts the tMEC UALCMP, which solicits the OSS to grant the on-boarding permissions. The same on-boarding request would then reach the MEO (step 1). At this point, the MEO would have, according to the current standard, to perform host selection. The current specification does not define how host selection is realized in practise. In such case, the cMEC could request the MEO to select the desired cMEC, and not an arbitrary host of the tMEC system selected by the MEO's algorithm. The actual package on-boarding and app instantiation processes are later triggered by the MEO in the cMEPM through the inter-Mm3 reference point (step 2).

Service Availability and Consumption

A MEC application, whether deployed in a cMEC or a tMEC, might also request a MEC service not locally available. As the Service Availability and Consumption box of Figure 5-35 represents, this can be tackled by issuing a service request to the OSS (step 1), followed by a lookup in the catalogue of interconnections (step 2). The lookup goal is to identify if a service is available in a cMEC or tMEC, which would then communicate the availability details (step 3) to the MEC application. If the requesting MEC (cMEC or tMEC) is not interconnected to the target MEC, the interconnection is invoked by the MEO or the OSS. The service consumption between the MEC application and the remote service can then occur via the *inter-Mp1* interface (step 4). Alternatively, a dedicated service management proxy can be introduced in every cMEC to manage service availability. However, it prevents the cMEC to benefit from remote services belonging to cMEC systems not directly interconnected: proxies must be known to the cMECs in advance. A last option can rely on sending queries about service availability directly towards the MEO, which then queries each of the cMEPMs and provides an answer based on the information stored in their cMEP's *service registries*.

5.4.1.5 Summary of the orchestration of the cloud continuum enabler

The orchestration of the cloud continuum enabler is summarized in Table 5-8.

Table 5-8 Summary of the orchestration of the cloud continuum enabler

Description	With 6G, services and network functions deployment is envisaged to be addressed considering the whole cloud-continuum, including the extreme-edge domain, that can be heterogeneous in technology, massive in scale, multi-stakeholder and volatile in terms of availability. Three complementary solutions have been designed to address the challenges of the cloud-continuum orchestration: Multi-cluster resource management, Decentralised orchestration, Orchestration of the extreme edge.
Key take-aways	Multi-cluster resource management: Unified and abstract interface for compute continuum resource management (inventory, provision, operate) Automated discovery mechanisms for virtualization platforms and extreme edge devices capabilities Enhanced placement mechanisms to efficiently deploy and migrate network functions that have proximity constraintsP Tailored monitoring jobs for extreme-edge devices attributes, status and behaviour to feed zero-touch closed-loop automation mechanisms

	Decentralised orchestration: Traditional mobile network operations were centralized within a single Mobile Network Operator (MNO). The decentralized approach disaggregates M&O components, allowing multiple stakeholders (e.g., MNOs, vertical industries, hyperscalers) to manage and orchestrate services across different domains, including extreme-edge networks. This approach aligns with cloud-native principles and aims to enhance interoperability and scalability by leveraging microservices, APIs, and dynamic orchestration mechanisms. Orchestration of the extreme edge: Modifications to current protocols are needed to account for constrained devices in close locality of the user. Research enables to include computing resources spread in the cloud continuum in the ETSI MEC edge infrastructure.
Requirements	Multi-cluster resource management: Need continuous access to extreme-edge devices and related control systems (such as Kubernetes nodes and controllers, that in case of mobile and volatile nodes may require access through a wireless network) Decentralised orchestration: The entire M&O framework must follow cloud-native principles, leveraging microservices, APIs, and service-oriented architectures (SOA) to ensure flexibility, scalability, and interoperability. Orchestration of the extreme edge: The edge system must be compliant with ETSI MEC.
Standard relations & regulations	Multi-cluster resource management: The solution proposed can play the role of particular enhanced ETSI NFV MANO VIM. Its functionalities are mainly focused on the management of the resources of the extreme-edge, edge and cloud continuum layer (infrastructure layer) while offering advanced functions like the management of multiple domains, dynamic discovery of the infrastructure layer resources (and their continuous monitoring), as well as the possibility to orchestrate service application over the managed resources. Orchestration of the extreme edge: Proposed modifications have been submitted for consideration in ETSI MEC in constrain devices study item.

5.4.2 6G Slicing and Orchestration

5.4.2.1 Introduction

The evolution of 6G networks introduces a transformative shift in how network resources and services are managed and orchestrated. Central to this evolution is the concept of network slicing, which builds on the foundational work done in 5G while expanding its scope to address the unique requirements of 6G. Network slicing allows the creation of multiple virtualized network instances over shared physical infrastructure, each tailored to meet specific performance, security, and reliability requirements. In 6G, the challenge lies in scaling these capabilities across multi-operator and multi-domain environments, ensuring seamless interworking while retaining logical isolation and robust service quality.

Orchestration in 6G moves beyond static configurations, focusing on dynamic, automated mechanisms that respond to real-time network conditions and user demands. Slicing orchestration involves coordinating resources to ensure that slices function efficiently across heterogeneous infrastructures. With the increasing coexistence of 5G and 6G deployments, solutions must address challenges such as unified slice identification and inter-domain resource management. Advanced techniques, including intent-based management, further streamline this process by enabling operators to define high-level goals rather than configuring low-level technical details. This abstraction not only simplifies operations but also enhances the agility required for modern, federated networks where multiple operators collaborate to share resources.

The traditional operator-centric slicing model is also being redefined in 6G to align with decentralized and multi-stakeholder approaches. Slices are no longer confined to a single operator's domain but span the entire network continuum, incorporating resources from diverse participants, including industries and edge providers. This shift necessitates the development of frameworks that support cross-domain orchestration and allow stakeholders to contribute to and benefit from shared slicing capabilities. These frameworks promote sustainability, cost efficiency, and innovation by enabling tailored slice definitions aligned with specific use cases and resource capabilities.

In parallel, slicing mechanisms are being integrated into decentralized compute-continuum management models, leveraging the distributed nature of 6G networks. This integration ensures that slices can be deployed and managed flexibly across the continuum, from centralized data centers to extreme-edge nodes. Such an approach fosters the creation of diverse, highly specialized slices that address the needs of various stakeholders, including IoT ecosystems, industrial applications, and latency-sensitive services.

These advancements in slicing and orchestration provide a solid foundation for 6G networks to support complex, multi-domain environments. By combining dynamic slicing, intent-based management, and decentralized orchestration, 6G networks aim to deliver unmatched flexibility, scalability, and service quality, ensuring that they can meet the diverse demands of future applications and use cases.

5.4.2.2 Intent based management/orchestration

In 6G networks, service orchestration needs to respond dynamically to a highly variable, multi-operator environment where user demands, network loads, and service conditions change continuously. Moreover, in a federated 6G environment, where multiple network operators share resources and infrastructure, an orchestrator module becomes essential. In a cloud continuum scenario, the orchestrator module serves as the key enabler for such collaborative frameworks. By supporting interoperability and peer-to-peer interaction among orchestrators from different operators, the orchestrator module in a 6G federated ecosystem ensures seamless integration of network capabilities and optimizes resource allocation. To meet these needs, the orchestration model must support a flexible, automated service flow and request handling mechanism that allows network operators to express their desired outcomes rather than specifying complex, low-level configurations.

Intent-based interfaces enable this abstraction by allowing operators to communicate "intents," or high-level service requests, which the orchestration model interprets and translates into actionable tasks that fulfil these intents. Intent-based interfaces in a 6G orchestration model allow network operators, service providers, and end-users to specify what they want to achieve without defining the exact means to accomplish it. These interfaces interpret high-level requests, such as "maximize network throughput" or "reduce latency for edge applications" and translate them into executable actions and configurations even across federated network infrastructures. The implementation of intent-based interfaces is essential for meeting the complex demands of modern networking applications, including real-time service adjustments, cross-operator coordination, and compliance with regulatory constraints in multi-national contexts. In federated scenarios, multiple network operators collaborate to form a cloud continuum, pooling their assets and capabilities. The orchestrator module acts as the centralized intelligence that enables coordination and integration across infrastructures.

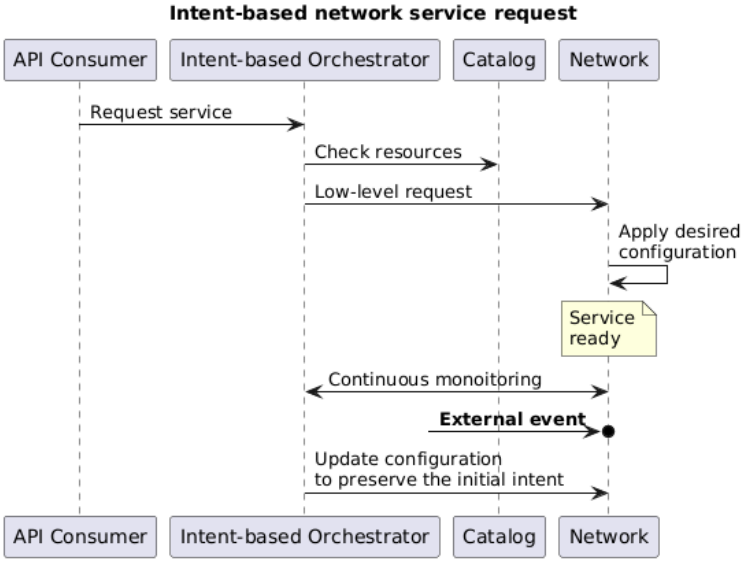

Intent-based network service request
API Consumer
Intent-based Orchestrator
Catalog
Network
Request service
Check resources
Low-level request
Apply desired configuration
Service ready
Continuous monoitoring
External event
Update configuration to preserve the initial intent

Figure 5-36: Intent-based network service request

The service flow in an intent-based orchestration model is a multi-step process that transforms high-level intents into low-level configurations and actions as shown in Figure 5-36.

In the initial stage, the network operator or application defines a service intent through the orchestration interface where these intents could be related to specific networking needs (e.g., enhanced QoS for specific applications, network slicing to allocate dedicated resources to critical applications or latency optimization for services requiring real-time responses). The orchestration model validates the service intent against predefined policies, SLAs, and regulatory requirements specific to each operator and national boundary. This ensures that service requests comply with data residency, privacy, and usage policies, essential for applications that operate across jurisdictions.

Once the intent passes policy checks, the orchestration model proceeds to allocate resources across the federated network infrastructure. This phase involves:

- **Mapping Intents to Resources**: The orchestrator assigns compute, storage, and network policies based on the service's needs.
- **Network Function Orchestration**: Configures the network functions necessary for the service request. For instance, in response to a “high-throughput” intent, the orchestrator may deploy additional VNFs to load-balance traffic across multiple data centers.
- **Dynamic Edge Node Selection**: For intents focused on reducing latency, the orchestrator identifies the nearest and least congested edge nodes to handle processing, improving performance for latency-sensitive applications.

Once the resources are provisioned, the orchestration model continuously monitors network performance and compliance to ensure that the intent is fulfilled. Through a real-time feedback loop, the orchestrator can detect variations in network load, service quality, or compliance status and make adjustments as needed. For example, if traffic spikes suddenly, the orchestrator may trigger automatic scaling to add additional capacity or reconfigure network slices to prioritize critical services. This feedback loop is also integral for intent-based requests that involve adaptive requirements, such as scaling up during peak hours or dynamically adjusting latency targets based on user demand.

In a federated cloud continuum, each network operator maintains an independent orchestrator that coordinates with other orchestrators via a peer-to-peer model. The orchestrator leverages a dedicated Federation Module to interact with federated partners. This distributed approach allows operators to retain autonomy over their infrastructure while collaborating for shared goals.

The orchestrator engine is responsible for coordinating multi-domain services across federated networks. This allows the deployment of complex services that span multiple operator networks managing service lifecycle (i.e., deployment, scaling, and termination of services across different domains), ensuring local policy compliance and fault tolerance recovery.

The orchestration model's use of intent-based interfaces in 6G federated networks marks a transformative approach to managing networking applications. By focusing on service flow automation and intuitive request patterns, this model aligns closely with the agile and adaptive demands of modern applications. Intent-based orchestration not only simplifies cross-border, multi-operator collaborations but also ensures that network resources are optimized, regulatory compliance is upheld, and security measures are robustly enforced. For network operators, adopting intent-based orchestration is a strategic move toward achieving efficient, resilient, and customer-focused services, crucial for realizing the full potential of the 6G cloud continuum.

5.4.2.3 6G Slicing

In Figure 5-38, once the 6G AMF determines the mapping between the 6G and 5G network slices used for inter-RAT slice interworking, the 6G AMF provides to the UE in addition to the 6G Allowed NSSAI list where it includes the 6G S-NSSAI, also an additional 5G Allowed NSSAI list where it includes the list of 5G S-NSSAIs. The UE shall store both Allowed NSSAI lists and use them respective to the RAT where the UE operates on. The benefit of UE having two Allowed NSSAI lists which are RAT dependent could be such that the UE does not have to be re-configured by the target RAT with a new Allowed NSSAI list. Furthermore, it could be used for cell reselection purposes. Utilizing two lists compared to a shared list can also be due to the fact that currently the Allowed NSSAI list has a limit of 8 S-NSSAIs and using the feature of inter-RAT handover mapping for inter-RAT slice interworking the amount of S-NSSAI in the list can grow larger.

In Figure 5-38, when the source 6G RAN decides to handover the 6G capable UE to a 5G RAN due to coverage issues, the source 6G RAN will send a Handover Required message to the 6G AMF including the PDU session ID and target ID as done today for NG handover. The 6G AMF will identify the proper AMF based on the mapped 6G slice to 5G slice used for inter-RAT slice interworking for that PDU session. The 6G AMF will then send a message related to creation of UE context towards the AMF indicating the PDU session ID and the mapped 5G slice that corresponds to the 6G slice used by the UE in the 6G network or alternatively 6G AMF once requesting the respective AMF to create UE context during the inter-RAT handover can forward both slice IDs to the AMF for later use. AMF can store both slice IDs and use it for instance when the UE goes back from 5G to 6G network for finding the proper 6G AMF that supports the slice of the UE. Since the AMF does not have information of 6G slices this stored information can be used in the communication with NRF to find the proper 6G AMF.

Once handover execution has terminated and the UE has completed the handover to the new target 5G RAN the UE will either register with the AMF but will not receive an updated Allowed NSSAI list since it already received one during its registration at the 6G network. This new approach can be achieved if we envision changes in current 5G AMF behavior. Otherwise, if the current AMF behavior will remain the same as in 5G, then the UE has to perform registration again with the AMF and UE will receive the 5G Allowed NSSAI list.

In the latter scenario, the UE although it receives an updated 5G Allowed NSSAI list it will still store the 6G Allowed NSSAI list to be used in case that the UE goes back to the 6G network again.

In short, Figure 5-37, the goal is to define two NSSAI lists, one per RAT and configure this to the UE. In Figure 5-38, the goal is to define the switching mechanism in case the UE changes from 5G to 6G. They are distinct but complementary solutions of the inter-RAT network slicing mechanism.

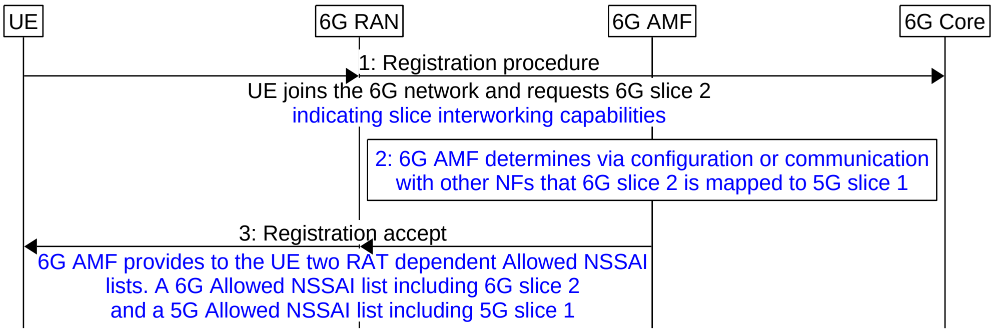

UE
6G RAN
6G AMF
6G Core
1: Registration procedure
UE joins the 6G network and requests 6G slice 2
indicating slice interworking capabilities
2: 6G AMF determines via configuration or communication
with other NFs that 6G slice 2 is mapped to 5G slice 1
3: Registration accept
6G AMF provides to the UE two RAT dependent Allowed NSSAI
lists. A 6G Allowed NSSAI list including 6G slice 2
and a 5G Allowed NSSAI list including 5G slice 1

Figure 5-37: Registration to a 6G slice

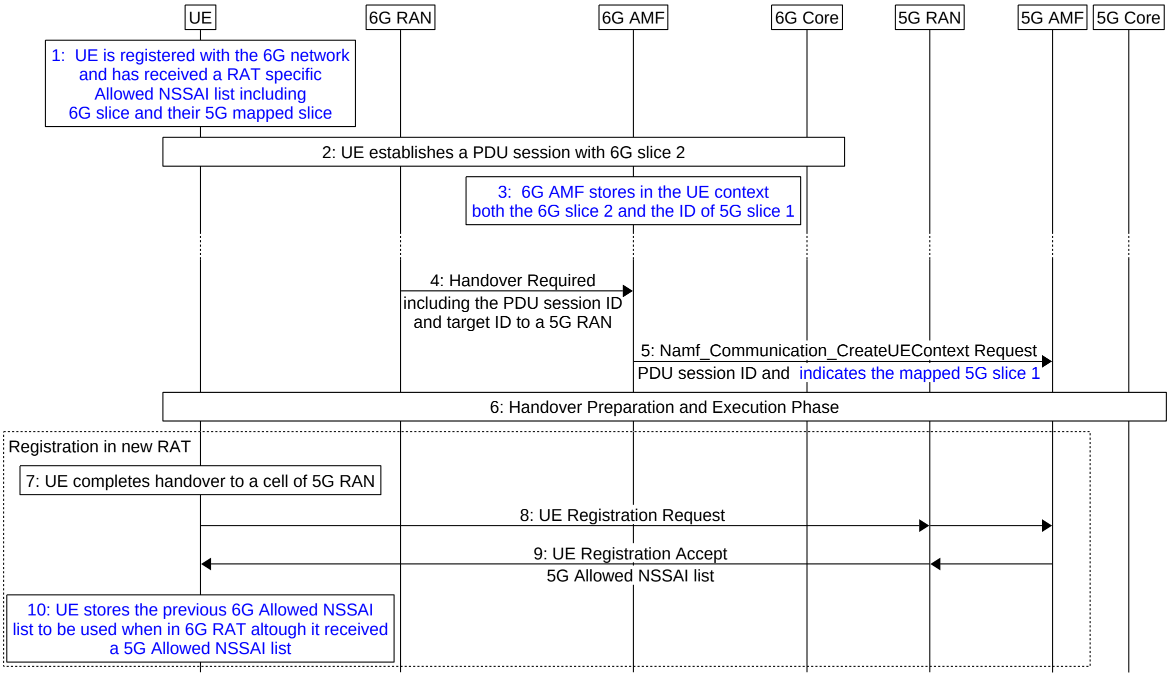

UE
6G RAN
6G AMF
6G Core
5G RAN
5G AMF
5G Core
1: UE is registered with the 6G network
and has received a RAT specific
Allowed NSSAI list including
6G slice and their 5G mapped slice
2: UE establishes a PDU session with 6G slice 2
3: 6G AMF stores in the UE context
both the 6G slice 2 and the ID of 5G slice 1
4: Handover Required
including the PDU session ID
and target ID to a 5G RAN
5: Namf_Communication_CreateUEContext Request
PDU session ID and indicates the mapped 5G slice 1
6: Handover Preparation and Execution Phase
Registration in new RAT
7: UE completes handover to a cell of 5G RAN
8: UE Registration Request
9: UE Registration Accept
5G Allowed NSSAI list
10: UE stores the previous 6G Allowed NSSAI
list to be used when in 6G RAT altough it received
a 5G Allowed NSSAI list

Figure 5-38: 6G AMF performs Switching of 6G to 5G slice and indicates it to 5G AMF

5.4.2.4 Network Slicing in the decentralised compute-continuum management approach

The concept of network slicing refers to the ability of a telecommunications network to virtually segment itself into multiple logical networks (or "slices") that operate independently while sharing the same underlying physical infrastructure. This technology has been identified as a cornerstone for enabling flexibility and customisation in 5G networks, addressing the diverse needs of users and applications. Each slice operates as a separate virtual network, ensuring that issues or failures in one slice do not impact others. Also, according to [GSM17], each slice is designed to meet specific dynamic segmentation requirements for quality of service (QoS), latency, bandwidth, or security, enabling tailored solutions for different use cases. Besides, the requirements outlined by the Next Generation Mobile Networks Alliance [NGM16] emphasise that slices need to be managed end-to-end, ensuring efficient allocation of resources across all network layers, from the core to the access layer. As a whole, the concept is strongly based on advanced integration of virtualisation and orchestration techniques.

However, this concept of network slicing is implemented within the domain of a specific operator, i.e., it is an operator that isolates and “slices” different assets in its own infrastructure (e.g., core, edge, access network resources) in order to offer isolated slices with specific features and purposes (low latency, high availability…) to other external stakeholders for them to implement their network services. In other words, the concept is a form of “outsourcing” of the operator's resources so that other parties can access the operator infrastructure and implement their own network services on it.

This concept of network slicing is represented at high level in Figure 5-39 [NGM16] where, as it can be appreciated, three layers are considered: the Resource Layer (identifying the resources belonging to the operator), the Network Slice Instance Layer (where different network slice instances are defined based on those resources), and the Service Instance Layer (representing the services implemented on those network slice instances).

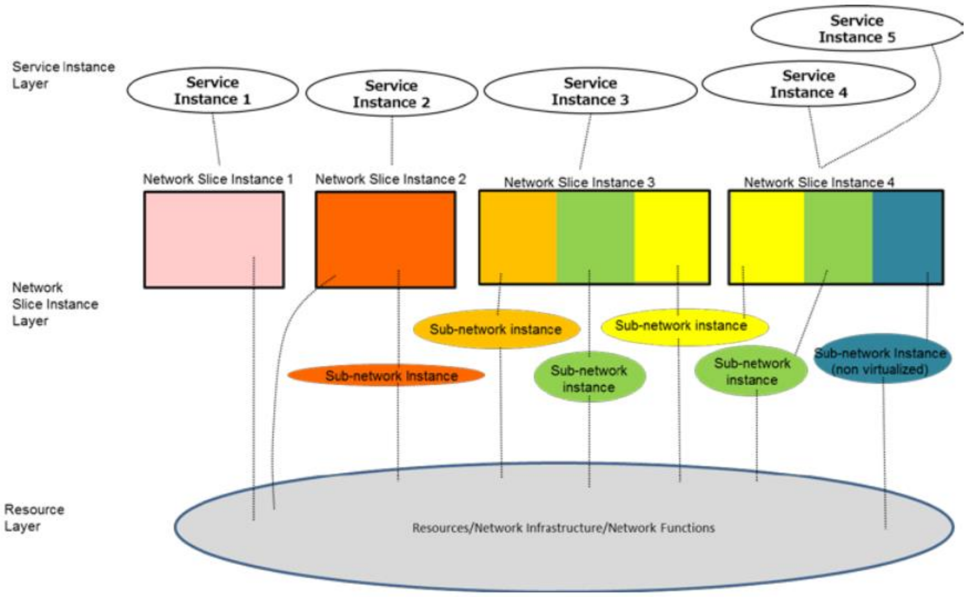


Figure 5-39: Network Slicing Conceptual Outline [NGM16]

At first glance, it might seem that this operator-centric view of the network slicing concept might collide with the network continuum concept that we are targeting in Hexa-X-II, and specifically, with the concept of the decentralized orchestration referred to in Section 5.4.1.3, since this view is essentially multi-stakeholder, and not focused on a single operator. In the network continuum, each operator would be just another operator part of that continuum, and the orchestration of services is proposed considering the infrastructure resources of the continuum as a whole, and not just those of a single operator.

Applying network slicing in scenarios where slices extend across infrastructure managed by multiple stakeholders presents challenges, but it is actually considered with different approaches. E.g., the multi-domain orchestration approach considers coordinating slices across different operator domains relying on a unified orchestration layer. In this regard, the ITU-T Recommendation [Y.3182] provides guidance on using machine learning for end-to-end multi-domain slice management, enabling consistent QoS and dynamic adjustments across stakeholders. Also, the federated resource control approach also considers managing slices spanning multiple domains, such as those described in [LHX+20] [AM21] [FHS18], with focus on federated approaches and enabling resources to be shared while maintaining autonomy for each stakeholder's infrastructure. Also, the usage of smart contracts and SLA automation has been proposed, e.g., in the 5GZORRO project [5GZ21], which proposes to integrate blockchain-based smart contracts to manage service-level agreements (SLAs) across stakeholders to ensures trust, transparency, and automation in managing slice resources and monitoring compliance with agreed performance levels, even in complex multi-operator setups.

What the decentralised M&O approach proposes for aligning with the network slicing concept is represented in Figure 5-40, still in line with the main abstractions in [NGM16] represented in Figure 5-39.

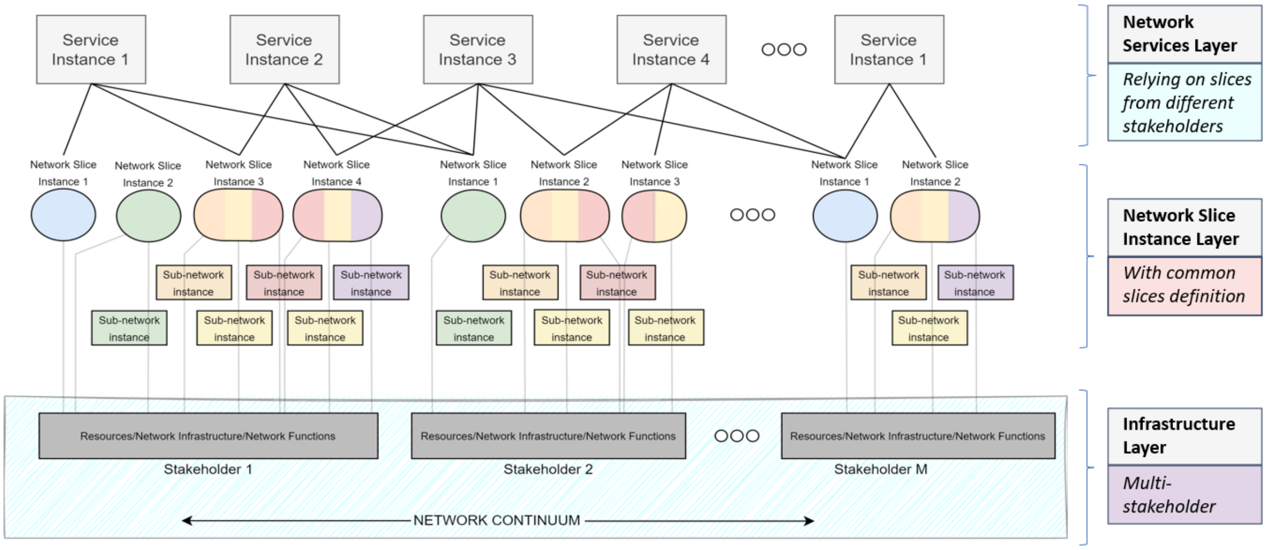
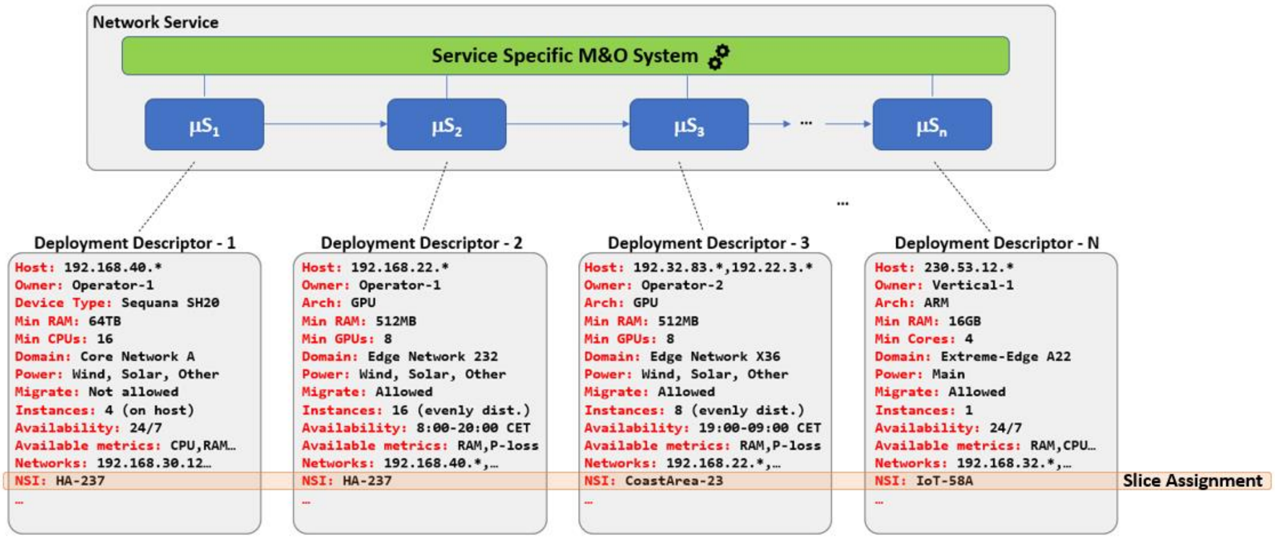


Figure 5-40: Alignment of the Network Slicing concept with the Decentralised M&O approach

As it can be appreciated in this case the Infrastructure Layer is explicitly multi-stakeholder, in line with the network continuum concept, i.e., the infrastructure layer to define the network slices are all the network infrastructure resources provided by the different stakeholders participating in the continuum. However, as it can be seen, the network slice instances are still defined per-stakeholder, i.e., it is the stakeholders participating in the continuum which, in their scope, decide which slices would be defined. However, as shown in the figure, this definition would be made according to a common slices definition schema that would be aligned among the different participants in the continuum (e.g., different stakeholders could offer IoT or low-latency slices based on their resources and capabilities, however, their slice instances would be labelled the same way, following a shared multi-stakeholder definition schema). This is represented in the figure by the different colours of the blocks representing the slice instances, which, as can be appreciated, can be repeated among the different stakeholders. Finally, the service layer on top represents the network services that would be deployed over the slices offered by the different stakeholders. As it can be seen, services can rely on the same slice types that could be implemented on infrastructure resources belonging to different stakeholders. In fact, that common slices definition schema referred before is what makes possible that a network service could rely on infrastructure resources provided by different stakeholders.

In practice, and in line with the concepts explained in [HEX224-D63] and [HEX224-D33] in relation to the service deployment process in the decentralized M&O model, the definition of the slice to which each service would be assigned would be done at the level of each microservice, and specifically, in their deployment descriptors. These deployment descriptors would refer the name of the slice instance in the common definition schema mentioned above. Figure -5-41 illustrates this concept. As it can be appreciated, since the slice assignment is performed per-microservice, a whole network service could be deployed on different slice types defined by different stakeholders in the continuum, relying on the microservices federation approach. Also, based on this, and on the highly distributed nature and heterogeneity of the extreme-edge domain, this approach could leverage the deployment of a diverse and rich slices ecosystem, in line with the specific infrastructure resources provided by the different stakeholders, e.g., depending on their geographical location, access to certain types of infrastructure resources (IoT devices, industrial devices, residential devices, etc.), or their mobility patterns, among others. Figure -5-41 illustrates this showcasing an example in which the different microservices composing the network service are deployed referring different network slice instances (NSI): a High Availability slice called “HA-237” in an MNO scope referred as “Operator-1”, the infrastructure in a geographically relevant instance (“CostArea-23”) provided by another “Operator-2”, and certain IoT infrastructure resources on a so-called “IoT-58A” instance provided by a vertical industry “Vertical-1”.



Deployment Descriptor - 1
Host: 192.168.40.*
Owner: Operator-1
Device Type: Sequana SH20
Min RAM: 64TB
Min CPUs: 16
Domain: Core Network A
Power: Wind, Solar, Other
Migrate: Not allowed
Instances: 4 (on host)
Availability: 24/7
Available metrics: CPU,RAM…
Networks: 192.168.30.12…
NSI: HA-237
…

Deployment Descriptor - 2
Host: 192.168.22.*
Owner: Operator-1
Arch: GPU
Min RAM: 512MB
Min GPUs: 8
Domain: Edge Network 232
Power: Wind, Solar, Other
Migrate: Allowed
Instances: 16 (evenly dist.)
Availability: 8:00-20:00 CET
Available metrics: RAM,P-loss
Networks: 192.168.40.*,…
NSI: HA-237
…

Deployment Descriptor - 3
Host: 192.32.83.*,192.22.3.*
Owner: Operator-2
Arch: GPU
Min RAM: 512MB
Min GPUs: 8
Domain: Edge Network X36
Power: Wind, Solar, Other
Migrate: Allowed
Instances: 8 (evenly dist.)
Availability: 19:00-09:00 CET
Available metrics: RAM,P-loss
Networks: 192.168.22.*,…
NSI: CoastArea-23
…

Deployment Descriptor - N
Host: 230.53.12.*
Owner: Vertical-1
Arch: ARM
Min RAM: 16GB
Min Cores: 4
Domain: Extreme-Edge A22
Power: Main
Migrate: Allowed
Instances: 1
Availability: 24/7
Available metrics: RAM,CPU…
Networks: 192.168.32.*,…
NSI: IoT-58A
…

Figure -5-41: Per-microservice slice definition

Of course, and in line with the Decentralised M&O concept described in Section 5.4.1.3, the network element in charge of solving the deployment on those specific instances would be the so-called Deployment Node (DN) with the support of the Infrastructure Registry (updated by the IRNs), on which each of these NSIs would be declared.

To summarise, the approach described here, aligned with the concept of the decentralised orchestration, enables networks to be efficiently shared and managed among multiple stakeholders (e.g., operators, enterprises, service providers, and even end users) without compromising service quality. This is particularly pertinent for future 6G networks, where the diverse use cases envisaged demand collaborative and flexible management. The model still retains benefits associated with 5G network slicing (logical isolation, tailored slices, efficient resource sharing, SLA-based collaboration, etc.), but it also provides:

- Logical isolation for stakeholders allowing customised slice definitions within their own networks, but at the same time, allowing them to share their slices in the network continuum.
- Enhanced operational efficiency, enabling a better utilisation of the existing infrastructure in the whole network continuum, so reducing costs and improving sustainability.
- Promotion of innovation by supporting very specialised slices by integrating extreme-edge infrastructure resources, so leveraging new services and business models.

5.4.2.5 Summary of the slicing and orchestration enabler

Table 5-9 presents the summary and key take-aways for the slicing and orchestration enabler.

Table 5-9 Summary of the slicing and orchestration enabler

Description	In this enabler, the intent-based control of 6G slices are investigated. More specifically, this enabler presents (1) 6G network slicing, (2) Intent based management and (3) Envisioned impact of intent-based management of 6G slices
Key take-aways	5G and 6G slicing should co-exist since we cannot ensure 6G coverage everywhere. Similar to 5G, using the same the S-NSSAI structure to identify a 6G Slice ensures smooth transition to 6G.5G S-NSSAI structure is sufficient. Current S-NSSAI structure contains 32 bits that can accommodate quite a large number of slice IDs thus no issue is foreseen for 6G slicing. Existing analysis does not indicate any need to extend the S-NSSAI structure; this ensures also backward compatibility with 5G slicing. Intent based orchestration enables faster onboarding of network functions to production including provisioning of underlying cloud infrastructure with a true cloud native approach. It also reduces the costs of adoption of cloud and network infrastructure and manages a huge number of clusters of servers across the telco network, handling a variety of infrastructure technologies with a uniform and consistent user experience, automatically installing and configuring additional plugins.

5.5 Quantum-enhanced network functionalities

5.5.1 Introduction

Quantum technology, characterized by its significant yet subtle versatility, is increasingly becoming integral to numerous organized scientific initiatives aimed at enhancing contemporary technological landscapes, encompassing areas such as computing, communication, and information science. Whether the emphasis in 6G communications is on the IoT or its forthcoming iteration known as the tactile internet, contemporary technologies may pave the way to meet specific KPIs without the typical compromises associated with conventional technology solutions. The integration of quantum technologies in advancing 6G networks focuses on enhancing trustworthiness (see appendix), synchronization, and communication efficiency. A critical analysis highlights the unique strengths and challenges posed by quantum advancements.

An examination of how trustworthiness, encompassing privacy, security, and reliability, can be quantitatively assessed in dynamic network environments is provided (see Annex for further details). Privacy is positioned as a pivotal factor in fostering trust, directly supporting compliance with upcoming 6G standards. The analysis connects secure quantum communication mechanisms, like satellite Quantum Key Distribution (QKD), with improved network resilience, particularly against vulnerabilities such as man-in-the-middle attacks. This concept, along with its associated principles, see Figure 5-42, provides a valuable framework for quantitatively assessing dynamic network performance as parameters evolve over time, while also relating to the network's capacity to withstand various perturbations, thereby indicating its resilience to challenges and disruptions. Consequently, trustworthiness is anticipated to serve as a crucial KPI for effective 6G network implementations, contributing to a heightened quantitative emphasis on resilience in future xURLLC communications.

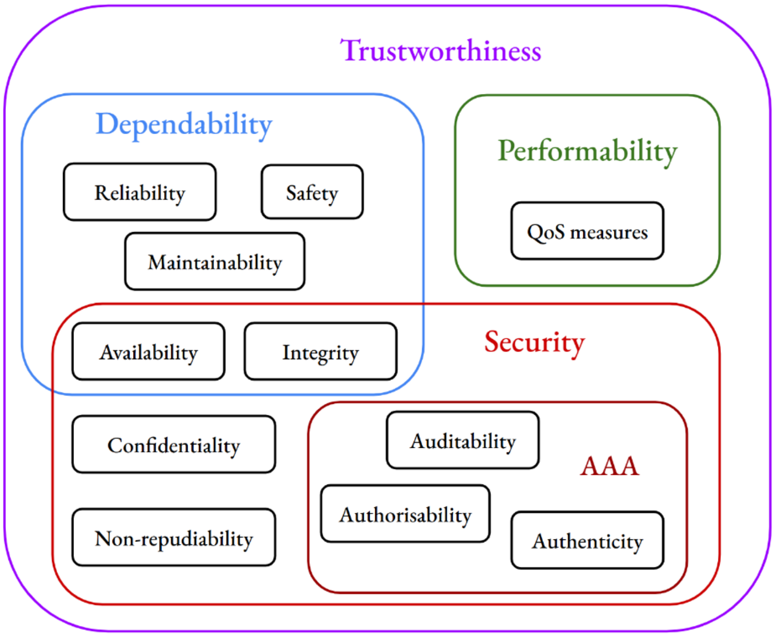


Figure 5-42 The concepts encompassed by trustworthiness in network contexts [RBF24].

Quantum synchronization through Time-Correlated Entangled Photons (TCEP) is analyzed as a revolutionary solution for achieving sub-nanosecond precision. This innovation addresses the limitations of traditional protocols like PTP by ensuring scalability, resilience to environmental disturbances, and decentralization. Experimental validation demonstrates TCEP's capability to synchronize distributed quantum nodes with picosecond-level accuracy, positioning it as a critical enabler for Ultra-Reliable Low-Latency Communications (URLLC), quantum computing, and secure communication frameworks.

Further**,** quantum-semantic communication framework that integrates semantic encoding and decoding techniques with quantum channel encoding is analyzed. By leveraging knowledge graphs and deep learning-based decoders, the analysis demonstrates improvements in communication efficiency and reliability.

Comparative assessments reveal significant computational advantages and fidelity gains over traditional Shannon-based methods, supporting 6G's demand for scalable, low-latency solutions.

5.5.2 Quantum Synchronization

Quantum synchronization is an innovative technology to achieve highly precise timing in distributed communication systems. As next-generation networks continue to develop, applications like URLLC, extended reality (XR), and quantum communication require synchronization accuracy that exceeds the capabilities of classical methods. Traditional synchronization methods, such as the Network Time Protocol (NTP) and Precision Time Protocol (PTP), have advanced significantly. While PTP achieves sub-nanosecond precision through hardware timestamping, it faces challenges such as network-induced jitter, asymmetric delays, and scalability issues in large-scale quantum systems [NGB+24, EFW02]. Time-Correlated Entangled Photons (TCEP) is a synchronization solution that has the potential to achieve sub-nanosecond (<100 ps) precision while also being scalable and resilient to environmental disturbances [HEX224-D33].

TCEP provides a quantum-based synchronization technique that leverages the temporal correlations inherent in entangled photon pairs. The theoretical groundwork laid by Jozsa et al. [JAD+00] and Giovannetti et al. [GLM01] demonstrated that quantum entanglement could surpass the precision limits of classical methods. Building on these insights, Valencia et al. [VSS04] experimentally validated TCEP synchronization using Spontaneous Parametric Down-Conversion (SPDC), achieving picosecond-level accuracy without centralized timing references. TCEP synchronization has shown resilience to environmental disturbances and strong scalability for distributed quantum networks, making it an ideal candidate for next-generation quantum-enabled applications [HEX224-D33, STS+22].

5.5.2.1 Time-Correlated Entangled Photons (TCEP)

TCEP, is the backbone of quantum synchronization, leveraging temporal correlations between entangled photon pairs generated through Spontaneous Parametric Down-Conversion (SPDC). These photon pairs serve as the quantum resource for measuring timing offsets between two nodes, Alice and Bob [NGB+24].

SPDC produces entangled photon pairs by splitting high-energy pump photons into two lower-energy photons, called signal and idler photons, while conserving energy and momentum. This process ensures that the generated photon pairs are temporally correlated, which is essential for synchronization. The arrival times of these entangled photons at Alice and Bob are inherently linked, allowing for the precise determination of the relative clock offset between the two nodes.

TCEP provides several advantages. It achieves sub-nanosecond timing precision (less than 100 picoseconds) due to strong temporal correlations. It enables decentralized synchronization, removing the need for central references such as GPS. Additionally, TCEP is robust against noise and asymmetries in the optical path, making it reliable in various operational conditions. These features position TCEP as a critical enabler for advanced quantum synchronization systems.

Experiments with SPDC-generated entangled photons have validated the cross-correlation function as a reliable tool for determining synchronization offsets between distant nodes. Timestamps recorded with picosecond precision by TDCs are analyzed to compute the cross-correlation function $C_{AB}(\tau)$, which identifies the clock offset through its sharp peak, see Figure 5-43. Gaussian or Lorentzian fits to this peak enhance precision, with observed timing jitter consistently below 100 ps [NGB+24]. Real-time FPGA integration ensures efficient computation, handling large datasets and high photon detection rates, making the system scalable and robust. This experimental framework demonstrates the practicality of TCEP-based synchronization for advanced quantum communication, achieving precise and scalable clock alignment essential for quantum networks and next-generation applications such as QKD and distributed quantum computing. The sharp peak at the synchronization offset, with a Full Width at Half Maximum (FWHM) of less than 100 ps, demonstrates the sub-nanosecond precision achieved using TCEP-based synchronization. This result confirms the robustness of the FPGA implementation for real-time photon timestamp processing and synchronization.

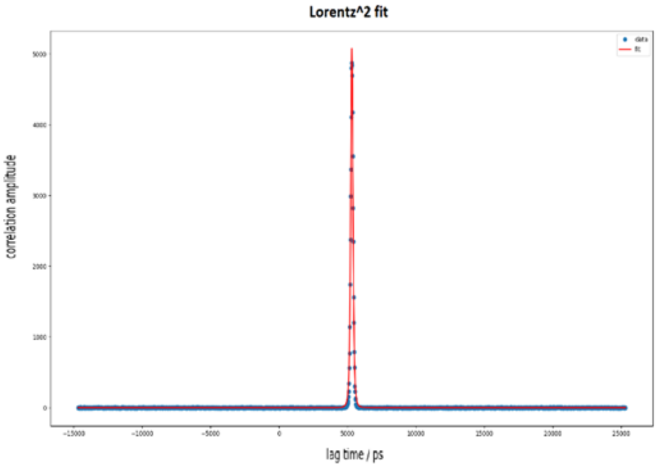


Figure 5-43 Experimental cross-correlation function between photon arrival times at two nodes (Alice and Bob).

In this context, TCEP-Based Synchronization offers several transformative applications in 6G networks. In URLLC, sub-nanosecond precision enables real-time navigation, extended reality (XR), and industrial automation. It plays a critical role in Quantum Key Distribution (QKD), ensuring secure communication through accurate clock alignment. High-precision synchronization facilitates advanced localization and navigation capabilities. In distributed quantum computing, quantum synchronization ensures efficient node synchronization for seamless operations. Additionally, it supports intent-based orchestration by dynamically optimizing resource allocation using precise timing.

The deployment if the TCEP is faced by environmental sensitivity challenges, such as variations in optical paths due to temperature changes or vibrations, and the significant cost and infrastructure required for large-scale implementation. Photon loss is another critical challenge, emphasizing the need to minimize transmission losses for consistent and reliable performance.

Further research is needed to enhance photon detection by improving detection efficiency and reducing timing jitter in SNSPDs. Long-distance networks will focus on implementing dispersion compensation to enable scalability. Expanding to dynamic multi-node networks with machine learning integration is another key area of development. Additionally, the development of hybrid systems will involve creating protocols to seamlessly integrate quantum and classical synchronization methods.

5.5.3 Quantum Semantic Communication

Recent advancements in machine learning (ML) and generative artificial intelligence (AI) have led to the exploration of the semantic communication paradigm as a way to overcome Shannon information's theory bottlenecks in supporting next-generation networks. Unlike Shannon's traditional communication model [SW49], which focuses solely on the transmission of bits without considering the meaning of the data [SB21], [MWQ+23], semantic communication facilitates communication that conveys both semantic content and context, offering the potential for higher data transmission rates that surpass conventional channel capacity limits [BBD+11].

However, developing an innovative semantic communication scheme poses challenges when transmitting the semantic-based systems (such as knowledge graphs) over a traditional communication channel due to the scalability of computational and associated resource costs [WN23]. The proposed quantum-semantic communication framework not only overcomes these barriers but also enhances cloud-native systems by enabling efficient, context-aware communication through metrics like quantum semantic fidelity and F1 score. Simulation results show significant improvements when compared to existing semantic communication methods and traditional Shannon-based encoding techniques, underscoring its transformative potential for cloud-centric and hybrid quantum-classical networks.

Performance analysis of quantum semantic

The scalability analysis in Figure 5-44 (a). illustrates the trade-offs when considering the details of quantum superconducting hardware. The average fidelity serves as a strict upper bound, even with increasing complexity of semantic encoding. On the other hand, the red curve shows quadratically increasing number of required quantum resources (for the communication task), showcasing the advantage compared to traditional communication scenarios, which grow exponentially for this case [NHS+24].

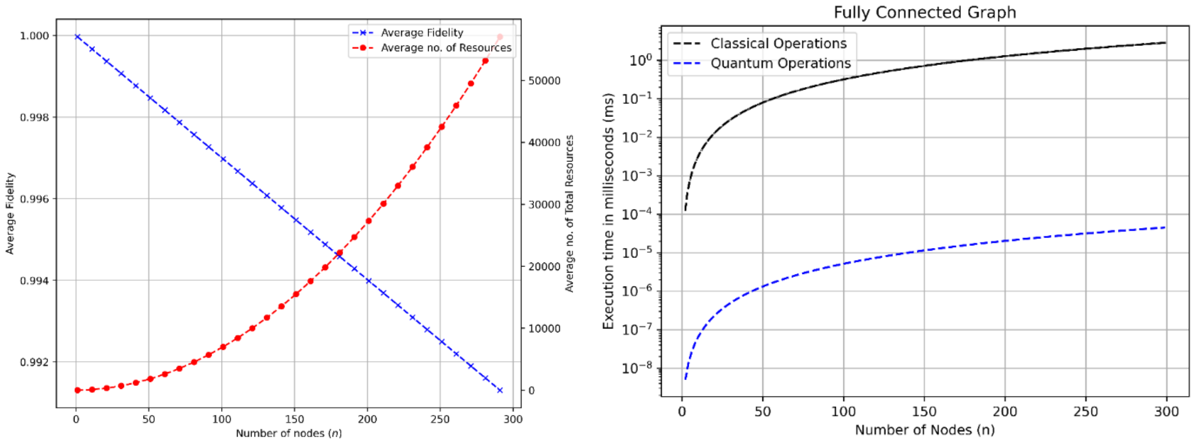


Figure 5-44 (left) Scalability of the quantum semantic communication approach, (right) The trade-offs between increasing complexity of KGs vs computational complexity (plotted in log scale)

A scalability analysis demonstrates a 'quartic polynomial' gain over conventional methods for computing complex semantic systems, as shown in in Figure 5-44 (b). Additionally, this computational complexity contributes to processing delays in conventional communication methods at the physical and link layers, affecting latency [NSB+24].

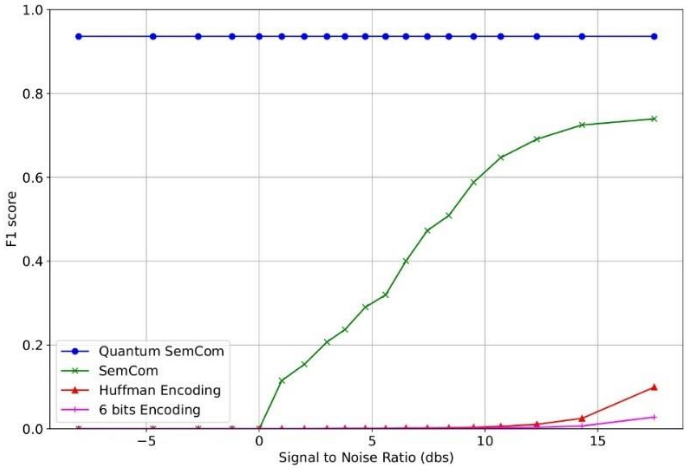


Figure 5-45 Performance evaluation of quantum semantic communication [NHS+24a].

Following the evaluation of quantum semantic fidelity, a comparative analysis between the quantum semantic communication approach and traditional encoding methods (Huffman, and 6-bits encoding) over AWGN channel is shown in Figure 5-45. This comparison highlights the advantage of our approach in terms of communication efficiency: Please refer to the appendix for extended results of the comparative analysis.

5.5.4 Summary of the quantum enabler

Table 5-10 summarizes the quantum enabler.

Table 5-10 Summary of the quantum enabler

Description	The quantum enabler integrates TCEP synchronization, quantum resilience, and quantum semantic communication into 6G networks.
Key take-aways	Quantum synchronization and quantum semantic communication introduce foundational capabilities for 6G networks, enabling ultra-precise timing and context-aware communication. The integration of Time-Correlated Entangled Photons (TCEP) ensures sub-nanosecond synchronization, essential for latency-sensitive applications like URLLC and distributed quantum computing. Quantum semantic communication provides a scalable approach to overcome Shannon-based limitations, enhancing the efficiency of knowledge representation and data exchange. Both services support seamless integration with cloud-native architectures, leveraging intent-based orchestration to dynamically optimize resource allocation while maintaining backward compatibility with hybrid quantum-classical systems. Challenges such as environmental sensitivity and photon loss are being addressed through advancements in hybrid protocols and machine learning-driven optimizations.

5.6 Fulfilment of the transformed architecture objectives

Chapter 5 directly contributes to Objective WPO3.2 by defining and analyzing solutions that integrate the flexibility of cloud technology with distributed processing nodes into self-contained modules. These modules are designed to have minimal dependencies, enabling stepwise network expansion and scalable deployments.

Section 5.2 defines the concept of self-contained modules with minimal dependencies, outlining their role in extending and scaling network deployments incrementally. It identifies the key design aspects of such modules, considering various deployment scenarios, including their integration within the Radio Access Network (RAN).

Section 5.3 analyzes different architectural solutions that leverage cloud technology’s flexibility in combination with distributed processing nodes. This analysis highlights how different configurations impact scalability, modularity, and network performance.

Section 5.4 builds on these findings by addressing the need for a flexible management and orchestration framework capable of handling the diverse module configurations and deployment scenarios within the cloud continuum.

Section 5.5 focuses on synchronization challenges in distributed architectures and presents a quantum-based solution to enhance precision and reliability in network operations.

These sections contribute to defining scalable, modular network solutions that integrate cloud flexibility with distributed processing, ensuring seamless network expansion with minimal dependencies.

6 Quantitative targets

The Hexa-X-II project objectives are defined by Hexa-X-II from the beginning of the project, and the work packages were assigned to investigated certain objectives that fit the work package description. For each objective, there are one or more so called Quantified Targets (QTs), which the work packages should try to fulfil. WP3 were assigned objective 3,4 and 5, see Table A-5. More details on how the quantified targets are fulfilled are shown in Annex B.

Objective 3: service coverage and low latency support.

Coverage is one of the main KPIs for 6G and a main KPIs for 6G and also important aspect for, identified 6G use cases [HEX223-D12]. The objective stipulates enabling more that 99% of the global population and more than 99% of the world area can be provided for at least one basic 6G use case, when and where needed, at sustainable cost levels. This means that connectivity shall be provided not only to urban areas but also to remote areas, relying on both TN and NTN. To address this target, several investigations have been performed, see Section B.1.

To investigate the coverage, a novel *coverage inequality index* is developed. The purpose is to quantify the fairness between urban and rural coverage and identify which identify how the networks should be adapted to improve the essential service coverage in remote areas. This coverage index takes two maps of a larger region (e.g. country, or region of validity of a spectrum license) as its input, one is a rurality map (see Section B.2), and the other is a TN/NTN coverage map.

The user throughput performance for a LEO satellite scenario has also been investigated. The simulations are performed for different number of users per beam. Using a bandwidth of 30 MHz, the DL user throughput for one user per beam is around 15 Mbps. The UL throughput is noticeably lower compared to the DL due to the lower transmit power of the UEs with a user throughput of around 0.2 and 0.4 Mbps.

To increase the coverage, a flexible network with autonomous UAVs has been developed, using dynamically deployed UAVs as relay nodes and access nodes. The simulations show that introducing a number of UAVs in an area lacking coverage, can achieve 100% coverage with an optimal number of drones, significantly reducing deployment and maintenance costs compared to static infrastructures.

As a summary, the investigations here shows that it is possible fulfill the coverage target of >99% of the global population and (>99%) of the world area. The 6G use case for this is the ubiquitous network use case [HEX223-D21], which stipulates a user experience of around 0.1 to 25 Mbps. The target also stipulates this has to be at a sustainable cost, this is treated in some more detail in [HEX22-D53] for the NTN coverage.

The second part of objective 3 is about low latency support and how the network can support less than 1 ms E2E and less than 0.1 ms in critical subnetworks). The focus here is on the <1 ms E2E latency for the user plane, but also addressing the latency for the control plane

To improve the core network latency, a data-centric Service-Based Architecture (SBA) with higher decomposition of core Network Functions (NFs) has been developed (see Section 5.2.2.4). The results emphasize the performance benefits of redesigning the network modules interactions based on DataFlow programming. The workflow completion time, depending on the protocol of choice, may be reduced by up to an order of magnitude.

The user plane latency is estimated following the methodology presented in [HEX23-D53]. The method includes the air interface delay (Uu) as well as the processing delays for each layer in the UE protocol stack, in the RU, in the gNB layers, and in the UPF. The transport delay when using fiber over 50 km distance to connect the RAN and core network domains as well as the ethernet switching delays is calculated to be equal to 196 µs. Accordingly, the total E2E delay is equal to 380 µs (excluding an additional delay for the server access).

The latency target of <1ms is evaluated and is possible to achieve for some scenarios, i.e., very fast Uu interface (<0.1 ms, see [HEX23-D23]), relative short (<50 km) and fast backhaul links (fiber), and fast processing per protocol layer (<8 µs, see [HEX23-D23]).

Objective 4: 20% performance increase using sensing, AI etc

The quantified target of objective 4 is to have at least 20% improvement in performance in at least one of energy efficiency, latency, bit rate or area capacity through use of sensing, localisation, traffic data, or mobility patterns for AI-based optimisation in selected use cases.

One main approach for fulfilling this target is to change the placement or offload functions, applications or services within the network, e.g, to move certain functions closer to the UE. This can be based e.g. on latency, energy efficiency, capacity and traffic data. The Integrated Network and Compute (INC) solution aims to jointly optimize network performance and compute processes in a way that meet application latency requirements (see section 3.4.1 for more details). The INC approach outperforms other solutions investigated and can achieve more than 35% enhancement of the number of satisfied latency requests. Another method is to offload certain processes or functions from a device to the network, to save energy in the device and also to reduce complexity of the device but still be able to run high demanding applications. In [HEX224-D33], a dynamic device offloading experiment was performed which showed that the the power consumption of the device could be reduced by approximately 50% during offloading (to the expense of more usage of the radio interface and network compute). Another developed method is the BCS consumer application placement in 6G networks. The genetic algorithm developed enhances overall network performance by strategically leveraging network traffic data. Simulations show a 50% decrease in end-to-end latency and a 55% reduction in power consumption for the genetic algorithm compared to a greedy placement strategy.

Another approach is to use AI and split learning for energy savings by offloading some functionality to the network. The evaluations shows that the offloading provides a 73% reduction in the device energy consumption at the application level.

As a summary, the target to energy and latency improvement the performance with >20% using AI, offloading, placement of functions etc, is fulfilled.

Object 5: (>25%) reduction in OPEX by using zero-touch automation.

This section describes how to achieve the quantified target for objective 5: (>25%) reduction in operating expenses (OPEX) by using zero-touch automation. One solution to optimize OPEX is using Wireless Hierarchical Federated Learning to minimize the transmitted power. The solution achieves up to 90% reduction in energy consumption, which can be seen as an OPEX reduction. Another solution here is to optimize the usage of the cloud resources. Several proposals are outlined in this deliverable (see section 5.3 for more details). E.g., managing multi-cluster resources, decentralized orchestration, and constrained devices in ETSI MEC scenarios. Quantitative studies show that resource optimization techniques in distributed cloud architectures can reduce energy consumption by 30% to 40%.

7 Summary and Conclusions

This deliverable describes the necessary components for a 6G architecture supporting enhanced mobile communication as well as new beyond communication services, such as sensing, compute offloading and AI. Further on, the deliverable presents a 6G architecture that is more flexible compared to previous generations, in order to support different types of network deployments, such as D-MIMO networks, local mesh ad-hoc networks, and satellite support. Several solutions are focused on how to make the architecture more scalable to support current and varying needs, to improve both sustainability and resource efficiency. This implies that the architecture should be cloud-native, modular and easily extendable and be able to scale down resources and modules when needed, while ensuring that the complexity remains manageable. The main goal of this deliverable is to analyse the enablers that support such a 6G architecture.

To achieve the first objective “6G architecture for AI and beyond communications” (WPO3.1), several enablers have been identified for the data-driven architecture, comprising DataOps, MLOps, and AIaaS. The proposed architecture is based on newly introduced or extended functions, such as the DataOps for handling data lifecycle tasks, the MLOps for end-to-end AI workflow management, the AIaaS for exposing AI functionalities. The deliverable describes a technical analysis focusing on issues of data exposure and quality assurance, AI model lifecycle management and privacy-preserving learning methods, distributed computing orchestration and failure prediction strategies. This deliverable also details how these enablers work together in the 6G E2E system blueprint. The AIaaS Framework includes the AI Orchestrator, AI/ML Catalogue, the MLOps Orchestrator and relies on that the DataOps delivers the training data. The AI Orchestrator manages AI service deployment, execution, and lifecycle management, allowing AI and ML models to optimize network traffic, predictive maintenance, and resource allocation. The AI/ML Catalogue serves as a repository of pre-trained models, algorithms and supports model versioning.

In a similar manner for other beyond communication services, several enablers have been defined already from the start. These enablers are ISAC, compute offloading and optimized application placement. The ISAC enabler defines the necessary architecture framework for 6G sensing. New functionality is needed to handle sensing requests, both in the form of an application that sends a sensing request and in the form of a function that receives and processes the request. A sensing management function is thereafter needed to further make use of the information in the request and apply the request information to control and configure usable sensing units (e.g. sensing units located in the correct places). There must also be suitable forms of exposure, and (radio) resources needed for the actual sensing measurements. A compute offloading architecture framework is defined in this deliverable. The compute offloading architecture framework is loosely integrated to the cellular network with no changes to RAN and NAS protocols, thus making it less disruptive to existing cellular standards than a solution using e.g., the RRC protocol to handle compute offloading. This allows for faster time to market realization and easier integration with already existing offloading resources. Note that it the framework still requires new procedures to interact with the network, for instance through AF via NEF. Several of these procedures are detailed in this deliverable, for example on how to handle the offload node discovery, node registration and offload procedures.

The next objective WPO3.3 “Architecture for flexible topologies” concerns solutions related to new access and flexible topologies. This deliverable describes including multi-connectivity solutions, as well as control and management solutions for programmable and context-aware transport. NTN and trustworthy flexible topologies are introduced as part of the network of networks enabler, which includes NTN architectural options and TN-NTN integration, subnetworks with new node roles (MgtN) and node coordination via various architectural options. Regarding the multi-connectivity solutions, the proposal is to have only one architectural option for 6G. This option should be based on evolved CA with improvements such as cell recovery mechanism described in the deliverable. Solutions for optimizing the use of the transport network infrastructure and packet switching are presented as part of the context-aware management of transport resources. Additionally, flexible allocation of edge resources and computation offloading decision-making across different computing options are proposed to adapt the compute infrastructure based on the context.

The last objective is WPO3.2 “Combine the cloud technology for a modular, scalable and extendable architecture “.In general, it is found that there is a trade-off between performance and flexibility when considering the design of a modular 6G architecture: More granular design results in higher flexibility in implementing and deploying modules but at the cost of reduced performance in terms of execution time and state management. Furthermore, procedure-based functional decomposition of the CN control plane is evaluated. The analysis shows that there is a reduction in the total number of messages needed for these procedures by using procedure-based functions. However, this design reduces the flexibility in deploying the more coarse-grained NFs.

There is a need to improve the cloud friendliness of the interface between RAN and the CN (NGAP/N2 interface). For this to happen, the SCTP protocol used in 5G needs to be evolved or replaced to support better decoupling between the different layers and avoid so called transport bindings, etc. It is found that using a Data-Centric Networking solution for the CN can enable enhanced scalability and flexibility through dynamic stateless NFs and simplified architecture with efficient resource management. Different RAN architectures have an impact on the performance for D-MIMO, indicating that a 6G RAN architecture with lower split may be suitable for D-MIMO operations. The deliverable also analyses the outcome of the 4G to 5G migration, and discusses the lessons learned and how to apply them to the 5G to 6G migration. One proposal is to use spectrum sharing for the 5G and 6G integration and the core network should be based on an evolved 5GC.

The cloud transformation for 6G includes how to evolve the cloud to be multi-cloud federation enabled. It is shown that a multi-cloud leads to higher resource availability and flexibility, which also lowers the total network load and latency times in service deployment, as well the ability to leverage computing power and storage that is superior to current solutions. There should be a unified and abstract interface for compute continuum resource management (inventory, provision, operate). 5G and 6G slicing should co-exist since 6G coverage cannot be ensured everywhere. Similar to 5G, using the same the S-NSSAI structure to identify a 6G slice ensures a smooth transition to 6G.

In addition to this, the deliverable has also investigated how to achieve several of the so-called quantified targets. The (user plane) latency target of 1 ms can be achieved using an optimal deployment. Furthermore, it is shown that the coverage target of >99% of the earth area for an essential service coverage of a basic 6G service can be supported (depending on bandwidth).

The deliverable also includes a detailed description of the two proof-of-concepts (PoCs) developed by WP3. The first PoC is called “Distributed ML model training and inference” for a remote-controlled robot use case. With cross-network function training, it demonstrates collaborative training without moving and sharing data between the entities. The cross-network function training enables an ML model to train with richer input obtained from different layers in the network stack. This way, the efficacy of the ML model can be improved. The second PoC “Trustworthy flexible topologies in 6G, leveraging on beyond communication aspects” investigates means for a network that enables versatile, robust and dynamic applications not only intended towards public use but also for industry needs, demonstrating the adaptability and resilience of the 6G network in a warehouse scenario.

7.1 Detailed summary of the enablers

Table 7-1 lists all enablers developed and presented in this deliverable, with a short background and key take-aways.

Table 7-1 Summary of the Key take-aways of the enablers

Enabler	Background	Key take-aways
DataOps	Data shall be delivered, pre-processed, and stored where and when required. This imposes requirements on a flexible data ingestion architecture.	A framework is presented for providing data in a heterogeneous 6G network slicing environment. Alignment with 3GPP and Figure 3-2 standards ensures interoperability and adoption of standardized data operation practices. DataOps can be used to gain insights of the 5G/6G Core network performance, by supplying Machine Learning models with data necessary for Failure Prediction

MLOps	MLOps offers a set of tools for efficient ML model lifecycle management, with a particular focus on distributed AI/ML functions in 6G networks.	ML model layer offloading allows efficiently utilizing resources across the computing continuum while minimizing on-device energy consumption as well as minimizing overall (network and on-device) energy consumption with increasing number of devices and applications. Due to measured model fitting time and inference time for different ML algorithms, and the possibility of implementation of ML algorithms in AIaaS, regression trees for failure prediction in 5G/6G Core were selected.
AIaaS	AIaaS is a framework that offers a wide range of AI services as well as personalized inference capabilities to the AI service itself	AIaaS Framework, encompassing tools like the AI Orchestrator, AI/ML Catalogue, and MLOps Orchestrator. The AI Orchestrator manages AI service deployment, execution, and lifecycle, optimizing network traffic, predictive maintenance, and resource allocation. The AI/ML Catalogue serves as a repository of pre-trained models, algorithms, and metadata, supporting model versioning, interoperability, and seamless integration into network environments. The MLOps Orchestrator oversees model lifecycle management, handling deployment, retraining, and adaptation to ensure sustained relevance and accuracy.
ISAC	The ISAC enabler encompasses architecture enhancements and protocols to integrate sensing services in a communication system	New functionality is needed to handle sensing requests, both in the form of an application that sends a sensing request and in the form of a function that receives and processes the request to further make use of the information in the request (the SeMF) and transform the request information into usable sensing units (e.g. units located in the correct places), suitable forms of exposure, (radio) resources needed for the actual sensing measurements. It is also necessary for a function that process of the measurement data and provide output in a format useful for the requester. It is also found that existing means for data exposure may need to be enhanced to support the sensing measurement data in different scenarios. Overhead in terms of radio resources from ISAC appears to be small, at least for the bi-static case when UE transmits sensing signals to the gNB.
Compute offloading	Future 6G networks will consider computing capabilities across the full network, from the extreme edge (including the UE) to Telco grade clouds (CC, Compute Continuum	A compute offloading architecture framework can be integrated in a cellular network with no significant changes to RAN and NAS protocols, thus making it less disruptive to existing cellular standards, allowing for faster time to market realization. In addition, several procedures are detailed, for example on how to handle the offload node discovery, node registration and offload procedures.
Optimized application Placement enabler	The optimized application placement procedure ensures optimal deployment of applications by balancing latency, energy consumption, and resource availability.	For smaller numbers of requests, Proximity-to-UE and INC provide similar results (edge-clouds that are in the proximity to the UE can accommodate all the requirements). As the number of formulated requests increases, the proposed INC approach outperforms the Proximity-to-UE solution (nearest edge-clouds cannot satisfy all the requirements, and a global optimization that takes into account both network and compute requirements is needed)
Multi-connectivity	Multi-connectivity enables multiple frequency ranges by different physically separated nodes, the aggregation of different radio access technologies, carriers, and access networks	Only one architectural option based on CA should be defined in the specifications. Proposed CA improvements inspired by the DC advantages and vice versa. New procedure proposed in this deliverable: PCell recovery via SCell solution. Coverage extension or increased reliability via WLAN-based UE relay for remote UEs.

		Carrier aggregation simulations show most gains when the aggregated connections have similar performance in terms of coverage and capacity. Scenarios where the PCC has much higher BW than the SCC do not lead to any significant gains in terms of capacity when CA is activated. A possible deployment for 6G MC is to use radio units connected to the same gNB using LLS, where the RUs are not co-located. The application of the RSMA technique for multi-server offloading enables optimized power and rate allocations, effectively meeting stricter QoS requirements for users compared to the conventional power-domain NOMA technique.
Network of networks	Integration of multiple subnetworks, including terrestrial and non-terrestrial networks in order to create a seamless and ubiquitous communication system	**NTN**: The most likely options for 6G NTN architecture are “RU on board” or “gNB on board”. NTN simulations in section B.1 shows a normalised beam throughput of around 0.5 bps/Hz in DL and 0.015 bps/Hz in UL which may be acceptable for basic 6G services. A new index for quantifying coverage inequality is proposed, which can be used for network planning and by regulators. A TN-NTN fast switch connectivity procedure should be introduced to switch faster to NTN in the cases where there is no good TN coverage. **Trustworthy flexible topologies:** Subnetworks: Complimenting the transparent and non-transparent architectures, the MgtN-SA, BS-MgtN DC, BS-SA with MgtN support architectures are described. A new lightweight subnetwork CP between the MgtN and the UEs in a subnetwork can be introduced. Procedures such as “RRC configuration with the aid of the MgtN”, which take advantage of the snCP, should be specified. Reliable network deployment in scenarios like disaster recovery and outdoor events through UAV-assisted flexible topologies. Seamless Network Integration: Interaction with functions such as NEF, AMF, and PCF for authentication, resource allocation, and mobility management, ensuring service continuity.
E2E context awareness management	Mechanisms to allow each network component to dynamically adapt to the context to ensure the expected E2E QoS	QoS guarantees for a specific slice can be achieved by creating an abstract view of transport resources and network functions, simplifying network management as the network scales. Implementation of context-aware path selection, switching, and packet processing can be achieved using P4 programmability based on the context provided by an SDN domain controller. Optimal semantic orchestrators for robotic use cases to maximize the number of allocated robotic functions in the system while minimizing the consumed energy at the robot ends. Intelligent task offloading decision-making and seamless switching between different computing options, such as (a) delayed computing and (b) approximate computing, to minimize the time and energy consumed across the network.
Design of a module	This enabler analyses the different module creation options and their requirements and impacts on the network (both advantages and disadvantages)	There is a trade-off between performance and flexibility when considering the design of modular 6G architecture: More granular design results in higher flexibility in implementing and deploying modules but at the cost of reduced performance in terms of execution time and state management.

		Effect of introduced delay on registration procedure duration per NF - Sensitivity to latency introduced towards AMF, UDM and UDR network functions. Impact of Service Mesh usage on procedure completion time – next step - to switch to a Sidecarless Service Mesh Procedure-based functional decomposition of the Core network control plane is evaluated, where the new set of NFs encompasses all the interactions between 5G Core NFs to complete a certain procedure. The analysis shows that there is a reduction in the total number of messages needed for these procedures by using procedure-based functions. However, this design reduces the flexibility in deploying the more coarse-grained NFs.
Interaction between entities enabler	The network modules and their interfaces need to support the coexistence of these use cases as well as the related services.	RAN-CN control plane interactions and interfaces: There is a need to improve the cloud friendliness of the interface between RAN and the CN (NGAP/N2 interface). For this to happen, we need to evolve or replace the SCTP protocol used in 5G to support better decoupling between the different layers and avoid so called transport bindings, etc. Data-Centric Service-Based Architecture for Edge-Native 6G Network: Transition to a fully distributed 6G system with Data-Centric Networking, enabling enhanced scalability and flexibility through dynamic stateless NFs and simplified architecture with efficient resource management.
Modularisation examples enabler	This enabler, the focus is to demonstrate how the new modules can be designed at the different network domains (e.g., RAN or UP)	Modularisation of the UPF into multiple functions can improve scalability of different sub-functions. Different RAN architectures have an impact on the performance for D-MIMO. The cell-free operation provides better user rates compared to cellular networks
Multi-cloud federation enabler	The main aim of the federation model is to enhance the efficiency, reliability, and flexibility of computing infrastructure by fostering interconnectivity among diverse infrastructures	Access to more cloud resources, raising the flexibility of services that can be made available, the devices that can be reached and allow for cross-domain deployment. Higher resource availability and flexibility also lowers the total network load and latency times in service deployment, as well the ability to leverage computing power and storage that is superior to current solutions.
Orchestration of the cloud continuum	Cloud-continuum orchestration: Multi-cluster resource management, Decentralised orchestration, Orchestration of the extreme edge	Unified and abstract interface for compute continuum resource management (inventory, provision, operate) Automated discovery mechanisms for virtualization platforms and extreme edge devices capabilities Enhanced placement mechanisms to efficiently deploy, migrate and distribute network functions that have proximity constraints Tailored monitoring jobs for extreme-edge devices attributes, status and behaviour to feed zero-touch closed-loop automation mechanisms
Slicing enabler	6G slices are investigated. More specifically, this enabler presents (1) 6G network slicing, (2) Intent based management and (3) Network Slicing in the	5G and 6G slicing should co-exist since we cannot ensure 6G coverage everywhere. Similar to 5G, using the same the S-NSSAI structure to identify a 6G Slice ensures smooth transition to 6G. 5G S-NSSAI structure is sufficient. Current S-NSSAI structure contains 32 bits that can accommodate quite a large number of slice IDs thus no issue is foreseen for 6G slicing. Existing analysis does

	decentralised compute-continuum	not indicate any need to extend the S-NSSAI structure; this ensures also backward compatibility with 5G slicing. Intent based orchestration enables faster onboarding of network functions to production including provisioning of underlying cloud infrastructure with a true cloud native approach. It also reduces the costs of adoption of cloud and network infrastructure and manages a huge number of clusters of servers across the telco network, handling a variety of infrastructure technologies with a uniform and consistent user experience, automatically installing and configuring additional plugins

8 References

[22.261] 3GPP TS 22.261, “Service requirements for the 5G system”, v20.1.0, January 2025

[23.003] 3GPP TS 23.003, “Numbering, addressing and identification,” v19.1.0, January 2025.

[23.288] 3GPP TS 23.288, “Architecture enhancements for 5G System (5GS) to support network data analytics services,” v19.1.0, December 2024.

[23.482] 3GPP TS 23.482, “Functional architecture and information flows for AIML Enablement Service;,” v19.0.0, January 2025.

[23.501] 3GPP TS 23.501 “System architecture for the 5G System (5GS)”, V18.4.0, December 2023.

[23.502] 3GPP TS 23.502 “Procedures for the 5G System (5GS)”, V18.4.0, December 2023.

[23.558] 3GPP TS 23.558 “Architecture for enabling Edge Applications”, V19.3.0, September 2024

[23.700-82] 3GPP TR 23.700-82, “Study on application layer support for AI/ML services”, v19.1.0, September 2024.

[28.105] 3GPP TS 28.105, “Management and orchestration; Artificial Intelligence / Machine Learning (AI/ML) management (Release 19)” v19.1.0, Jan 2025.

[29.222] 3GPP TS 29.222 “Technical Specification Group Services and System Aspects; Common API Framework (CAPIF) - Framework for Northbound APIs; Stage 3 (Release 17)”, v17.5.0., 2022. Available at: https://www.3gpp.org/ftp/Specs/archive/29_series/29.222/.

[29.502] 3GPP TS 29.502, “Technical Specification Group Core Network and Terminals; 5G System; Session Management Services; Stage 3 (Release 18),”, v18.1.0, August 2022.

[29.510] 3GPP TS 29.510 “5G System; Network function repository services”, V18.4.0, September 2023.

[29.512] 3GPP TS 29.512, “Technical Specification Group Core Network and Terminals; 5G System; Session Management Policy Control Service; Stage 3 (Release 18)”, v18.0.0, December 2022

[29.522] 3GPP TS 29.522 “Network Exposure Function Northbound APIs; 5G; 5G System; Stage 3” version 16.4.0 Release 16

[36.300] 3GPP TS 36.300 “E-UTRA and E-UTRAN Overall Description; Stage 2”, V18.2.0, August 2024

[38.300] 3GPP TS 38.300 “NR and NG-RAN Overall Description; Stage 2”, V18.2.0, August 2024

[38.321] 3GPP TS 38.321 “Medium Access Control (MAC) protocol specification”, V18.3.0, September 2024

[38.331] 3GPP TS 38.331 “NR Radio Resource Control (RRC)”, V18.2.0, August 2024.

[38.412] 3GPP TS 38.412 “NG signalling transport”, V18.1.0, July 2024

[38.413] 3GPP TS 38.413 “NG-RAN; NG Application Protocol (NGAP)”, V18.4.0, December 2024

[38.801] 3GPP TR 38.801 “Study on new radio access technology: Radio access architecture and interfaces”, V14.0.0, March 2017.

[38.817] 3GPP TR 38.817 “General aspects for Base Station (BS) Radio Frequency (RF) for NR”, V15.11.0, September 2023.

[YAM23] Y. Yao, H. Al-kanani, and S. Mwanje, “AI/ML Management for 5G Systems,” September 2023. [Online]. Available: https://www.3gpp.org/technologies/ai-ml-management. September 2023.

[5GZ21] 5GZORRO Consortium, “D2.2: Zero-Touch Automation Framework for Multi-Stakeholder 5G Networks,”. Dec. 2021. [Online]. Available: www.5gzorro.eu/deliverables

[6GN23-D31] 6G-NTN, “Deliverable D3.1: Report on 3D multi layered NTN architecture,” July 2023. [Online]. Available: https://www.6g-ntn.eu/download/d3-1-report-on-3d-multi-layered-ntn-architecture-1st-version

[6GN24-D35] 6G-NTN “Deliverable D3.5 Report on 3D multi layered NTN architecture,” July 2023. [Online]. Available: https://6g-ntn.eu/public-deliverables/

[6GS23-D42] 6G-SHINE Deliverable D4.2 “Preliminary results on the management of traffic, computational and spectrum resources among subnetworks in the same entity, and between subnetworks and 6G network,” 6G-SHINE project, 2023, [Online]. Available: https://6gshine.eu/wp-content/uploads/2024/09/D4.2_Preliminary-results-on-the-management-of-traffic-v1.0.pdf.

[6GW24] 6G World. “Good News – Growth For A Telco! Might Be Bad News For Some, Though…”, Sep. 2024. [Online]. Available: https://www.6gworld.com/exclusives/growth-for-a-telco-might-be-bad-news-for-some-though/?utm_medium

[802.11-2016] IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012), pp. 1–3534, Dec. 2016, doi: 10.1109/IEEESTD.2016.7786995.

[AAE16] N. Alshuqayran, N. Ali and R. Evans, “A Systematic Mapping Study in Microservice Architecture,” 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA), Macau, China, 2016, pp. 44-51, doi: 10.1109/SOCA.2016.15

[BMG+22] A. Blanco, P. J. Mateo, F. Gringoli, and J. Widmer, “Augmenting MmWave localization accuracy through sub-6 GHz on off-the-shelf devices,” in Proc. of ACM MobiSys,, pp. 477–490, 2022.

[BS20] E. Björnson and L. Sanguinetti, “Scalable Cell-Free Massive MIMO Systems,” IEEE Transactions on Communications, vol. 68, no. 7, pp. 4247–4261, 2020.

[ACC+23] M. Amaral, H. Chen, T. Chiba, R. Nakazawa, S. Choochotkaew, E. K. Lee, "Kepler: A Framework to Calculate the Energy Consumption of Containerized Applications," 2023 IEEE 16th International Conference on Cloud Computing (CLOUD), Chicago, IL, USA, 2023, pp. 69-71, doi: 10.1109/CLOUD60044.2023.00017.

[AM21] A. Agrawal, and C. Makaya, “End-to-end Network Slice Architecture and Distribution Across 5G Micro-Operator Leveraging Multi-Domain and Multi-Tenancy,”. EURASIP Journal on Wireless Communications and Networking, 2021.

[Arg24] ArgoCD. ArgoCD Overview. 2024. Web site. [Online]. Available: https://argo-cd.readthedocs.io/en/stable/ (Accessed: Nov. 2024)

[AZL+23] B. Amjad et al., “Radio SLAM: A review on radio-based simultaneous localization and mapping,” IEEE Access, vol. 11, pp. 9260–9278, 2023.

[BBD+11] J. Bao, P. Basu, M. Dean, C. Partridge, A. Swami, W. Leland, and J. A. Hendler, “Towards a theory of semantic communication,” IEEE Network Science Workshop. IEEE, 2011, pp. 110–117.

[BCD+14] S. Brânzei, Y. Chen, X. Deng, A. Filos-Ratsikas, S. Frederiksen, and J. Zhang, “The Fisher Market Game: Equilibrium and Welfare”, AAAI Conference on Artificial Intelligence, vol. 28, no. 1, Jun. 2014.

[BD17] S. Buzzi and C. D’Andrea, “Cell-Free Massive MIMO: User-Centric Approach,” IEEE Wireless Communications Letters, vol. 6, no. 6, pp. 706–709, 2017.

[BMV2] 2024. BEHAVIORAL MODEL (bmv2). https://github.com/p4lang/behavioral-model.

[BQG+24] G. Baldoni, J. Quevedo, C. Guimarães, A. de la Oliva and A. Corsaro, "Data-Centric Service-Based Architecture for Edge-Native 6G Network," in IEEE Communications Magazine, vol. 62, no. 4, pp. 32-38, April 2024, doi: 10.1109/MCOM.001.2300178.

[CB17] H. Corrigan-Gibbs and D. Boneh, “Prio: Private, Robust, and Scalable Computation of Aggregate Statistics,”, 14th USENIX Symposium on Networked Systems Design and implementation (NSDI 17), 2017.

[Cam24] Linux Foundation. CAMARA Project. Web site. Available at: https://camaraproject.org (Accessed: Nov. 2024).

[CAPG] Capgemini Research Institute, “Networks with Intelligence – Why and how the telecom sector should accelerate its autonomous networks journey”, 2024

[CB17] H. Corrigan-Gibbs and D. Boneh Prio, “Private, Robust, and Scalable Computation of Aggregate Statistics,” Stanford University, March 14, 2017.

[CDHAC01] Commonwealth Department of Health and Aged Care, "Measuring Remoteness: Accessibility/Remoteness Index of Australia (ARIA)", Revised Edition, Occasional Papers: New Series No. 14,, 2001. ISBN 0642503273

[CDP24] P. Charatsaris, M. Diamanti and S. Papavassiliou, "Joint User Association and Resource Allocation for Hierarchical Federated Learning Based on Games in Satisfaction Form," in IEEE Open Journal of the Communications Society, vol. 5, pp. 457-471, 2024, doi: 10.1109/OJCOMS.2023.3347354.

[CDT18] T. Cerny, M. J. Donahoo, and M. Trnka, “Contextual understanding of microservice architecture: current and future directions,” ACM SIGAPP Applied Computing Review, vol. 17, no. 4, pp. 29-45, 2018.

[CHE24] 2024. Chaos Engineering. https://www.ibm.com/topics/chaos-engineering

[CHM24] 2024. Chaos mesh. https://chaos-mesh.org/docs/

[Con24] HashiCorp. What is Consul?. 2024. Web site. [Online]. Available: https://developer.hashicorp.com/consul/docs/intro (Accessed: Nov 2024)

[Cur04] G R. Curry, “Radar system performance modelling,” Artech House, second edition, 2004.

[DNC+01] M. Dissanayake et al., “A solution to the simultaneous localization and map building (slam) problem,” IEEE Transactions on Robotics and Automation, vol. 17, no. 3, pp. 229–241, 2001.

[DPT+2024] M. Diamanti, C. Pelekis, E. E. Tsiropoulou and S. Papavassiliou, "Delay Minimization for Rate-Splitting Multiple Access-Based Multi-Server MEC Offloading," in IEEE/ACM Transactions on Networking, vol. 32, no. 2, pp. 1035-1047, April 2024, doi: 10.1109/TNET.2023.3311131.

[Sam21] Samsung, “Technical White Paper, Dynamic Spectrum Sharing,” Samsung 2021 link: https://images.samsung.com/is/content/samsung/assets/global/business/networks/insights/white-papers/0122_dynamic-spectrum-sharing/Dynamic-Spectrum-Sharing-Technical-White-Paper-Public.pdf

[EBP24] eBPF. https://ebpf.io/, Accessed Dec. 2024

[EFW02] John C Eidson, Mike Fischer, and Joe White,. “IeeeIEEE-1588™ standard for a precision clock synchronization protocol for networked measurement and control systems,”. In Proceedings of the 34th Annual Precise Time and Time Interval Systems and Applications Meeting, pages 243–254, 2002.

[EFH+23] S. Euler, X. Fu, S. Hellsten, C. Kefeder, O. Lidberg, E. Medeiros, E. Nordell, D. Singh, P. Synnergren, E. Trojer, I. Xirouchakis, “Using 3GPP technology for satellite communication”, Ericsson Technology Review, June 2023, Online: https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/3gpp-satellite-communication

[ETSI36] ETSI, “White Paper 36, harmonizing Standards for Edge Computing; A Synergized Architecture Leveraging ETSI ISG MEC and 3GPP Specifications,” ETSI, Sophia Antipolis, July 2020. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/ETSI_wp36_Harmonizing-standards-for-edge-computing.pdf

[FA18] M. Filipenko and I. Afanasyev, “Comparison of various SLAM systems for mobile robot in an indoor environment,” in Proc. of IS, pp. 400–407, 2018.

[FHS18] Y. Fan, Z. Hui, and Q. Su, “Federation of Cross-Domain Edge Resources: A Brokering Architecture for Network Slicing,” IEEE NetSoft 2018 - International Workshop on Advances in Slicing for Softwarized Infrastructures, 2018.

[FLO] Flower, https://flower.ai/, Accessed Dec. 2024

[Free5GC] “Free5GC.” Accessed: Jul. 2022. [Online]. Available: https://www. free5gc.org/

[FSE+22] P. Foley, M. Sheller, B. Edwards, S. Pati, W. Riviera, M. Sharma, P, Narayana Moorthy, S. H. Wang, J. Martin, P. Mirhaji, P. Shah, and S. Bakas, “OpenFL: the open federated learning library,” Physics in Medicine & Biology, vol. 67, no. 21, pp. 214001, 2022.

[FSP+20] F. Faticanti, M. Savi, F. D. Pellegrini, P. Kochovski, V. Stankovski and D. Siracusa, "Deployment of Application Microservices in Multi-Domain Federated Fog Environments," 2020 International Conference on Omni-layer Intelligent Systems (COINS), Barcelona, Spain, 2020, pp. 1-6, doi: 10.1109/COINS49042.2020.9191379

[GPP+24] T. Geoghegan, C. Patton, B. Pitman, E. Rescorla, A. Wood, “Distributed Aggregation Protocol for Privacy Preserving Measurement,,” May 21, 2024. [Online]. Available: https://www.ietf.org/archive/id/draft-ietf-ppm-dap-11.html.

[GAT24] P. Garau Burguera, H. Al-Tous, and O. Tirkkonen, “Distributed user-centric cell-free massive MIMO with architectural constraints”, in Proc. EuCNC & 6G summit, 2024, pp. 541-546.

[GAT24b] P. Garau Burguera, H. Al-Tous, and O. Tirkkonen, “Remote radio head multiclustering based cell-free massive MIMO systems”, in Proc. VTC-Fall, 2024.

[GCF+22] M. Gramaglia et al., “Network intelligence for virtualized ran orchestration: The daemon approach,” in 2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), 2022, pp. 482–487.

[GKV+19] T. Goethals, S. Kerkhove, L. Van Hoye, M. Sebrechts, F. De Turck and B. Volckaert, “FUSE : a microservice approach to cross-domain federation using docker containers.” In V. M. Munoz, D. Ferguson, M. Helfert, & C. Pahl (Eds.), Closer: Proceedings of the 9th International Conference on Cloud Computing and Services Science, pp. 90–99, 2019. https://doi.org/10.5220/0007706000900099

[GLM01] V. Giovannetti, S. Lloyd, and L. Maccone,. “Quantum-enhanced positioning and clock synchronization,”. Nature, vol. 412,: pp. 417–419, 2001

[GSH+23] E. Goshi, R. Stahl, H. Harkous, M. He, R. Pries and W. Kellerer, “PP5GS - An Efficient Procedure-Based and Stateless Architecture for Next-Generation Core Networks,” in IEEE Transactions on Network and Service Management, vol. 20, no. 3, pp. 3318-3333, Sept. 2023.

[GSM17] GSM Association “An Introduction to Network Slicing,” 2017. [Online]. Available at: https://www.gsma.com/solutions-and-impact/technologies/networks/wp-content/uploads/2017/11/GSMA-An-Introduction-to-Network-Slicing.pdf (Accessed: Nov. 2024).

[Har22] H. Harkous, “Performance Modeling, Optimization, and Applications for the Deployment of Programmable Packet Processors in Cloud Environments”, Technical University of Munich, Germany, 2022, [Online]. Available: https://mediatum.ub.tum.de/doc/1661439/1661439.pdf

[Har24] Harbour. What is Harbour?. 2024. Web site. [Online]. Available at: https://goharbor.io/ (Accessed: Nov. 2024)

[HEL24] Helm, Package manager for Kubernetes, 2024. [Online]. Available: https://helm.sh/docs/

[HEX223-D12] Hexa-X-II Deliverable D1.2 "6G Use Case and Requirements", Hexa-X-II project, 2023. [Online]. Available: https://hexa-x-ii.eu/wp-content/uploads/2024/01/Hexa-X-II_D1.2.pdf.

[HEX223-D32] Hexa-X-II Deliverable D3.2 "Initial Architectural enablers", Hexa-X-II project, 2023. [Online]. Available: https://hexa-x-ii.eu/wp-content/uploads/2023/11/Hexa-X-II_D3.2_v1.0.pdf.

[HEX224-D13] Hexa-X-II Deliverable D1.3 "Environmental and social view on 6G", Hexa-X-II project, 2024. [Online]. Available: https://hexa-x-ii.eu/wp-content/uploads/2024/03/Hexa-X-II_D1.3_v1.00_GA_approved.pdf.

[HEX224-D23] Hexa-X-II Deliverable D2.3 “Interim overall 6G system design,” Hexa-X-II project, 2024, [Online]. Available: https://hexa-x-ii.eu/wp-content/uploads/2024/07/Hexa-X-II_D2.3-v1.1.pdf

[HEX224-D33] Hexa-X-II Deliverable D3.3 “Initial analysis of architectural enablers and framework,” Hexa-X-II project, 2024, [Online]. Available: https://hexa-x-ii.eu/wp-content/uploads/2024/04/Hexa-X-II_D3.3_v1.0.pdf

[HEX224-D63] Hexa-X-II Deliverable D6.3 "Initial Design of 6G Smart Network Management Framework", Hexa-X-II project, Jun. 2024. [Online]. Available: https://hexa-x-ii.eu/wp-content/uploads/2024/07/Hexa-X-II_D6-3_v1.0.pdf.

[HEX224-D65] Hexa-X-II Deliverable D6.5 "Final Design on 6G Smart Network Management Framework", Hexa-X-II project, to be released Feb. 2025.

[HEX23-D53] Hexa-X Deliverable D5.3 “Final 6G architectural enablers and technological solutions,” Hexa-X project, 2023, [Online]. Available: https://hexa-x.eu/wp-content/uploads/2023/05/Hexa-X_D5.3_v1.0.pdf

[HEX23-D23] Hexa-X Deliverable D2.3, “Radio models and enabling techniques towards ultra-high data rate links and capacity in 6G”, 2023, [Online]. Available: https://hexa-x.eu/wp-content/uploads/2023/04/Hexa-X-D2_3_v1.0.pdf

[Ick+23] S. Ickin, "Automated Feature Selection with Local Gradient Trajectory in Split Learning," NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium, Miami, FL, USA, 2023, pp. 1-7, 2023, doi: 10.1109/NOMS56928.2023.10154435.

[IBM24] IBM, “What is machine learning (ML)?”. 2024. [Online] Available at: https://www.ibm.com/topics/machine-learning

[ILE24] Infrastructure Layer Emulator (ILE). Web site. Available at: https://gitlab.com/decentralized-continuum-orchestration/infrastructure-layer-emulator (Accessed: Nov. 2024).

[JAD+00] Richard R. Jozsa, Daniel D. S. Abrams, Jonathan J. P. Dowling, and Colin C. P. Williams,. “Quantum clock synchronization based on shared prior entanglement,”. Physical Review Letters, vol. 85, no. (9),: 2000

[JW24] P-H. Juan, J-L. Wu, "Enhancing Communication Efficiency and Training Time Uniformity in Federated Learning through Multi-Branch Networks and the Oort Algorithm", in Algorithms, 2024.

[K3S] K3S Lightweight Kubernetes. Web site. Available at: https://k3s.io (Accessed: Nov. 2024).

[K8S] Kubernetes. Web site. Available at: https://kubernetes.io (Accessed: Nov. 2024).

[Kepler] 2024. Kepler. https://sustainable-computing.io/

[KP17] Kratzke, N., & Peinl, R. (2017). ClouNS - A Cloud-native Application Reference Model for Enterprise Architects. Retrieved from https://arxiv.org/abs/1709.04883

[KSER] Kserve, https://kserve.github.io/website/latest/

[LHX+20] Y. Li, A. Huang, Y. Xiao, X. Ge, S. Sun, H. C. Chao, “Federated orchestration for network slicing of bandwidth and computational resource,” 2020. arXiv preprint arXiv:2002.02451. [online] Available at: https://arxiv.org/abs/2002.02451 (Accessed: Nov. 2024)

[LIM24] 2024. "available metrics"- Linkerd. https://linkerd.io/2-edge/reference/proxy-metrics/

[LIN24] 2024, Linkerd. https://linkerd.io/2.17/overview/

[LJZ+22] P. Liu, J. Jiang, G. Zhu, L. Cheng, W. Jiang, W. Luo, Y. Du, Z. Wang, "Training time minimization for federated edge learning with optimized gradient quantization and bandwidth allocation", in Frontiers of Information Technology & Electronic Engineering, 2022, pp.1247-1263

[LXD24] LXD documentation. Web site. Available at: https://documentation.ubuntu.com/lxd/en/latest. (Accessed: Nov. 2024).

[MEC003] ETSI GS MEC 003: Multi-Access Edge Computing (MEC); Framework and Reference Architecture, ETSI Std. v2.2.1, 2020.

[MEC035] GR MEC 035: Multi-Access Edge Computing (MEC); Study on Inter-MEC systems and MEC-Cloud Systems Coordination, ETSI Std. v3.1.1, 2021.

[MES24] Apache Mesos. Web site. Available at: https://mesos.apache.org (Accessed: Nov. 2024)

[MIC] MicroK8s. Web site. Available at: https://microk8s.io (Accessed: Nov 2024)

[Mic24] Microsoft. What is DevOps?. 2024. [Online] Available: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-devops

[Min24a] MiniKube. Web site. Available at: https://minikube.sigs.k8s.io/docs (Accessed: Nov. 2024)

[Min24b] MinIO, https://min.io/, (Accessed: Nov 2024)

[MLF24] MLFlow, https://mlflow.org/, (Accessed: Nov 2024)

[Mon24] MongoDB. https://www.mongodb.com/docs/, (Accessed: Nov 2024)

[MRB+24] S. Maheshwari, V. Raman, R. Bassoli, F. H. P. Fitzek, "Entanglement-assisted decision making for VNF migration in 6G Communication Networks", November 2024, DOI:10.1109/NFV-SDN61811.2024.10807502

[MWQ+23] S. Ma, Y. Wu, H. Qi, H. Li, G. Shi, Y. Liang, and N. Al-Dhahir, "A theory for semantic communications," arXiv preprint arXiv:2303.05181, 2023.

[MY5G] 2024. My5GRANTESTER. https://github.com/my5G/my5G-RANTester

[NAY+17] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-Free Massive MIMO Versus Small Cells," IEEE Transactions on Wireless Communications, vol. 16, no. 3, pp. 1834–1850, 2017

[NFV006] ETSI GS NFV 006 Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Architectural Framework Specification, May 2024

[NGB+24] S. S. Nande, A. Garbugli, Ri. Bassoli, and F. H. P. Fitzek. "Time synchronization in communication networks: A comparative study of quantum technologies", IEEE Wireless Communications and Networking Conference (WCNC), pp 1–6 (2024)

[NGM16] NGMN 5G Project Requirements & Architecture – Work Stream E2E Architecture Version 1.0, 13th January 2016.

[NHS+24a] N. Nunavath, N. Hello, E. C. Strinati, R. Bassoli, and F. H. Fitzek, "Towards Quantum Semantic Communications: A Framework for Integrating Quantum and Semantic Technologies," (accepted in IEEE GlobeCom workshop 2024)

[NHS+24] N. Nunavath, M. I. Habibie, E. C. Strinati, R. Bassoli, and F. H. Fitzek, "Quantum semantic communications for graph-based models," Authorea Preprints, 2024.

[NIK24] 2024. NIKSS. https://github.com/NIKSS-vSwitch/nikss.

[Nom24] Nomad. Web site. Available at: https://www.nomadproject.io (Accessed: Nov. 2024).

[NRS99] National Research Council. 1999. Trust in Cyberspace. Washington, DC, The National Academies Press. https://doi.org/10.17226/6161.

[NSB+24] N. Nunavath, E. C. Strinati, R. Bassoli, and F. H. Fitzek, "Pragmatic semantic communication through quantum channel," Authorea Preprints, 2024

[OAD24] O-RAN Work Group 1 (Use Cases and Overall Architecture), "O-RAN Architecture Description", V11.00, February 2024

[OCI24] Open Container Initiative. https://opencontainers.org/, accessed 2024.

[OFL24] OpenFL, https://openfl.io/, Accessed Dec. 2024

[ONF24] ONF. Software-Defined Networking (SDN) Definition. 2024. Web site. Available at: https://opennetworking.org/sdn-definition/ (Accessed: Nov. 2024)

[OPK+22] T. Osiński, J. Palimąka, M. Kossakowski, F. Dang Tran, E.-F. Bonfoh, and H. Tarasiuk, "A novel programmable software datapath for Software-Defined Networking". In Proceedings of the 18th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '22). Association for Computing Machinery, New York, NY, USA, 245–260, 2022.

[ORAN003] O-RAN Technical Specification, "Non-RT RIC: Architecture", O-RAN.WG2.Non-RT-RIC-ARCH-R003-v05.00, 2023.

[OSA24] O-RAN Work Group 1 (Use Cases and Overall Architecture), "Slicing Architecture", V12.00, February 2024

[OVS24] 2024. Open vSwitch. https://www.openvswitch.org/.

[PER09] Perry, G. (2009). Open cloud manifesto: Version 1.0.9. Retrieved from https://gevaperry.typepad.com/Open%20Cloud%20Manifesto%20v1.0.9.pdf

[PGB+23] P. Picazo, M. Groshev, A. Blanco, C. Fiandrino, A. de La Oliva and J. Widmer, "waveSLAM: Empowering Accurate Indoor Mapping Using Off-the-Shelf Millimeter-wave Self-sensing," 2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall), Hong Kong, Hong Kong, 2023, pp. 1-7, doi: 10.1109/VTC2023-Fall60731.2023.10333840.

[Pre24] Prefect, https://www.prefect.io/, Accessed Dec. 2024.

[PRO22] “Prometheus.” Accessed: Jul. 2022. [Online]. Available: https://prometheus.io/

[RBF24]. V. Raman, R. Bassoli, F. H. P. Fitzek “On Destination Anonymity and Trustworthiness in 6G Networks: Can Quantum Entanglement Offer Unexpected Advantages?”, Accepted for Publication (January 2025)a

[RFC-1034] P. Mockapetris, "Domain Names - Concepts and Facilities", IETF Network Working Group, November 1987. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc1034

[RFC2544] Scott Bradner and Jim McQuaid, “Benchmarking Methodology for Network Interconnect Devices,” RFC 2544, 1999.

[RGO+24] E. Rojas, C. Guimarães, A. de la Oliva, C. J. Bernardos and R. Gazda, "Beyond Multi-Access Edge Computing: Essentials to Realize a Mobile, Constrained Edge," in IEEE Communications Magazine, vol. 62, no. 1, pp. 156-162, January 2024, doi: 10.1109/MCOM.017.2300056.

[SB21] E. C. Strinati and S. Barbarossa, “6g networks: Beyond shannon towards semantic and goal-oriented communications,” Computer Networks, vol.190, p. 107930, 2021.

[SEL24] Seldon, https://docs.seldon.io/projects/seldon-core/en/v2/index.html, accessed Dec. 2024.

[SFU24] M. Shaheen, M. S. Farooq and T. Umer, "AI-empowered mobile edge computing: inducing balanced federated learning strategy over edge for balanced data and optimized computation cost", in Journal of Cloud Computing, 2024

[SGS21] D. Saxena, R. Gupta, and A. K. Singh, “A Survey and Comparative Study on Multi-Cloud Architectures: Emerging Issues and Challenges for Cloud Federation”, August 2021. Arxiv https://arxiv.org/pdf/2108.12831

[STS+22] C. Spiess, S. Töpfer, S. Sharma, A. Kržič, M. C. Ponce, U. Chandrashekara, N. L. Döll, D. Rieländer, F. Steinlechner, „Clock synchronization with correlated photons“,arXiv:2108.13466 (2022)

[SW49] C. Shannon and W. Weaver, The Mathematical Theory of Communication, ser. Illini books. University of Illinois Press, 1949, no. v. 1.

[SWA] Docker Swarm. Web site. Available at: https://docs.docker.com/engine/swarm (Accessed: Nov. 2024).

[SWI] RedHat OpenSwift. Web site. Available at: https://www.redhat.com/en/technologies/cloud-computing/openshift (Accessed: Nov. 2024)

[TAL+22] T. Taleb, et al., "6G Technologies: Visions, Requirements, and Key Enablers." IEEE Network, (2022).

[UBW24] Ubiwhere, "Urban Platform", Ubiwhere, 2024 [Online] Available at: https://urbanplatform.city/

[UNe10] H. Uchida, Hirotsugu; and A. Nelson, Andrew (2010) : “Agglomeration index: Towards a new measure of urban concentration,” WIDER Working Paper, No. 2010/29, ISBN 978-92-9230-264-1, The United Nations University World Institute for Development Economics Research (UNU-WIDER), Helsinki, 2010.

[URB+21] M. A. Uusitalo, P. Rugeland, M. R. Boldi, E. C. Strinati, P. Demestichas, M. Ericson, G. P. Fettweis, M. C. Filippou, A. Gati, M. -H. Hamon, M. Hoffmann, M. Latva-Aho, A. Pärssinen, B. Richerzhagen, H. Schotten, T. Svensson, G. Wikström, H. Wymeersch, V. Ziegler, and Y. Zou, “6G Vision, Value, Use Cases and Technologies From European 6G Flagship Project Hexa-X,” IEEE Access, vol. 9, pp. 160 004–160 020, 2021

[VSS04] Alejandra A. Valencia, Giuliano G. Scarcelli, and Yanhua Y. Shih,. “Distant clock synchronization using entangled photon pairs,”. Applied Physics Letters, vol. 85, no.(13,) pp.: 2655–2657, 2004.

[WDR09] World Bank, “World Development Report 2009: Reshaping Economic Geography,” 2009. [Online]. Available: http://hdl.handle.net/10986/5991.

[WN23] D. Wheeler and B. Natarajan, “Engineering semantic communication: A survey,” IEEE Access, vol. 11, pp. 13 965–13 995, 2023

[Y.3182] International Telecommunication Union, “Machine learning based end-to-end multi-domain network slice management and orchestration,” (ITU-T Recommendation Y.3182). Geneva, Switzerland: ITU. 2022. [online] Available at: https://www.itu.int (Accessed: Nov. 2024).

[Yeg11] S. Yegge, “Stevey's Google Platforms Rant,” 2011. [Online]. Available: https://gist.github.com/chitchcock/1281611 (Accessed: Nov. 2024)

[YER24] R. Yersainov, “Beyond communication services in future networks: 6G-enabled driving assistance through localization and sensing”, Master Thesis, 2024

[YWS+24] V. Yadhav, A. Williams, O. Smid, J. Kjällman, R. Islam, J. Halén, W. John, “Benefits of Dynamic Computational Offloading for Mobile Devices”, Proceedings of the 14th International Conference on Cloud Computing and Services Science - Volume 1, p265-276, SciTePress, doi={10.5220/0012719800003711},

Annex A Further details of the studies

A.1 DataOps

A.1.1 Kubernetes-based failure detection and prediction data sourcing for 5G Core

This section provides more details about the data sourcing and failure detection processes that are referenced in section 3.1.1.4. The 5GCOP platform sources data from multiple sources via the Prometheus scraping process (as seen in Figure A-1)

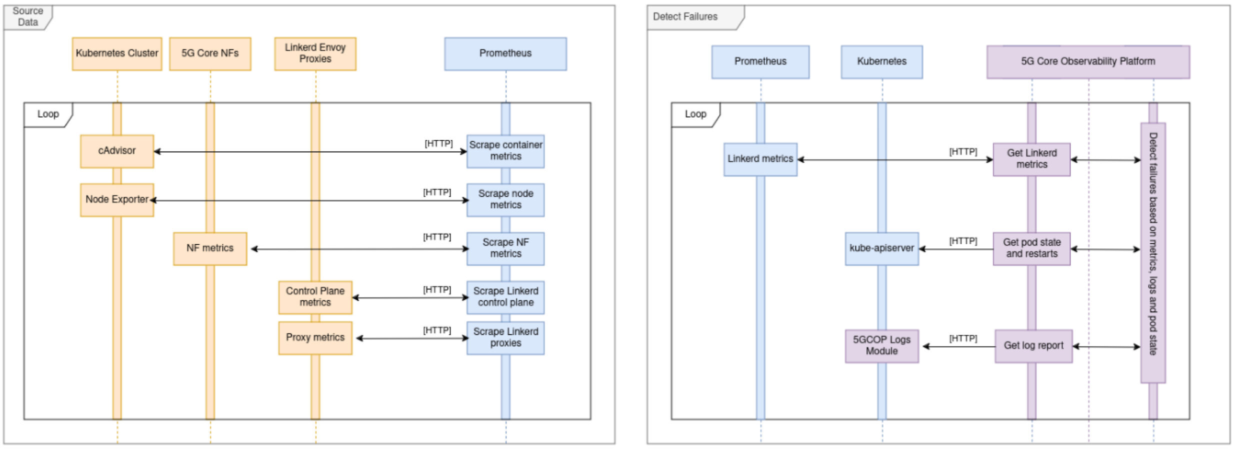
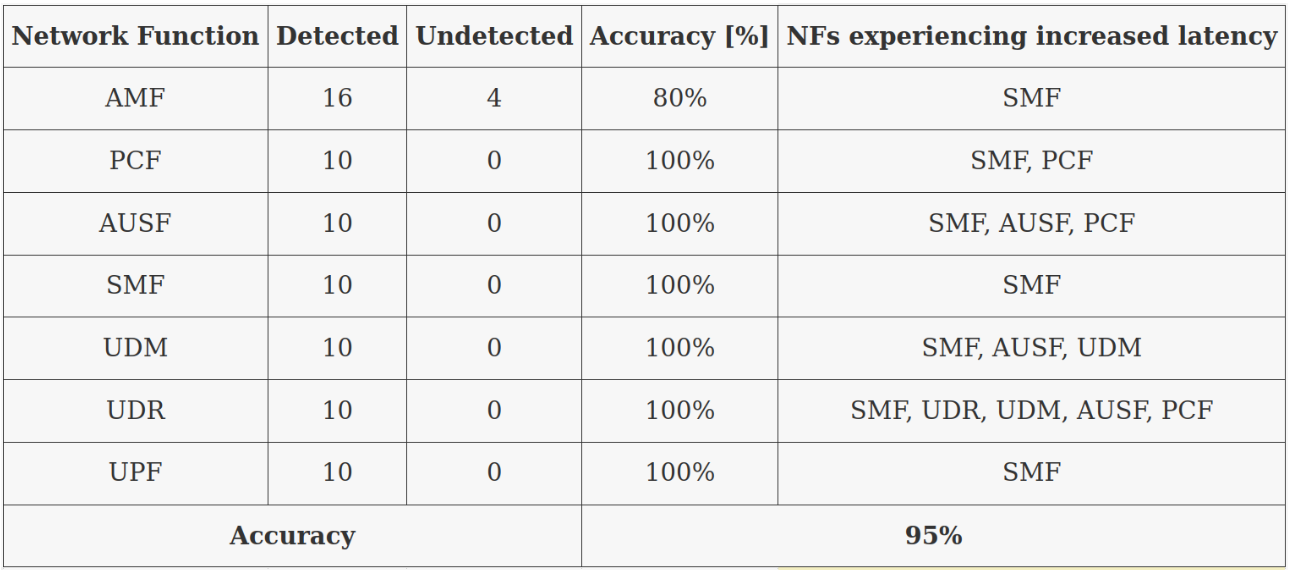


Figure A-1 5GCOP data sourcing and failure detection sequence diagrams

Data sources include the Kubernetes cluster, mainly the cAdvisor, supplying information regarding the containers, and the Node Exporter, providing information about Node status. Metrics available via Open5GS network functions and Linkerd Service Mesh containers are also scraped and collected via Prometheus. The scraping process is run continuously every 10 seconds (or any other value, should we need to change the frequency), in a loop.

The failure detection sequence (also pictured in Figure A-1) is also run continuously. The 5GCOP platform requests the Linkerd data from Prometheus via the exposed API. In a similar manner, the platform also obtains data about pod restarts and statuses from the Kubernetes cluster, via the kube-apiserver api. Finally, the platform requests the log report from the 5GCOP log module, which parses logs from 5G network functions and MongoDB database, looking for keywords and phrases that could indicate a failure or an anomaly. Finally, based on the collected log reports, metrics, statuses and restarts, the 5GCOP platform runs the Failure Detection algorithm. If any service level objective criteria are not met, the failure is detected, and the platform provides information regarding the type of failure and the problematic network function.

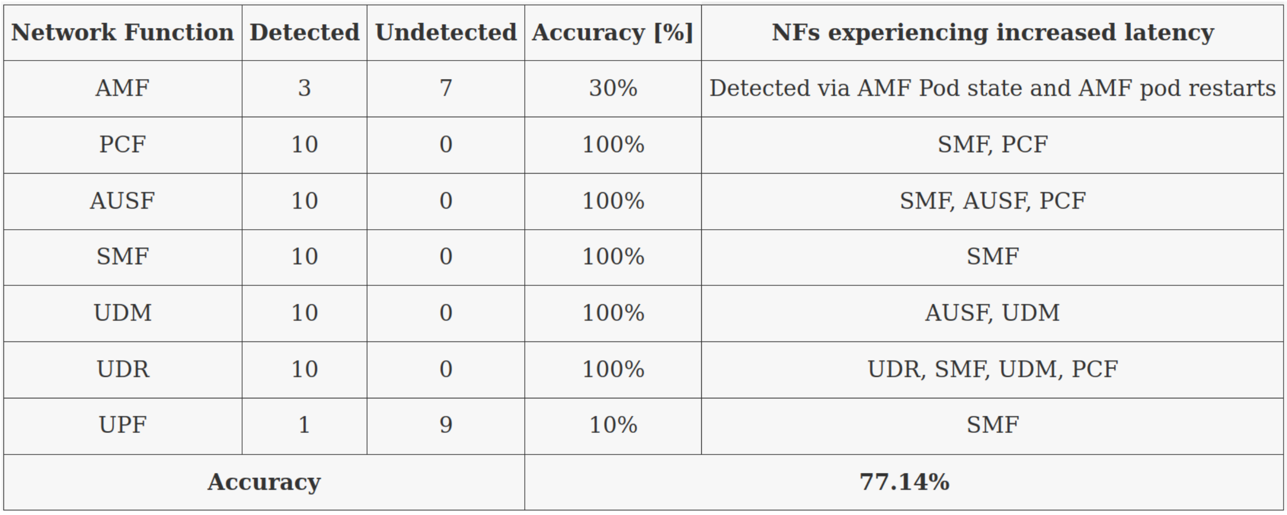
Table A-1 Detection results for the Jitter experiment



Network Function	Detected	Undetected	Accuracy [%]	NFs experiencing increased latency
AMF	16	4	80%	SMF
PCF	10	0	100%	SMF, PCF
AUSF	10	0	100%	SMF, AUSF, PCF
SMF	10	0	100%	SMF
UDM	10	0	100%	SMF, AUSF, UDM
UDR	10	0	100%	SMF, UDR, UDM, AUSF, PCF
UPF	10	0	100%	SMF
Accuracy			**95%**	

The undetected cases for AMF network function (Table A-1) can be explained by the frequency between data scraping carried out by Prometheus (equal to 10 seconds), which comes closes in value to the test duration, during which the platform attempts to detect failures (also 10 seconds).

Table A-2 Detection results for the Packet Loss experiment



Network Function	Detected	Undetected	Accuracy [%]	NFs experiencing increased latency
AMF	3	7	30%	Detected via AMF Pod state and AMF pod restarts
PCF	10	0	100%	SMF, PCF
AUSF	10	0	100%	SMF, AUSF, PCF
SMF	10	0	100%	SMF
UDM	10	0	100%	AUSF, UDM
UDR	10	0	100%	UDR, SMF, UDM, PCF
UPF	1	9	10%	SMF
Accuracy			**77.14%**	

The inaccuracy in detection for packet loss introduced towards AMF and UPF (Table A-2) can be explained by two factors - no retransmissions between the traffic generator and the AMF network function (which means that if packet loss occurs during the first message of procedure initiation, that message is lost) and the rarity of the given network functions participation in the procedure (the UPF only exchanges 1 message with the SMF). These results suggest that including the RAN in failure detection would benefit test cases, where failure occurs between the RAN and the Core network.

A.1.2 High-level procedures for privacy preserving data collection

The main introduced architectural components are UE aggregation unit, which performs privacy-preserving data aggregation by using secure aggregation techniques and/or data anonymization to collect user data, and data-driven network control unit, an entity on the network side that configures the base station by performing automation, optimization, and intelligence services based on both network and UE data.

As introduced in [HEX224-D33], a potential implementation may be based on privacy-preserving cryptographic protocols like Prio [CB17]. It consists of Clients, which split data into shares and send them to two different Aggregators (Leader and Helper, residing in non-colliding trust domains), responsible for aggregating UE data shares, and Collector, an entity that obtains the aggregated UE data. The Leader and the

Helper send aggregated shares to the Collector that computes the final aggregation result, as shown in Figure A-2. Moreover, data can be additionally protected by differential privacy (DP), by adding noise to original data samples.

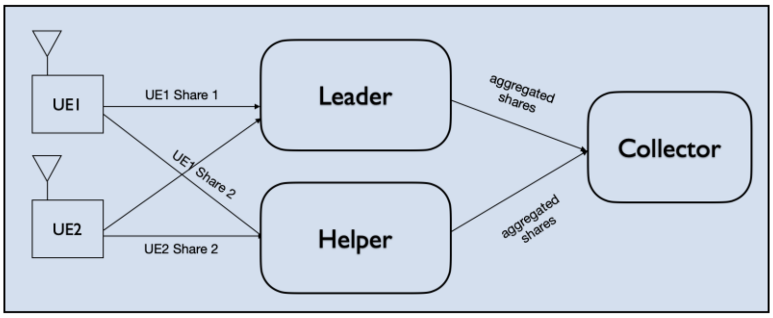


Figure A-2 Prio-based aggregation system architecture.

A possible realization of Prio protocol in future cellular network architecture can be based on Network Data Analytics Function (NWDAF) service framework. NWDAF, introduced in [23.288], provides analytics to 5G Core NFs to enable autonomous network operations and service management, allowing for the integration of various data-driven AI/ML technologies into the 5G networks.

Acting as a service consumer, NWDAF can collect data using service-based interface (SBI) request/response (Data collection procedures) from NFs, AFs, and OAM, and, acting as a service producer, provides them with analytics (traffic load and resource utilisation, network/service performance measurements, user mobility pattern analysis, etc.) and predictions (temporal/spatial traffic distribution prediction, UE location prediction, etc.). However, NWDAF framework raises potential privacy concerns, such as exposure of privacy sensitive UE data to network operator.

RAN deployment of privacy-preserving data collection

The potential contextual data to be shared can be classified into *UE contextual data*, containing, e.g., application type, number of applications, etc., and network (NW) contextual data, containing, e.g., measurements (RSRP, RSRQ and SINR) and network configurations (e.g., frequency band, beam index, number of layers, etc.).

The contextual data can have different levels of granularity, which are associated with different privacy sensitivity levels. These impose different privacy requirements for contextual data collection that need to be met by network, i.e., by associating them with different privacy-preserving methods (i.e., whether to use Prio with or without Differential Privacy (DP)), with different noise levels.

RAN deployment with Collector in RAN

The high-level message exchange of a privacy-preserving aggregation procedure for a potential RAN deployment in future cellular network is shown in Figure A-3. Compared to the deployment in Section 3.1.1.3, here the Collector is implemented in RAN, allowing for collection of aggregate statistics in a smaller time scale.

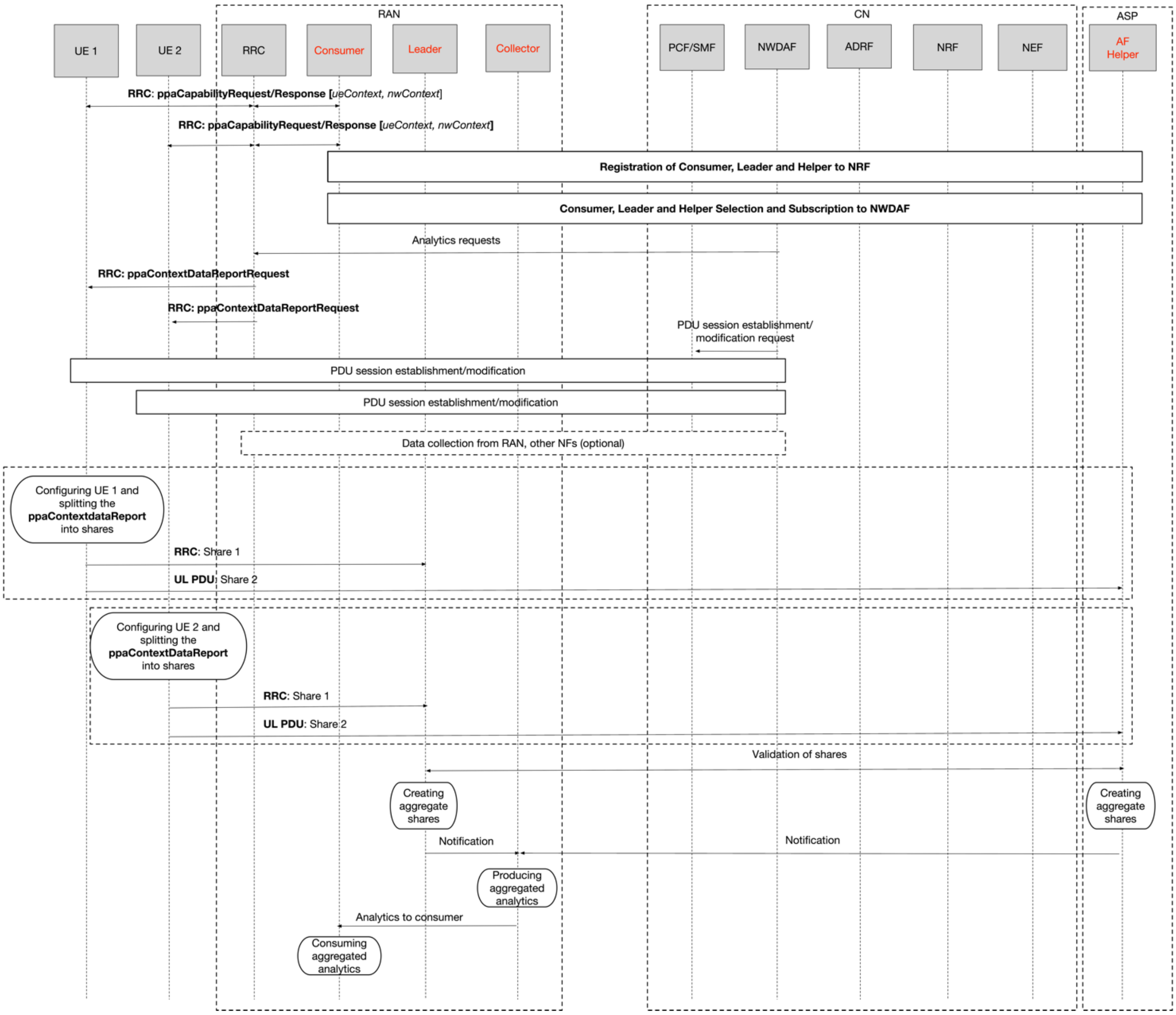

RAN
CN
ASP
UE 1
UE 2
RRC
Consumer
Leader
Collector
PCF/SMF
NWDAF
ADRF
NRF
NEF
AF Helper
RRC: ppaCapabilityRequest/Response [ueContext, nwContext]
RRC: ppaCapabilityRequest/Response [ueContext, nwContext]
Registration of Consumer, Leader and Helper to NRF
Consumer, Leader and Helper Selection and Subscription to NWDAF
Analytics requests
RRC: ppaContextDataReportRequest
RRC: ppaContextDataReportRequest
PDU session establishment/ modification request
PDU session establishment/modification
PDU session establishment/modification
Data collection from RAN, other NFs (optional)
Configuring UE 1 and splitting the ppaContextdataReport into shares
RRC: Share 1
UL PDU: Share 2
Configuring UE 2 and splitting the ppaContextDataReport into shares
RRC: Share 1
UL PDU: Share 2
Validation of shares
Creating aggregate shares
Creating aggregate shares
Notification
Notification
Producing aggregated analytics
Analytics to consumer
Consuming aggregated analytics

Figure A-3 An example of the RAN deployment privacy-preserving aggregation procedure in future cellular network (Option 2 – Collector in RAN).

A.2 MLOps

A.2.1 Cooperative inference in SN

The high-level message exchange for cooperative inference in SN is shown in Figure A-4. UE1, UE2 and UE3 create local SN by choosing U1 as a SN management node (MgtN). The model can be stored in 5GC (ADRF) or, potentially, in UE1 or UE2. From each UE, NWDAF obtains its *data privacy level*, *cooperation budget* and *available computing resources*. Application acquires the list of UEs with capability, or that are willing, to cooperate. AF sends the *CooperationRequest* to NWDAF. UE3 (from SN) and UE4 (not part of SN), also subscribe for cooperation to NWDAF. NWDAF then decides on creation of cooperative group (UE1, UE2, UE3, all in subnetwork), choosing GH (UE1), which would do aggregation of inferred data. Based on application characteristics and the selected cooperative group, NWDAF requests Cooperative session establishment from PCF/SMF by sending *CooperationSessionEstablishmentRequest*. This is followed by cooperative session establishment, consisting of UEs DL/UL sessions and local SN communication modifications according to cooperation requirements. If the model is stored in Analytics Data Repository Function (ADRF), NWDAF retrieves the model from ADRF and sends it to cooperative group. If the model is stored in UE1, acting as a GN, UE1 sends the model to UE2 and UE3. UE2 and UE 3 infer local data on model obtained either from NWDAF, or from GH (UE1), or on local model and sends the inferred data over

SN to GN (UE1). Locally inferred data is aggregated at GH and sent over SN to the subscribed consumer UE3. GH (UE1) can also send aggregated data to Application (via AF). Optionally, data can be used for model retraining in NWDAF and sent to UE 4. The model can also be stored in ADRF.

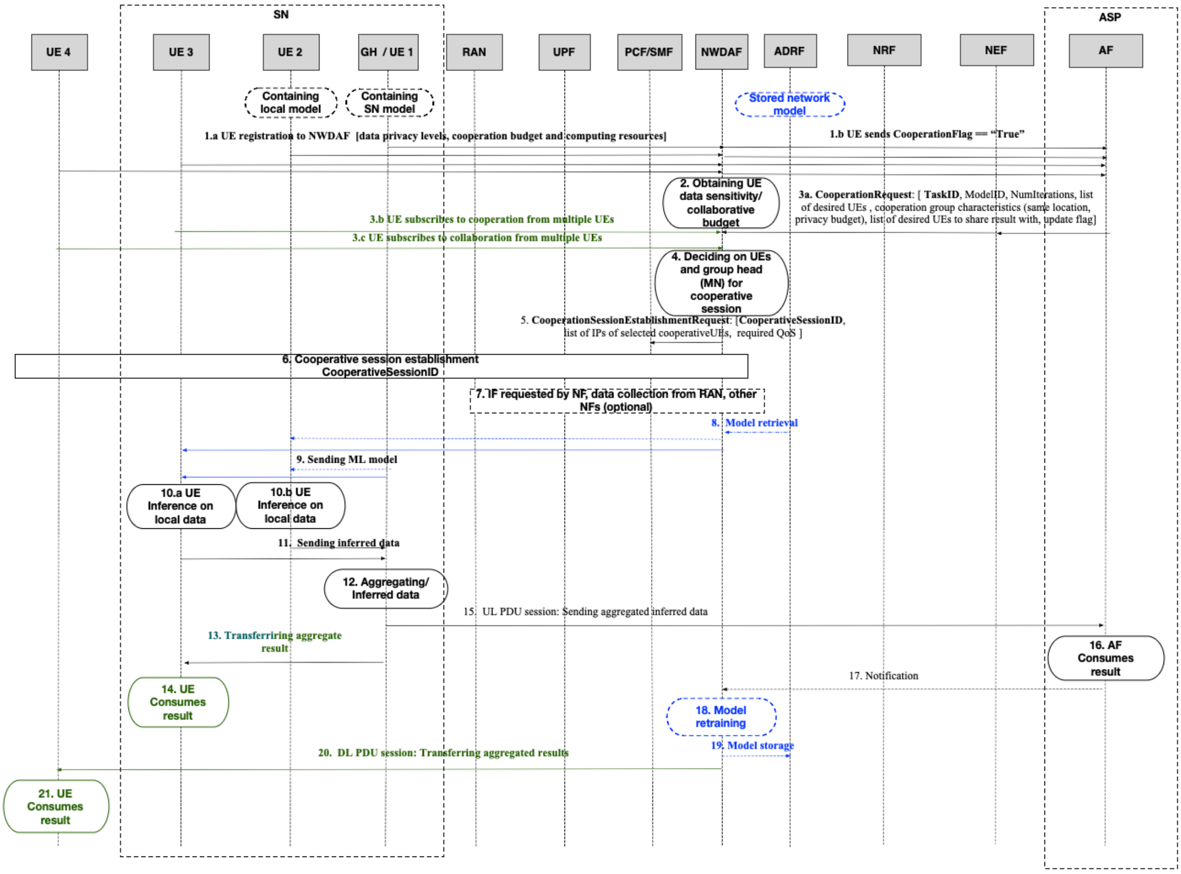

SN
ASP
UE 4
UE 3
UE 2
GH / UE 1
RAN
UPF
PCF/SMF
NWDAF
ADRF
NRF
NEF
AF
Containing local model
Containing SN model
Stored network model
1.a UE registration to NWDAF [data privacy levels, cooperation budget and computing resources]
1.b UE sends CooperationFlag == "True"
2. Obtaining UE data sensitivity/ collaborative budget
3a. CooperationRequest: [TaskID, ModelID, NumIterations, list of desired UEs , cooperation group characteristics (same location, privacy budget), list of desired UEs to share result with, update flag]
3.b UE subscribes to cooperation from multiple UEs
3.c UE subscribes to collaboration from multiple UEs
4. Deciding on UEs and group head (MN) for cooperative session
5. CooperationSessionEstablishmentRequest: [CooperativeSessionID, list of IPs of selected cooperativeUEs, required QoS]
6. Cooperative session establishment CooperativeSessionID
7. IF requested by NF, data collection from RAN, other NFs (optional)
8. Model retrieval
9. Sending ML model
10.a UE Inference on local data
10.b UE Inference on local data
11. Sending inferred data
12. Aggregating/ Inferred data
13. Transferriring aggregate result
14. UE Consumes result
15. UL PDU session: Sending aggregated inferred data
16. AF Consumes result
17. Notification
18. Model retraining
19. Model storage
20. DL PDU session: Transferring aggregated results
21. UE Consumes result

Figure A-4 Cooperative inference in subnetworks.

Cooperative inference in collaborative group

The high-level message exchange for cooperative inference in collaborative group is shown in Figure A-5. Compared to cooperative inference in SN, shown in Figure A-4, all UEs (UE 1-4) are globally distributed. Here, *CooperationSessionEstablishmentRequest* is followed by cooperative session establishment assuming UEs DL/UL sessions modifications according to cooperation requirements. UE2 and UE 3 infer local data on model obtained from NWDAF, or on local model (UE1 and UE2), and send inferred data to NWDAF. Locally inferred data is aggregated at NWDAF and sent to GH UE1, which sends the data to application consumer (via AF). Optionally, aggregated data can be used for model retraining in NWDAF and sent to UE4 and stored in ADRF. NWDAF sends aggregated data to UE4.

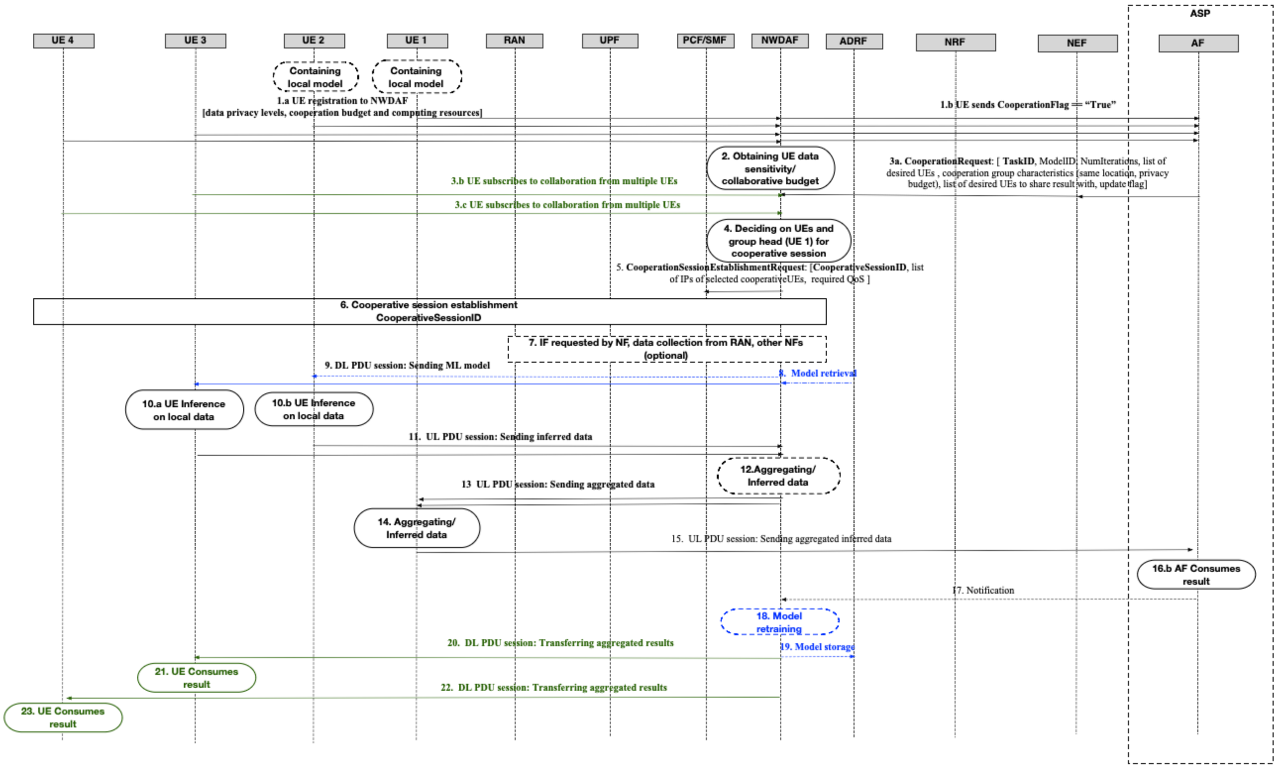

ASP
UE 4
UE 3
UE 2
UE 1
RAN
UPF
PCF/SMF
NWDAF
ADRF
NRF
NEF
AF
Containing local model
Containing local model
1.a UE registration to NWDAF [data privacy levels, cooperation budget and computing resources]
1.b UE sends CooperationFlag == "True"
2. Obtaining UE data sensitivity/ collaborative budget
3a. CooperationRequest: [TaskID, ModelID, NumIterations, list of desired UEs , cooperation group characteristics [same location, privacy budget), list of desired UEs to share result with, update flag]
3.b UE subscribes to collaboration from multiple UEs
3.c UE subscribes to collaboration from multiple UEs
4. Deciding on UEs and group head (UE 1) for cooperative session
5. CooperationSessionEstablishmentRequest: [CooperativeSessionID, list of IPs of selected cooperativeUEs, required QoS]
6. Cooperative session establishment CooperativeSessionID
7. IF requested by NF, data collection from RAN, other NFs (optional)
8. Model retrieval
9. DL PDU session: Sending ML model
10.a UE Inference on local data
10.b UE Inference on local data
11. UL PDU session: Sending inferred data
12.Aggregating/ Inferred data
13 UL PDU session: Sending aggregated data
14. Aggregating/ Inferred data
15. UL PDU session: Sending aggregated inferred data
16.b AF Consumes result
17. Notification
18. Model retraining
19. Model storage
20. DL PDU session: Transferring aggregated results
21. UE Consumes result
22. DL PDU session: Transferring aggregated results
23. UE Consumes result

Figure A-5 Cooperative inference in collaborative group.

A.3 AIaaS

A.3.1 AIaaS software prototype for distributed AI/ML services

With the aim of supporting highly distributed AI/ML services, like in the case of exposing federated learning capabilities and services, this implementation of the AIaaS platform described in section 3.1.3.4 has been enhanced with additional features. In particular, FEDaaS (Federation as a Service) and FLaaS (Federated Learning as a Service) solutions have been introduced and integrated in the AIaaS platform by integrating the open-source framework OpenFL [FSE+22]. Along with a Flask Federation Catalogue, OpenFL provides two abstractions that are well suited to implement FEDaaS and FLaaS in the AIaaS platform prototype:

Federation (FEDaaS): distributed set of machines composed of a director and several envoys in which the director represents the central point of the federation and coordinates the envoys in the distributed training; these components are long-life units that can be used to launch federated training experiments. A common data format is defined and shared across the federation using the DataShard abstraction: a data class that defines the data format shared across the federation and provides information about the data location in each of the envoys.

Experiment (FLaaS): represents a federated training experiment that is run over an existing federation. During an experiment the director creates an aggregator and each envoys create a collaborator; these are short-life components, with TTL equal to the experiment, that are actually participate in the distribute training. The role of each collaborator is performing local training using the data stored in the envoy while the role of the aggregator is to create global updates of the model aggregating the collaborators model updates.
The integration of OpenFL into the AIaaS platform prototype is performed by designing four different AI/ML pipelines (implemented as Prefect flows), as listed in the following table Table A-3.

Table A-3: AIaaS AI/ML pipelines for FEDaas and FLaaS

AI/ML Pipeline	Description

Start Director	It setups and starts a director node as service on the host (e.g. as a process in a virtual machine, or as a pod in a Kubernetes cluster). Input Parameters: Federation Name, DataShard.
Start Envoy	It setups and starts an envoy as service on the host (e.g. as a process in a virtual machine, or as a pod in a Kubernetes cluster) linking it to an existing director. Input Parameters: Director FQDN, DataShard.
Start Federation	It embeds the start director and start envoy flows as sub-flows to setup and start a complete federation, saving its information in the Federation Catalogue. Input Parameters: Federation Name, Director FQDN, Number of envoys, Shard Descriptor.
Run Experiment	It runs federated training over an existing federation resulting in a new trained model in the catalogue. The experiment can be performed to train a new model provided by the user or to train a new version of a model that is already available in the ML catalogue (providing the URI of the model). Input Parameters: Federation Name, (opt) Model URI, Aggregator, Rounds.

To actually setup a federation, when the Start Federation pipeline is run, as first step two different Prefect work pools are created: one for the Director of the federation and another for the Envoys. Once these work pools are created, a Prefect worker (as process on bare metal or pod in a kubernetes cluster) that is already running on the Director machine is linked to the Director work pool. This worker will pull and run the Start Director flow, spawning a Director on the host machine and saving the information of the Federation in the Federation Catalogue. Now that the director is up and running, several Prefect workers that are already running on envoys machines and linked to the envoys work pool, will pull and run the Start Envoy flow. This process will spawn an envoy, connected to the director, on each of the envoy's machine. The whole sequence of actions is depicted in Figure A-6 while a high-level representation of the setting and the workflow is represented in Figure A-7.

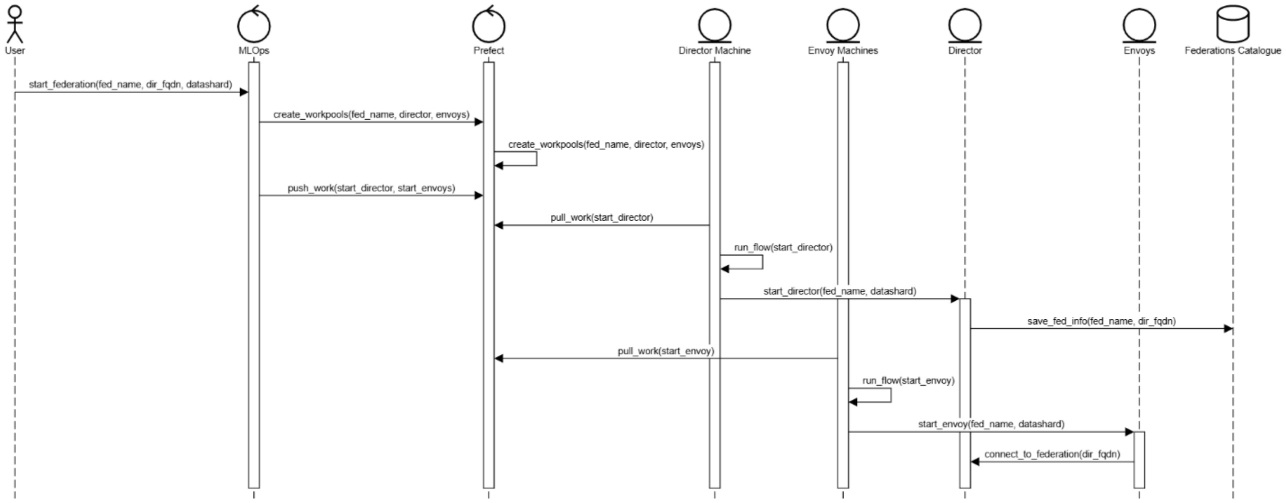

User
MLOps
Prefect
Director Machine
Envoy Machines
Director
Envoys
Federations Catalogue
start_federation(fed_name, dir_fqdn, datashard)
create_workpools(fed_name, director, envoys)
create_workpools(fed_name, director, envoys)
push_work(start_director, start_envoys)
pull_work(start_director)
run_flow(start_director)
start_director(fed_name, datashard)
save_fed_info(fed_name, dir_fqdn)
pull_work(start_envoy)
run_flow(start_envoy)
start_envoy(fed_name, datashard)
connect_to_federation(dir_fqdn)

Figure A-6: Federation (FEDaaS) Setup sequence diagram

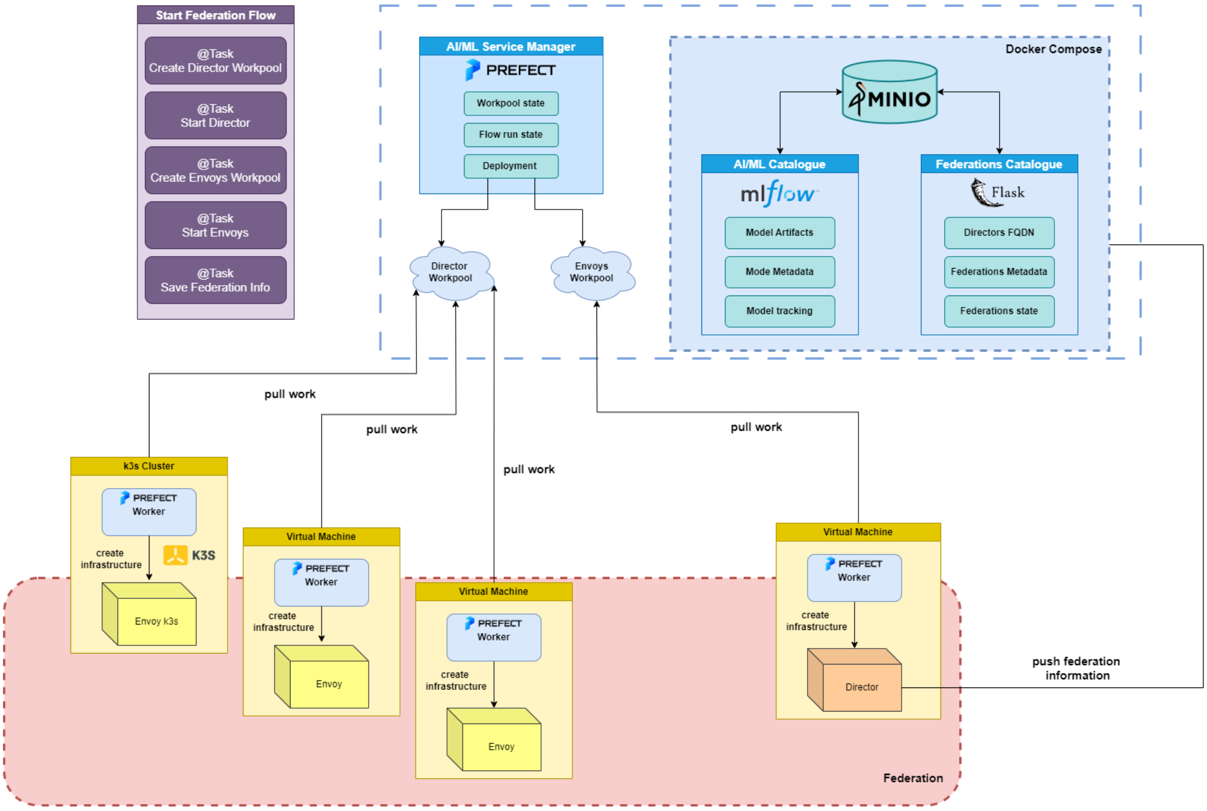


Figure A-7: Federation (FEDaaS) Setup high level representation

On the other hand, to run a federated training over an existing federation, a Start Experiment pipeline is run. As first step, an experiment starter work pool is created: the worker that will link to this work pool is the one that contains the necessary data to validate the model after training. Once the flow starts, the first step is the definition of the model that will be used in the experiment: this model can be provided explicitly by the user or can be pulled from the models' catalogue providing a model URI. Along with the model, the information about the federation is pulled from the federations catalogue. The model and the along with all the other needed parameters (aggregator to be used, number of rounds) are provided to the director which will start and coordinate the federated training. Once the number of rounds is reached a version of the model trained on the federations data is provided to the experiment starter, which will validate it over its own validation data. Finally, if the model satisfies an acceptance threshold (e.g. in terms of validation of model accuracy, or other performance attributes), it is saved into the models' catalogue. The whole sequence of actions is depicted in Figure A-8 while a high-level representation of the setting and the workflow is represented in Figure A-9.

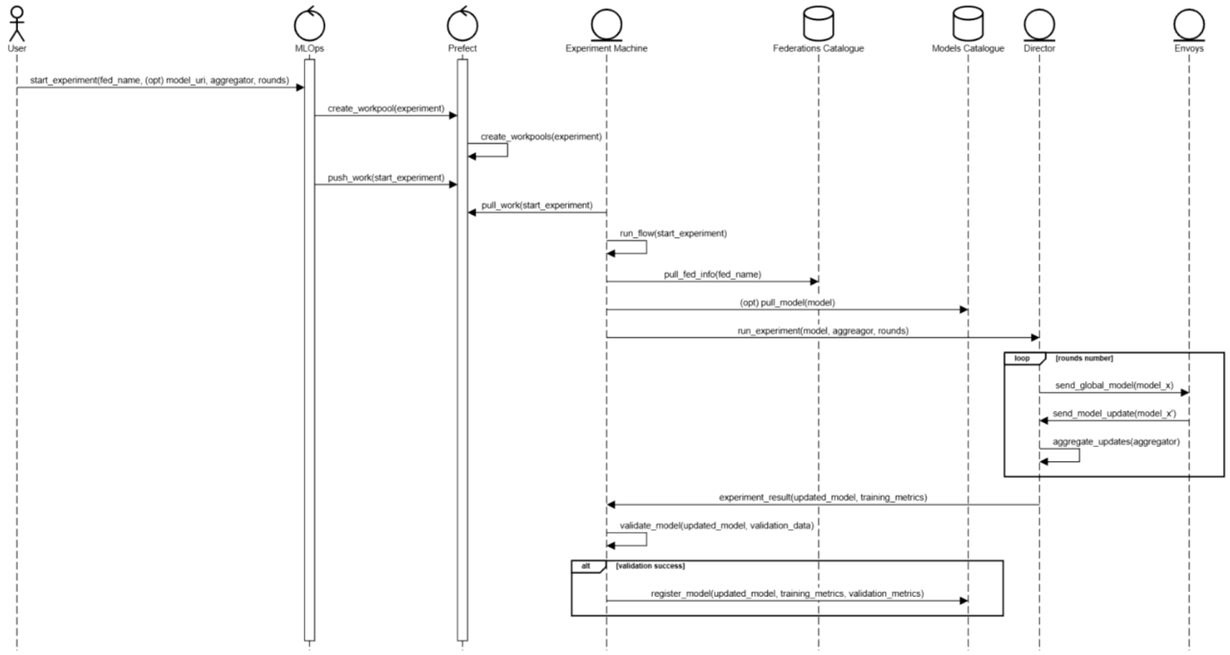

User
MLOps
Prefect
Experiment Machine
Federations Catalogue
Models Catalogue
Director
Envoys
start_experiment(fed_name, (opt) model_uri, aggregator, rounds)
create_workpool(experiment)
create_workpools(experiment)
push_work(start_experiment)
pull_work(start_experiment)
run_flow(start_experiment)
pull_fed_info(fed_name)
(opt) pull_model(model)
run_experiment(model, aggreagor, rounds)
loop
[rounds number]
send_global_model(model_x)
send_model_update(model_x')
aggregate_updates(aggregator)
experiment_result(updated_model, training_metrics)
validate_model(updated_model, validation_data)
alt
[validation success]
register_model(updated_model, training_metrics, validation_metrics)

Figure A-8: Experiment (FLaaS) sequence diagram

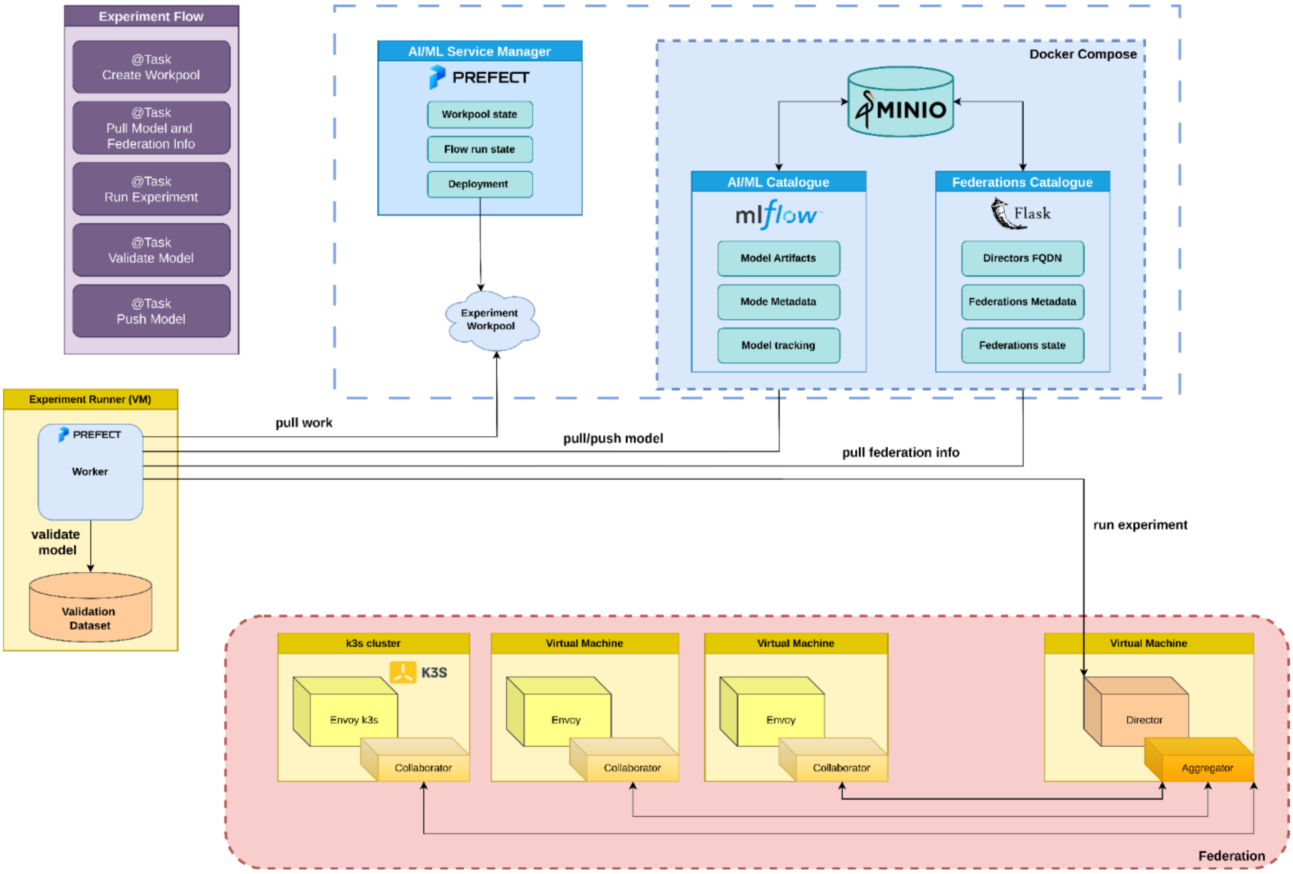


Figure A-9: Experiment (FLaaS) high level representation

A.4 Compute

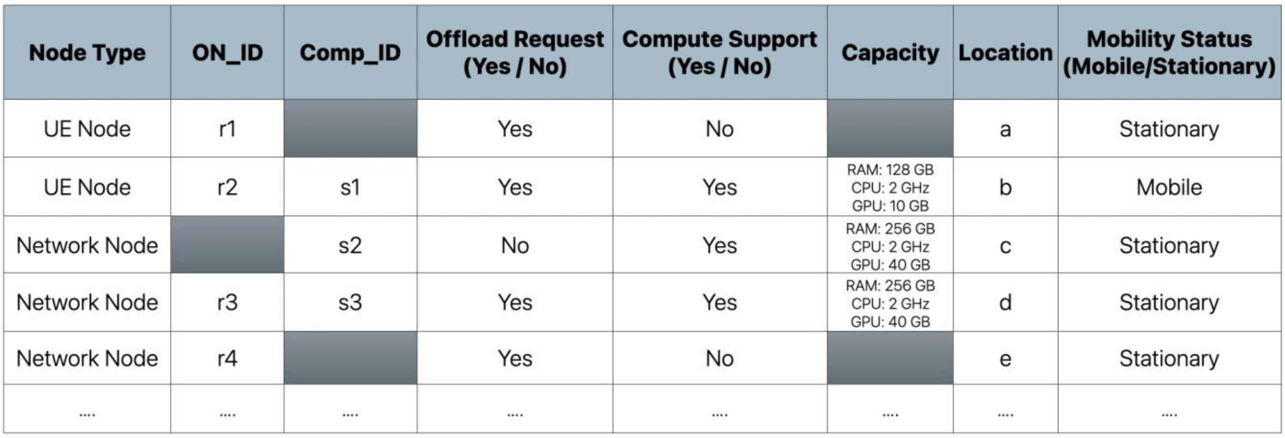
A.4.1 High-level procedures for compute offloading and service layers

The general architecture and novel functional entities for computational offloading are presented in [HEX223-D32]. The main introduced functional components are: Offloading Node (ON), a network node having a compute task to be offloaded, Computing Node (CompN), a network node with certain compute capability, Compute Offload Controlling Node (CCN), a network node that collects all compute capabilities from all available CompNs and makes compute offload decision based on their current load, and capacity requirements. and optional Routing Node (RN) for routing computation loads and results between ON(s) and CompN(s).

Compute services

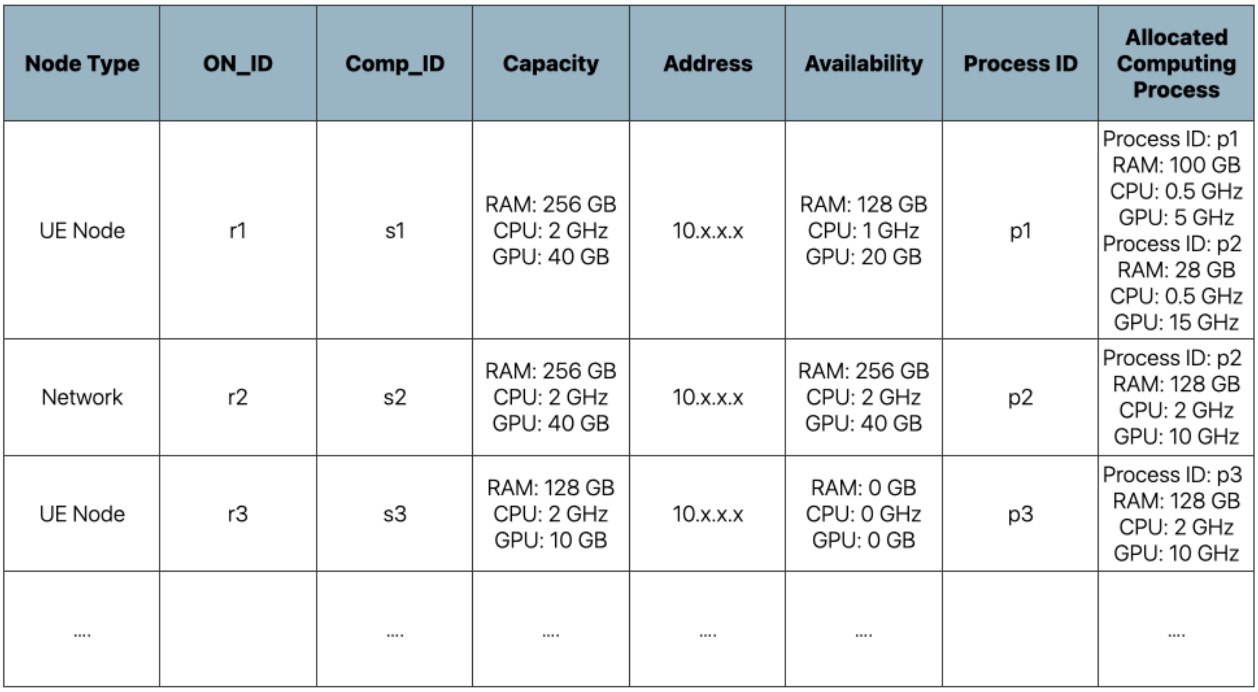
An example of CRMS database is shown in Table A-4. Based on CRMS database, CPMS, responsible for process and event management, selects the compute nodes (i.e., either Network nodes or UE nodes). Available capacity, location and mobility status define the basis for the offloading selection algorithm via CPMS, e.g., by assisting with the latency and jitter estimation for such selection. CPMS on CCN maintains the database with the list of nodes that support computing along with their resource availability status. An example of protentional CPMS database is shown in Table A-5. When a new offloading request arrives from an ON, CPMS on CCN checks its database and chooses the suitable CompN(s) to provide the offloading service based on the latency, e.g., choosing the CompN(s) closest to the ON, and capacity, e.g., by requesting the offloading process against the availability of CPU/GPU/RAM capability in the CompN(s).

Table A-4: An example of CRMS database used for CPMS process and event management.



Node Type	ON_ID	Comp_ID	Offload Request (Yes / No)	Compute Support (Yes / No)	Capacity	Location	Mobility Status (Mobile/Stationary)
UE Node	r1		Yes	No		a	Stationary
UE Node	r2	s1	Yes	Yes	RAM: 128 GB CPU: 2 GHz GPU: 10 GB	b	Mobile
Network Node		s2	No	Yes	RAM: 256 GB CPU: 2 GHz GPU: 40 GB	c	Stationary
Network Node	r3	s3	Yes	Yes	RAM: 256 GB CPU: 2 GHz GPU: 40 GB	d	Stationary
Network Node	r4		Yes	No		e	Stationary
….	….	….	….	….	….	….	….

Table A-5: An example of CPMS database.



Node Type	ON_ID	Comp_ID	Capacity	Address	Availability	Process ID	Allocated Computing Process
UE Node	r1	s1	RAM: 256 GB CPU: 2 GHz GPU: 40 GB	10.x.x.x	RAM: 128 GB CPU: 1 GHz GPU: 20 GB	p1	Process ID: p1 RAM: 100 GB CPU: 0.5 GHz GPU: 5 GHz Process ID: p2 RAM: 28 GB CPU: 0.5 GHz GPU: 15 GHz
Network	r2	s2	RAM: 256 GB CPU: 2 GHz GPU: 40 GB	10.x.x.x	RAM: 256 GB CPU: 2 GHz GPU: 40 GB	p2	Process ID: p2 RAM: 128 GB CPU: 2 GHz GPU: 10 GHz
UE Node	r3	s3	RAM: 128 GB CPU: 2 GHz GPU: 10 GB	10.x.x.x	RAM: 0 GB CPU: 0 GHz GPU: 0 GB	p3	Process ID: p3 RAM: 128 GB CPU: 2 GHz GPU: 10 GHz
....		....	....	....	....		....

High-level message exchange for compute offloading

Depending on whether the compute results size Z [bytes] is fixed or variable, there are two variants for the realization of the Computational Offload Procedure. In both cases, the compute task (i.e., Y [bytes]) could be sent via dynamic or configured scheduling. If the compute results (i.e., Z [bytes]) are foreseen to have fixed size, they may be sent via *configured* scheduling, such that DL transmission for compute results is planned between **ON** and **CompN** before compute results are ready. This approach enables increased sleep time of the ON, thus resulting in energy efficiency, and is further detailed in B.4.

If the compute results (i.e., Z [bytes]) are foreseen to have variable size, they may be sent via *dynamic* scheduling, such that the DL transmission is controlled by ON, which sends a Pull Request when ready to receive the compute results of size Z [bytes], as illustrated in Figure A-10. Moreover, ON may also request a certain DL time window to assist the CompN in resource management and monitor the window for reception of compute results. There might be also several HARQ retransmissions on that planned DL of compute results. If the compute results are not ready at CompN within the T [ms], they are discarded, the offloading fails and CompN indicates to ON that response is not ready. If the stored compute results are not pulled by ON within T [ms], they could also be discarded. After the task is complete and the results are delivered, the status of CompN can be updated to indicate its readiness for another task.

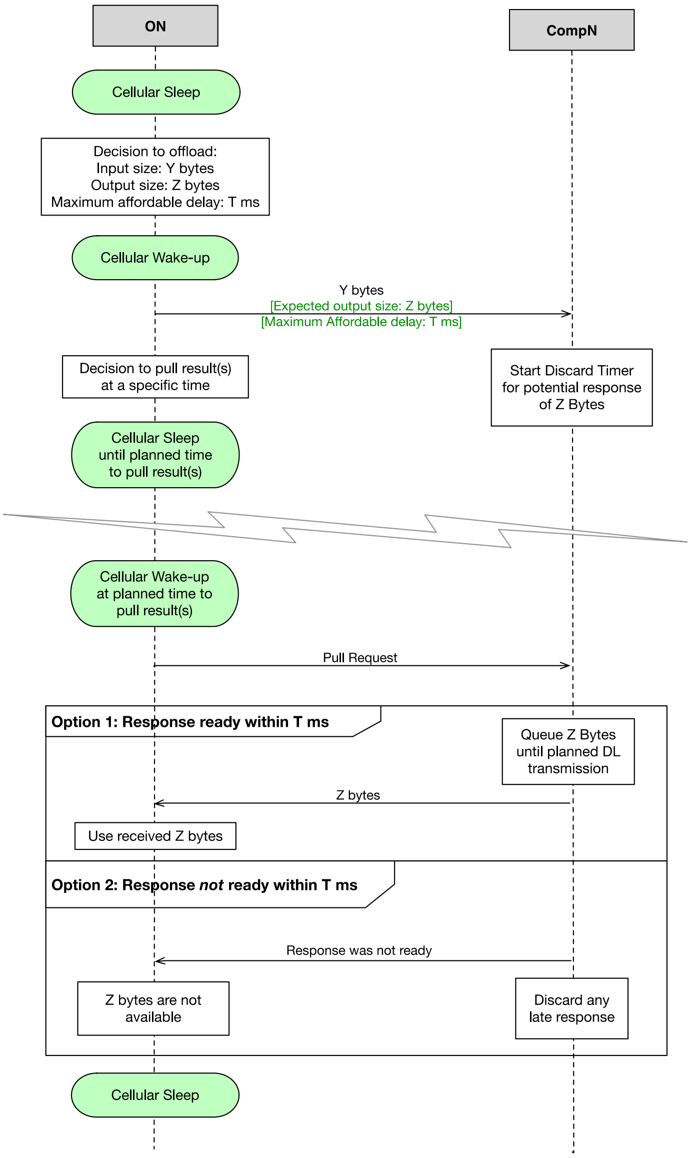

ON
CompN
Cellular Sleep
Decision to offload:
Input size: Y bytes
Output size: Z bytes
Maximum affordable delay: T ms
Cellular Wake-up
Y bytes
[Expected output size: Z bytes]
[Maximum Affordable delay: T ms]
Decision to pull result(s) at a specific time
Start Discard Timer for potential response of Z Bytes
Cellular Sleep until planned time to pull result(s)
Cellular Wake-up at planned time to pull result(s)
Pull Request
Option 1: Response ready within T ms
Queue Z Bytes until planned DL transmission
Z bytes
Use received Z bytes
Option 2: Response not ready within T ms
Response was not ready
Z bytes are not available
Discard any late response
Cellular Sleep

Figure A-10: An example of flow and message content exchange for pulled DL.

A.5 Multi-connectivity

This section provides more details about the topics analysed in Section 4.2.

A.5.1 CA/DC Evolution

As mentioned in Section 4.2.2.3, the RRC Reestablishment procedure is associated with service interruption time, since the UE shall select a new PCell, perform the re-establishment procedure and may also need to perform security authentication with the target PCell, as illustrated in Figure A-11.

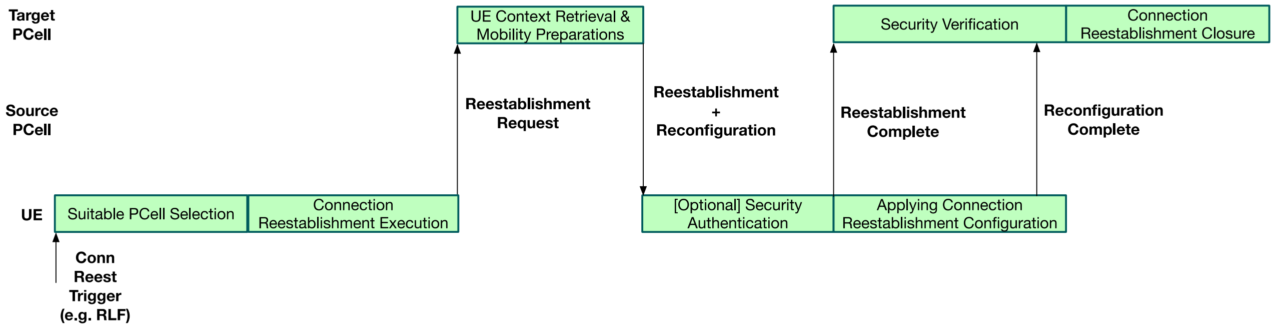

Target PCell
Source PCell
UE
Suitable PCell Selection
Connection Reestablishment Execution
Conn Reest Trigger (e.g. RLF)
Reestablishment Request
UE Context Retrieval & Mobility Preparations
Reestablishment + Reconfiguration
[Optional] Security Authentication
Reestablishment Complete
Security Verification
Applying Connection Reestablishment Configuration
Reconfiguration Complete
Connection Reestablishment Closure

Figure A-11 Connection re-establishment procedure

Figure A-12 shows a flowchart describing the logic at the UE for an SCell-aided fast PCell recovery procedure from the moment UE RRC receives the RLF indication up to the moment when the SCell becomes the PCell, or a handover is performed to a different target cell, or the RRC Re-establishment procedure is initiated. Regarding the transmission of the failure information, if a configured UL grant is active at the SCell, it could be used for the transmission. Otherwise, if PUCCH has been configured for sending a scheduling request at the SCell, it could be used for the UE to receive an UL grant. If the SCell is not in a Supplementary DL (SDL) band and if no configured UL grant is active and no PUCCH has been configured, a random-access (RA) procedure could be initiated in the SCell for the UE to receive an UL grant. After the failure information has been transmitted, the UE may start a timer during which it monitors the SCell for an indication or RRC reconfiguration for the new PCell. If the timer expires and no indication has been received, then the RRC Re-establishment procedure is initiated.

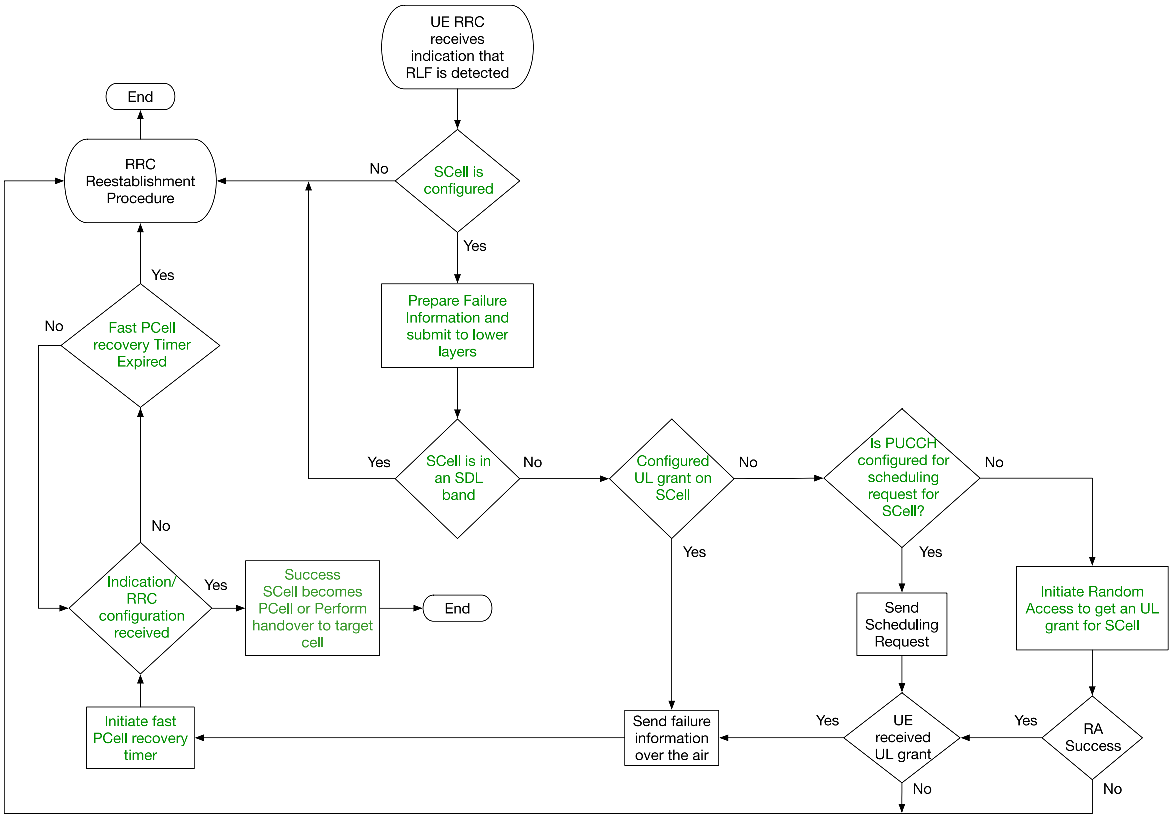

UE RRC receives indication that RLF is detected
SCell is configured
No
Yes
End
RRC Reestablishment Procedure
Prepare Failure Information and submit to lower layers
Fast PCell recovery Timer Expired
Yes
No
SCell is in an SDL band
Yes
No
Configured UL grant on SCell
No
Yes
Is PUCCH configured for scheduling request for SCell?
No
Yes
Send Scheduling Request
Initiate Random Access to get an UL grant for SCell
Indication/ RRC configuration received
Yes
No
Success SCell becomes PCell or Perform handover to target cell
End
Initiate fast PCell recovery timer
Send failure information over the air
UE received UL grant
Yes
No
RA Success
Yes
No

Figure A-12 Flow chart with the UE logic for the SCell-aided fast PCell recovery procedure

A.5.2 Aggregation of different access networks

A message sequence chart representing an example of WCA operation with a WT and a DL transmission is provided in Figure A-13. As a first step, the WT discovery takes place. One option is for the UE to detect the available WTs in its vicinity and provide the NW with a list of down-selected WTs (i.e., trusted/untrusted), requesting a NW configuration. The NW would then select a subset of the obtained list of WTs and preconfigure the UE with all necessary information about those WTs, such as the NW-selected list of trusted and untrusted WTs and the split RB configuration. Afterwards, the UE would select a WT, and the connection establishment procedure would start.

To establish a connection, the UE selects the specific WT from an available list. The UE may communicate that decision to the BS to support the use case where DL transmission happens first. For UL transmissions, the UE can connect and send UL to a WT without coordinating again with the BS. However, in DL, the BS requires the UE WLAN ID, which may or may not be explicitly sent by the UE to the BS prior to the need for DL transmission. If the UE has not sent its UE WLAN ID to the BS, then the DL connection cannot be used until the first UL packet has been received. The reason is that in the first UL packet sent via the WT, the WT will include the UE WLAN ID in the WRAP header. Afterwards, the BS can create and maintain a UE cellular ID – UE WLAN ID map, which will be used in DL.

After the connection has been established and when a DL packet arrives at the BS, it is mapped to a UE-BS RB. The packet can already be transmitted at this point via the direct BS-UE path. At the same time, the WRAP layers of the BS, the WT and the UE operate as described in Figure A-13 to convey the packet over the BS-WT-UE path.

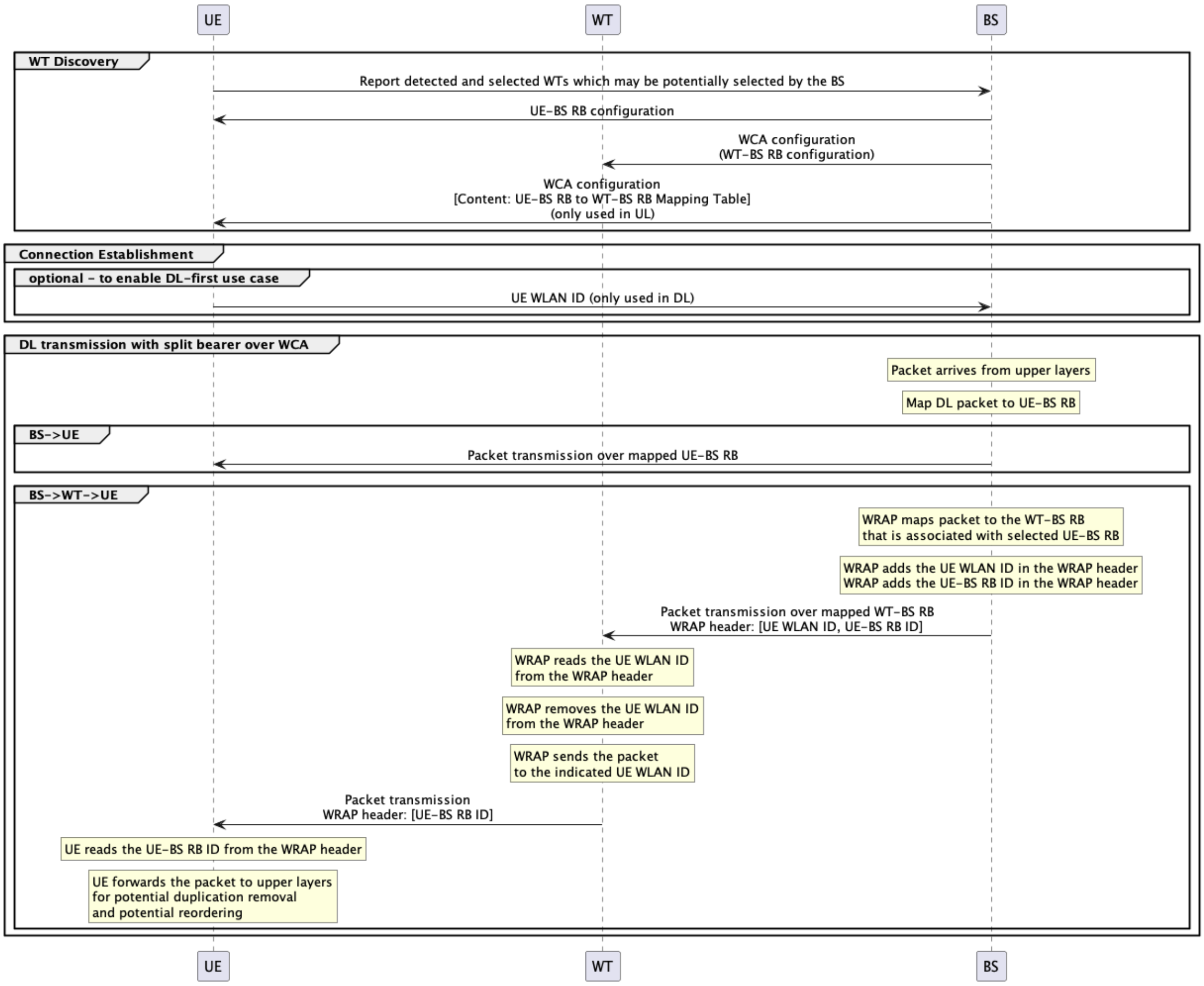

UE
WT
BS
WT Discovery
Report detected and selected WTs which may be potentially selected by the BS
UE–BS RB configuration
WCA configuration
(WT–BS RB configuration)
WCA configuration
[Content: UE–BS RB to WT–BS RB Mapping Table]
(only used in UL)
Connection Establishment
optional – to enable DL–first use case
UE WLAN ID (only used in DL)
DL transmission with split bearer over WCA
Packet arrives from upper layers
Map DL packet to UE–BS RB
BS–>UE
Packet transmission over mapped UE–BS RB
BS–>WT–>UE
WRAP maps packet to the WT–BS RB
that is associated with selected UE–BS RB
WRAP adds the UE WLAN ID in the WRAP header
WRAP adds the UE–BS RB ID in the WRAP header
Packet transmission over mapped WT–BS RB
WRAP header: [UE WLAN ID, UE–BS RB ID]
WRAP reads the UE WLAN ID
from the WRAP header
WRAP removes the UE WLAN ID
from the WRAP header
WRAP sends the packet
to the indicated UE WLAN ID
Packet transmission
WRAP header: [UE–BS RB ID]
UE reads the UE–BS RB ID from the WRAP header
UE forwards the packet to upper layers
for potential duplication removal
and potential reordering
UE
WT
BS

Figure A-13 WLAN-Cellular Aggregation DL with WLAN Terminal

The WCA is a different solution that the LTE-WLAN Aggregation (LWA) [36.300]. More specifically, LWA supports both co-located WT and eNB, as well as non-co-located WT and eNB by using the Xw interface between the WT and the eNB in the latter case. In contrast to that, in WCA, the WT is always in cellular coverage of the BS and it uses the cellular interface to connect to the BS. This allows for a greater flexibility in the deployment and makes it more homogeneous with the current technology direction, where the cellular interface is used for intermediate hops, as for example in IAB [38.300]. During data transfer, the WT in LWA may be configured multiple E-UTRAN Radio Access Bearers (E-RAB) – UE IDs mappings, but at any transmission over the Xw interface, only data of a single E-RAB is transmitted. Therefore, there is no multiplexing of different E-RABs in a single data transfer, which can only carry User Plane (UP) data. In WCA, different options are possible for mapping the UE IDs to the RBs, based on the capabilities and configurations of the WT, and multiplexing of different UE and UE-BS RBs is possible in a single data transfer. Also, both UP and CP data is supported to be transmitted via the WT. In general, LWA shall be setup in a connection-switch manner (mapping E-RABs to UEs), while WCA proposes a more flexible, packet-switch approach, since it relies on in-band signalling.

The WCA also differs from the LTE-WLAN radio level Integration using IPsec tunnel (LWIP) [36.300]. While WCA aggregates the packets in the PDCP layer, LWIP aggregates the packets on the supported LWIP Encapsulation Protocol (LWIPEP) layer. A header with a newly generated sequence number and a DRB ID is added to the IP PDU at the transmit entity, regardless of the path it will follow. The header is removed at the receive LWIPEP entity. In LWIP, the IP packets transferred between the UE and the LWIP-Security Gateway (LWIP-SeGW) are encapsulated using IPsec, to provide security to the packets that are transmitted over WLAN, while in WCA the packets transmitted both over cellular and over WLAN may use cellular security. One main difference is also that, in LWIP, packet re-ordering happens at two places; once in the receive RLC entity for the cellular packets, as well as in the LWIPEP receive entity for the packets received via the cellular and the IP tunnel. The RLC entity reordering is redundant in this case, but it cannot be disabled. On the other hand, packet re-ordering in WCA happens only at a single place; the receive PDCP entity, which reorders the packets received via the cellular and the WLAN paths.

A.6 Subnetworks

This section provides more details about the topics analysed in Section 4.3.3.1.

A.6.1 Subnetwork Control Plane (snCP)

The first option is when the UE CP terminates at the UE, as shown in Figure A-14. In this option, the UE (i.e., UE3 in the Figure A-14) supports the full cellular CP. This allows any CP data to be forwarded from the BS to the UEs via the MgtN. The UE may also support the separate snCP, which is for subnetwork-specific procedures.

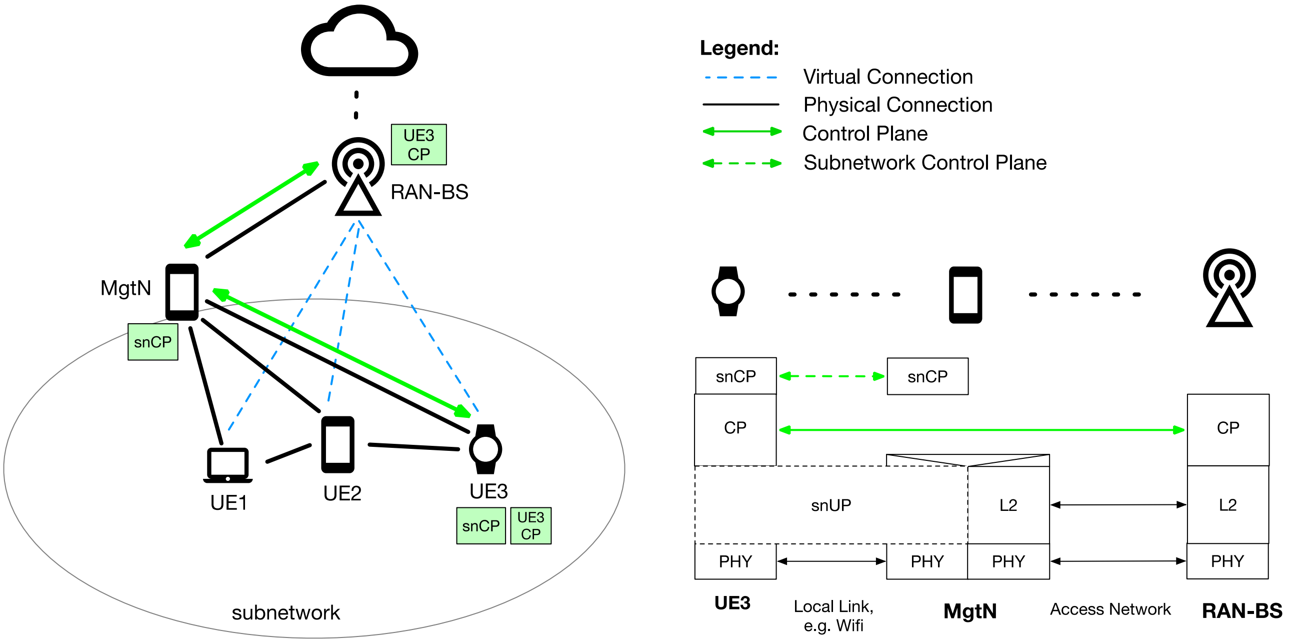


Figure A-14 UE Control Plane terminates at the UE

The second option is when the CP of the UEs is aggregated at the MgtN, as presented in Figure A-15. In this option, the UEs offload their CP or part of it towards the MgtN. At the same time, the SubNW uses the snCP between the UE and the MgtN. As in the first option, the snCP is transparent to the NW and includes the configuration and procedures within the local subnetwork, as well as the offloaded UE CP information. However, in this case it also includes all procedures that are necessary for the UE to operate, since it is not required by the UE to support the full cellular CP.

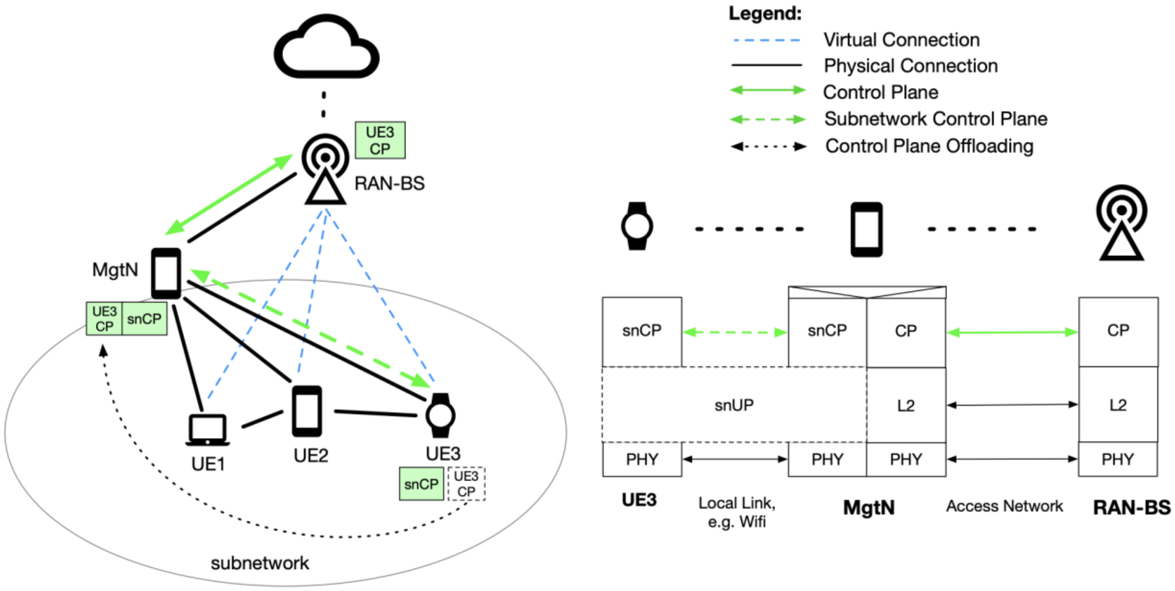


Figure A-15 MgtN aggregates the CP of the UEs

The third option is when the UE offloads its CP to another UE without the BS's awareness, as illustrated in Figure A-16. In more detail, the target UE has offloaded its CP to the MgtN without BS awareness. In Step 1, the BS communicates with the UE.

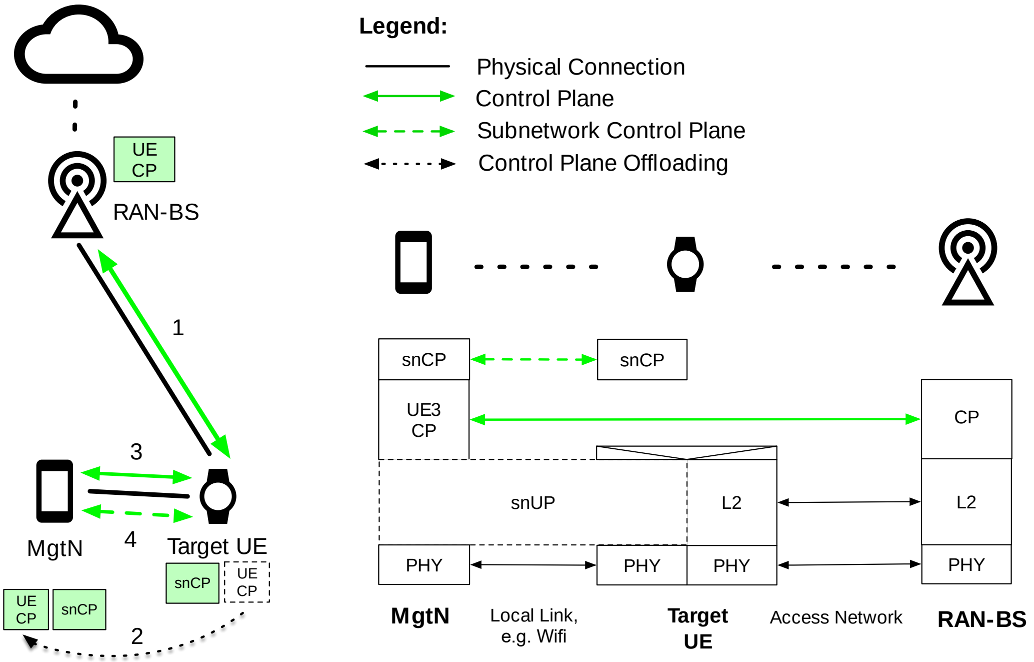


Figure A-16 UE CP offloaded to another UE

Step 2 indicates that the CP of the target UE has been offloaded to the MgtN. When the target UE receives a CP message from the BS, it forwards it to the MgtN (Step 3), which in turn decodes it and sends back to the target UE only the relevant information via the snCP (Step 4). This enables lower capability UEs and/or power saving by letting more capable UEs in the SubNW do the heavy lifting. The fact that the target UE does not support a full cellular CP and supports only snCP is abstracted from the BS.

A.6.2 MgtN-aided RRC configuration

The L-RRC could be either defined by 3GPP or by each device manufacturer. If it is defined in 3GPP, then the 3GPP specifications shall define a common set of features for all types of devices and specialized features for each category of devices (e.g., sensors, VR/AR glasses, etc.). Therefore, specifications shall define different L-RRC versions, where each L-RRC specification shall be used by a certain device category. The advantage of defining the L-RRC in 3GPP is that there will be no interoperability issues in communication across different devices implementing the L-RRC. On the other hand, if the L-RRC is defined by each device manufacturer, then each manufacturer shall define their own specifications according to the supported/not-supported features of their produced devices. In that case, a device manufacturer shall produce both the MgtN and the UE, so that its co-located devices can communicate using the proprietary L-RRC specifications. The advantage of this approach is that since specifications would be defined by each device manufacturer, then the L-RRC logic can be simplified as much as possible to exactly match the devices' supported/not-supported features.

A.7 Context-aware management

A.7.1 Abstraction technique applied to an exemplary meshed network

This section provides more details about the topics analysed in Section 4.4.2. The key aspect for abstraction is to defines mechanisms that allow combining the scalability/stability of the solutions with the good performances.

To meet scalability/stability, the abstraction view should be dynamically update on a long-time frame and not for any service request, anyway, to meet optimization of resources, the abstraction should follow actual traffic behaviour. In the following a method of basket is introduced that aim to meet both the previous requirements.

Figure A-17 presents a transport network where each link between nodes (e.g., A, B, C) is characterized by parameters which help determine the most efficient paths for data to travel across the network. In this exemplary network with 5 nodes, each link from a node X to a node Y is characterized by three parameters: a cost parameter, the link supported bandwidth (throughput), and the maximum latency.

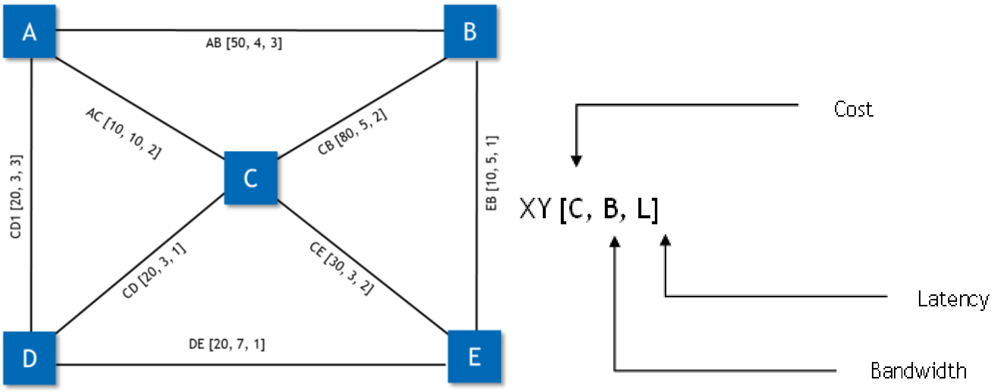


Figure A-17 Exemplary meshed network with five nodes

The definition of "cost" is out of scope of this example. Cost and latency are considered cumulative across links while the bandwidth of a sequence of links is equal to the minimum bandwidth among such links.

Consider not, as in Figure A-18, to focus on connectivity from A to B, which are edge nodes of the physical mesh. A path computation engine determines K physical paths between A and B.

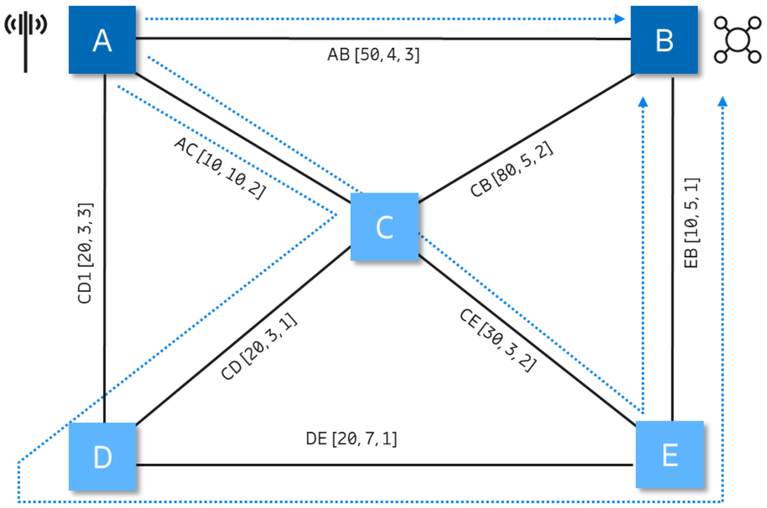


Figure A-18 Possible paths between node A and node B

Assuming K = 3:

- AB [50, 4, 3] via AB

- AB [50, 3, 5] via AC - CE - EB[- AB [60, 3, 5] via AC – CD – DE - EB[The three physical paths can be advertised, to the higher layers, as one or more baskets of possible virtual links characterized by given costs, bandwidths, latencies. There are different alternatives for aggregation of the three physical paths in the virtual links that will populate the basket as reported in the Figure A-19.

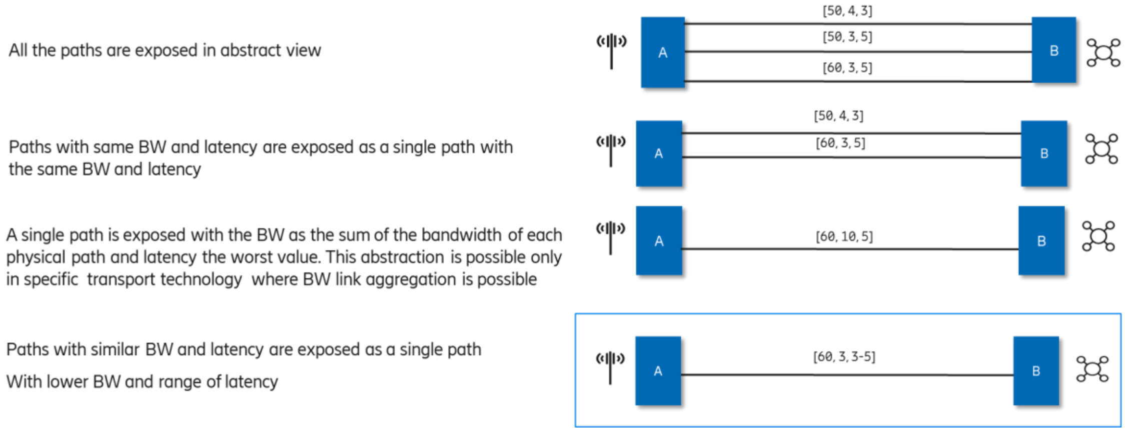

All the paths are exposed in abstract view
[50, 4, 3]
[50, 3, 5]
[60, 3, 5]
A
B
Paths with same BW and latency are exposed as a single path with the same BW and latency
[50, 4, 3]
[60, 3, 5]
A
B
A single path is exposed with the BW as the sum of the bandwidth of each physical path and latency the worst value. This abstraction is possible only in specific transport technology where BW link aggregation is possible
[60, 10, 5]
A
B
Paths with similar BW and latency are exposed as a single path
With lower BW and range of latency
[60, 3, 3-5]
A
B

Figure A-19 Alternative options for aggregation in virtual links

Suppose to advertise the connectivity in the fourth way: paths with similar bandwidth and latency are exposed as a single path. Then, as reported in Figure A-20 consider the case in which a new traffic request shall be accommodated. It is demanded with bandwidth = 1 and latency < 6. The first abstracted virtual link is used to serve the request and the related available bandwidth is scaled from 4 to 3. We assume that the latency of the virtual link is not impacted by the new traffic flowing in the network. The current abstraction scenario remains stable, and it’s not affected in terms of available virtual links. The basket preserves its content.

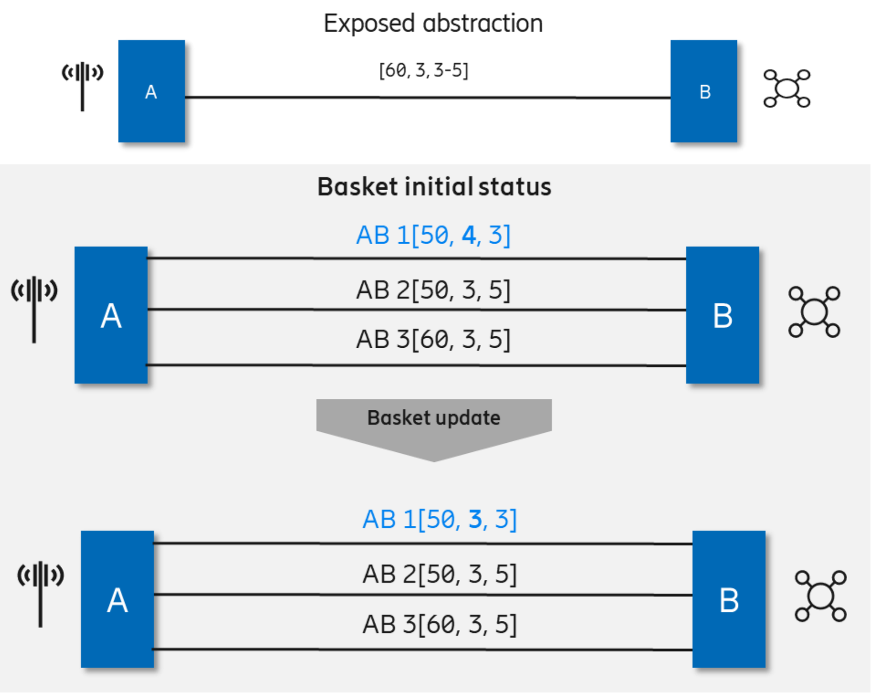


Figure A-20 Connectivity basket update after accommodated traffic

By maintaining the current abstraction without updating it (e.g., searching for new available physical paths), the solution becomes more scalable. However, if a new traffic flow with a requested bandwidth of 4 is submitted, it may not be served even if the necessary physical connectivity is available.

Resiliency is one of the key aspects for future service and it could be part of the requirements of the service itself. For such reason it is useful to include the resiliency capability of the transport network in the abstraction view. The abstraction view can include additional parameters such as resiliency. The algorithm to associate the physical path to the resiliency type depends on the transport technology: e.g., in case of packet nodes it is possible to have a secondary path for fault recovery shared with other traffic because the switching time is quite fast, while in case of optical is preferred to have dedicated optical channel for working and protection. In the Figure A-21 an example is reported.

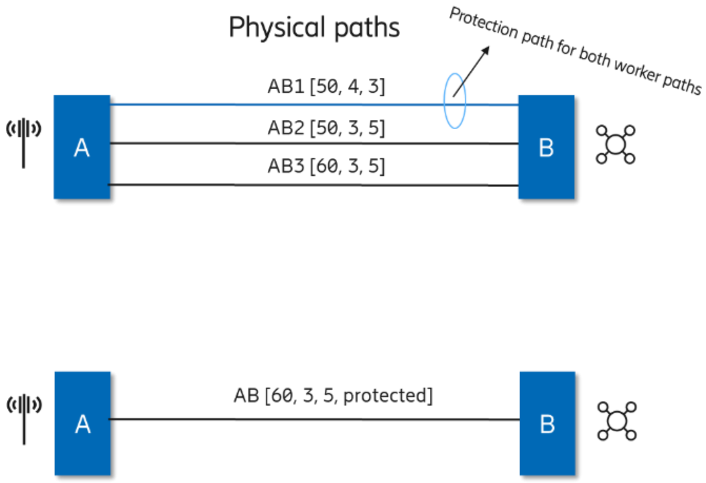


Figure A-21 Abstraction of a protected link

A.7.2 Transport Network in the end-to-end scenario for resiliency support.

This section provides more details about the topics analysed in Section 4.4.2 regarding the transport network resilience. Figure A-22 provides a basic scenario which frames transport in the end-to-end (E2E) context. Here, a “site” is constituted by servers/nodes supporting a set of virtual and/or physical network functions: CNF or containers, Virtual Network Functions (VNF), and Physical Network Functions (PNF). Transport connects such functions both inside radio sites (intra-site transport) and between different radio sites (inter-site transport).

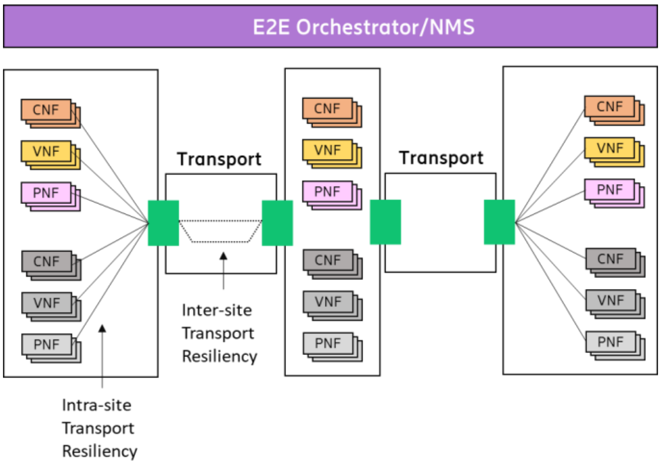

E2E Orchestrator/NMS
CNF
VNF
PNF
CNF
VNF
PNF
Transport
Transport
Inter-site Transport Resiliency
Intra-site Transport Resiliency

Figure A-22 End to End architecture - Basic Scenario

Redundancy operates at the function level (i.e., duplicated function) and at the connection level (e.g., alternative paths). It’s appropriate to define intra-site recovery, to protect from internal faults, and inter-site recovery, to protect from faults among radio sites, for example by duplicating the entities to be protected. Note that such protected entities are specific functions, rather than the entire node (server/physical node).

Transport must be considered as an integral part of E2E scenarios, where relevant transport connections are defined by a set of demarcation points. In typical E2E scenarios, these demarcation points include the antenna site and the core system. However, the specific points may vary depending on the deployment of radio and core functions within the network architecture.

It is important to note that various demarcation points may exist within a network. These points can be associated with one or more user plane paths related to a service, such as connections between antenna sites and other sites hosting user plane functions—for example, UPFs in a 5G system.

Additionally, there may be demarcation points linked to control plane traffic that supports the user plane, which may not necessarily coincide with user plane demarcation points. These include connections between antenna sites and sites hosting core NFs, such as the AMF in a 5G system, as well as connections between sites hosting CN-related functionalities.

Using the 5G system as an example, demarcation points could exist between AMF(s) and Session Management Functions (SMFs), as well as between SMF and UPF(s). The following are some notable examples where resiliency affects both intra-site and inter-site connectivity recovery. While not exhaustive, this list represents relevant cases.

Figure A-23 illustrates an exemplary scenario where two radio sites (e.g., antenna site, Baseband site, etc. including the case of sites where a mix of RAN and CN functions are located) are connected through a transport network. Under normal conditions (i.e., no faults), the primary functions of the first radio site, located on the left-hand side of the figure, are connected to the primary functions of the second radio site, located on the right-hand side, through a primary 'logical' link between two border nodes (demarcation points) of the transport network.

)

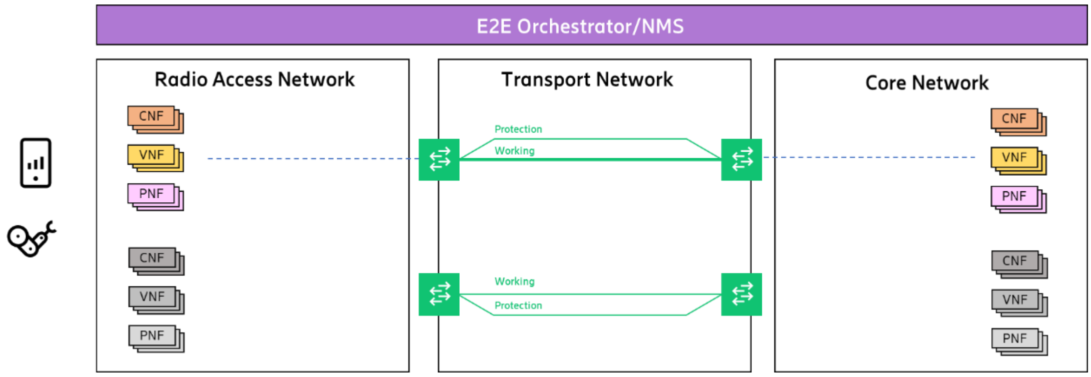


Figure A-23 Exemplary radio-transport network with primary and protection paths on the same demarcation points

Exemplary intra-site recovery scenarios are illustrated in Figure A-23 and Figure A-24 which a server failure impacts the functions of the first radio site, causing the corresponding backup functions to activate.

In Figure A-24, radio traffic is rerouted through the same pair of demarcation points in the transport network to reach the primary functions of the second radio site.

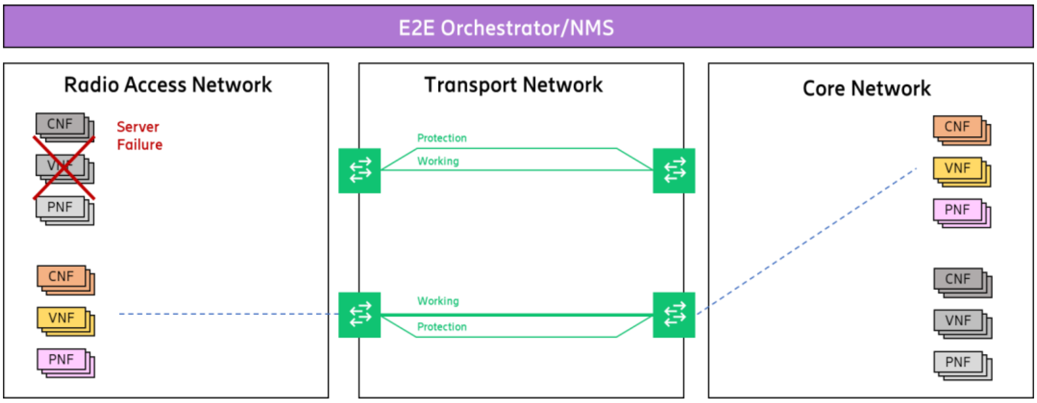


Figure A-24 Intra-site recovery with common demarcation points

In Figure A-25, the radio traffic is rerouted through a different pair of demarcation points in the transport network to reach the primary functions of the second radio site.

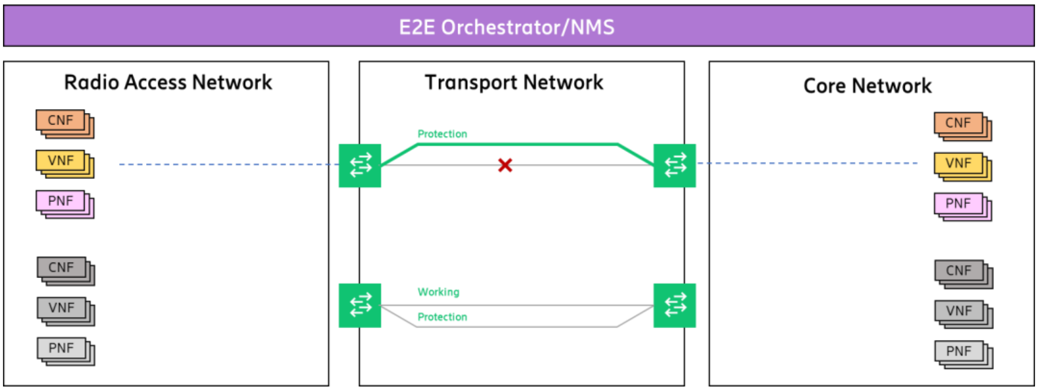


Figure A-25 Intra-site recovery with alternative demarcation points

As previously mentioned, various resiliency techniques are employed to handle failures in a transport network. The choice of technique depends on the transport technology in use (e.g., packet, optical, etc.). These techniques can be implemented in either the control plane or the data plane.

This separation of roles allows for simple and fast data plane processing. Historically, resilience to network failures was managed in the control plane: ensuring connectivity was considered the responsibility of the control plane, while the data plane focused on forwarding packets at line-speed. Widely deployed routing schemes like OSPF and IS-IS include control plane mechanisms that utilize global message exchanges and computations to determine recovery paths for link failures. However, the slow reaction times of control plane mechanisms can adversely impact performance.

To address this, some solutions have been designed to allow the network to recover from failures only after the control plane has computed new paths and updated the state in all routers. This approach resolves the disparity in timescales between packet forwarding in the data plane (which can occur in less than a microsecond) and control plane convergence (which can take hundreds of milliseconds).

Recent advancements in centralized routing solutions, such as those based on SDNs, have improved recovery times. In SDNs, all routing computations are performed by a central controller, which then pushes the results to the affected routers, enabling faster recovery from failures. To match fast timing requirements, data-plane recovery have bene implemented. In a lot of cases, data-plane recovery is configured in a specific topology configuration such as t a ring, where the primary and backup paths are respectively mapped on a clockwise or opposite direction of the physical ring.

With respect to the allocation of backup capacity, two main options exist: dedicated or shared back up capacity. With dedicated capacity, a specific backup resource corresponds to one working path. In this case there is a one-to-one relationship between back up resources and working path. This establishes a one-to-one relationship between backup resources and the related working paths, with the backup resources being dedicated (1+1 resources) and unavailable for other traffic during normal operations. With shared backup capacity, however, the resources can be shared among multiple working paths.

Resilience methods in transport are well-established. In fact, there are standardized IETF solutions for transport resilience targeting different RAN use cases and slicing.

A.8 Performance impact of disadvantageous conditions in CNF-based 5G Core

In addition to the study, we've measured impact of increasing jitter introduced in addition to constant delay of 100ms towards the UDM network functions on the Procedure Completion Time.

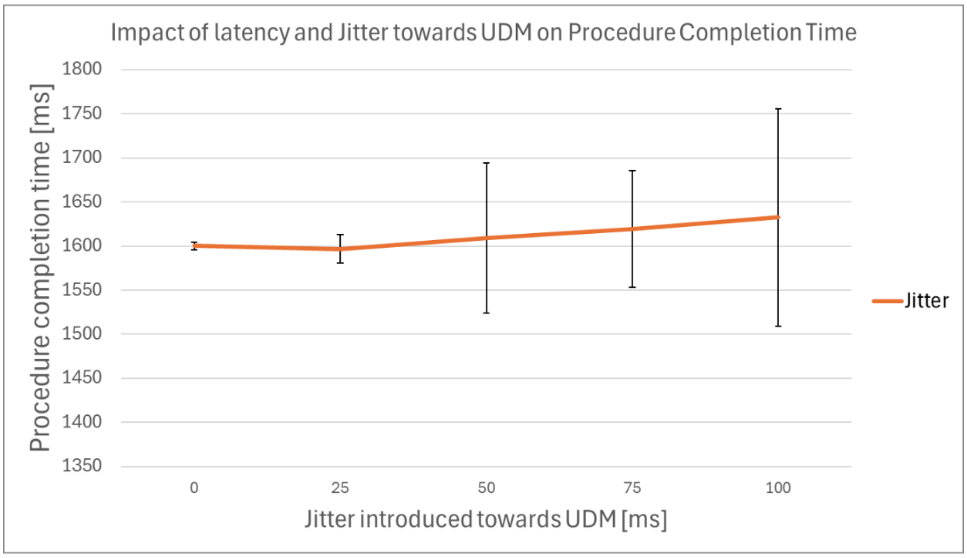


Figure A-26 Effect of increasing introduced jitter towards the UDM network function on registration procedure duration.

In Figure A-26 we can see that introducing additional jitter in addition to a constant delay, results in a wider confidence interval, but the average value for Procedure Completion Time remains similar. We notice an increase of 2.01% of PCT for 100ms of jitter introduced towards the UDM network function, and an increase of 0.55% for 50ms of jitter, which are insignificant given the confidence intervals.

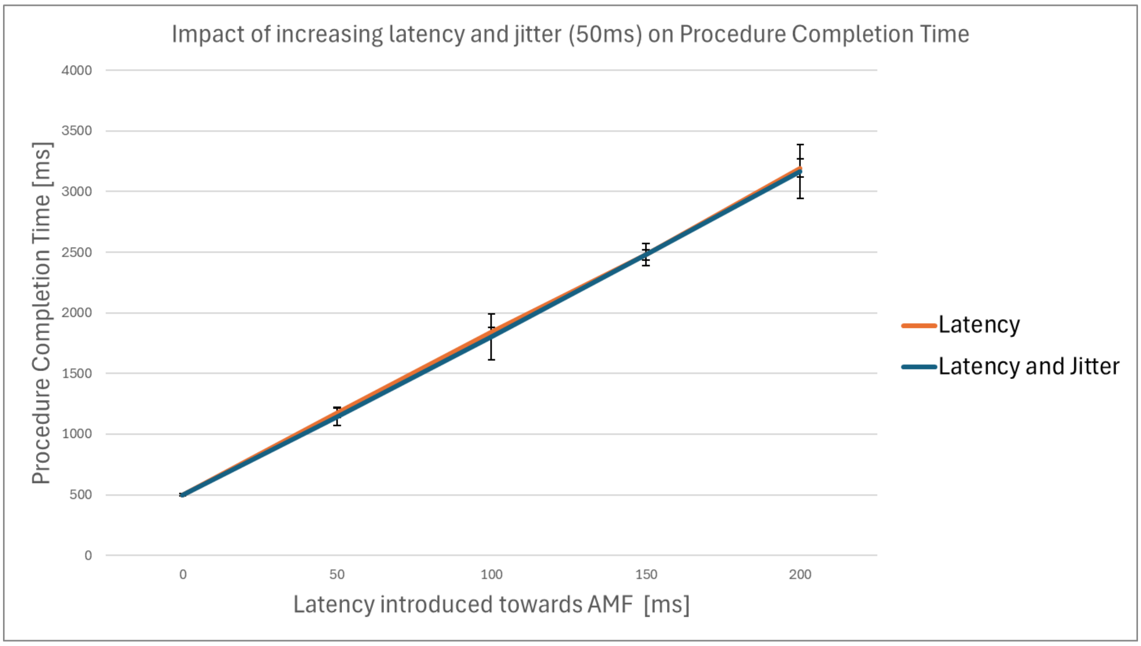


Figure A-27 Effect of introduced delay and jitter (50ms) towards the AMF network function on registration procedure duration.

A similar effect can be seen in Figure A-27, portraying the results of increasing the delay towards the AMF network function while keeping the jitter at 50ms. An increase in extreme values results in wider confidence intervals, while the average value remains within 2% of the original value.

A.9 Disaggregated RAN to enable cell-free massive MIMO

Following from what has been explained in section 5.2.3.3, the problem reduces to

The problem then transforms into a geometry problem, intending to find the minimum number of clusters needed to satisfy the distance requirement for all users, and in turn do it with the least fronthaul network extension. Given the cluster radius R and the radius of good performance r, with the assumptions made, the minimum number of extra clusters is [GAT24b]

$$n = \begin{cases} 0 & r = R \\ \left\lceil \dfrac{\pi}{\arcsin \frac{r}{R}} \right\rceil & \dfrac{R}{2} \le r < R \\ \left\lceil \dfrac{2\pi}{\arccos \frac{\frac{R}{r} - 1}{2}} \right\rceil & \dfrac{R}{3} < r \le \dfrac{R}{2} \\ \infty & r \le \dfrac{R}{3} \end{cases}$$

Given the minimum number of extra clusters, the smallest possible radius $s_{\min}$ of the circle formed by the n circles covering the annulus is

$$s_{\min} = R \cos\frac{\pi}{n} - \sqrt{r^2 - R^2 \sin^2\frac{\pi}{n}}.$$

A.10 Quantum Technology for 6G

A.11 Quantum Semantic

The model for quantum semantic communication is simulated in Qiskit 1.0.2, using an IBM quantum cloud simulator ibmq-qasm-simulator, for simulating quantum circuits subject to noise modelling.

In evaluating the effectiveness of our protocol, we underscore the trade-offs encountered between the rising complexity of semantic graphs and reliability as fidelity is assessed across an increasing number of qubits (corresponding to increasing size of KG). Our simulations are performed using the ibmq-qasm cloud-based simulator, which offers access to up to 32 qubits, on a Mac Studio system with 128 GB of memory. Our results demonstrate that under amplitude noise conditions (see Figure A-28), fidelity remains acceptable for up to 16 qubits, but a slight decline is observed for larger qubit counts due to the limitations of superconducting (quantum) technology. This decline is attributed to increased circuit complexity, leading to trade-offs between inherent properties of the superconducting qubits like temperature and gate lifetime ultimately reducing average fidelity [NSB+24].

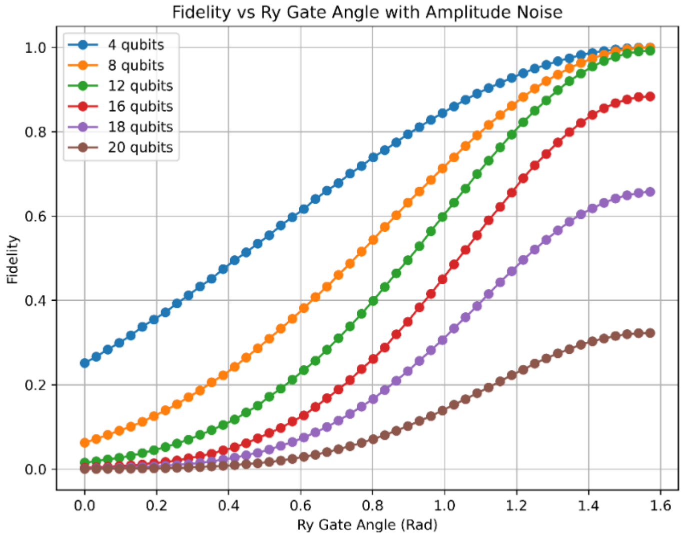


Figure A-28 The trade-offs observed between increasing qubits vs computational complexity (semantic encoding complexity) with amplitude noise.

A.11.1 Trustworthiness

An important element that must be considered in 6G communication networks is trustworthiness, which encompasses factors such as security, reliability, and safety, all of which ultimately enhance the network's overall resilience.

Trustworthiness in network information transfer encompasses several dimensions, one of which is privacy. Privacy, including its associated concepts like confidentiality, forms the foundation upon which trust is built in any communication system. Generally speaking, privacy can be defined as the selective management of information dissemination, and the assurance of varying levels of privacy within the network is essential for the viability of financial services, security services, and secure pharmaceutical data storage and retrieval, among others. Anonymization, and consequently privacy, represents a crucial parameter in a network system that has the potential to facilitate compliance with forthcoming 6G standards.

One of the foundational texts in the field of cybersecurity states that trustworthiness refers to the assurance that a system is deserving of trust [NRC99]; it is expected to function reliably even in the face of environmental disturbances, human errors, malicious attacks, and flaws in design and implementation. Trustworthy systems bolster the confidence that they will consistently exhibit expected behaviours and remain impervious to subversion.

A.11.2 Confidentiality: Destination Anonymity

In the realm of IoT integration within 6G and tactile internet networks, the application of quantum communication technology presents significant benefits in meeting various KPIs established by the industry. Additionally, it enhances the resilience of the solutions developed against noise, cyberattacks, energy leaks, and other vulnerabilities. As the technology in 6G and subsequent generations mandates that all essential services comply with at least ultra-reliable low-latency communication (URLLC) standards, many industries face challenges in fulfilling the latency and computational demands of their industrial and consumer technology solutions. This often results in substantial costs associated with suboptimal energy efficiency or inadequate security measures. Therefore, it is noteworthy that quantum technology has demonstrated potential in bolstering the resilience of existing classical solutions, ensuring compliance with evolving standards in terms of security and energy efficiency.

One notable instance is the cloud RAN based on 'quantum entanglement,' which demonstrated a reduction in time delay compared to classical C-RAN by approximately 0.3 milliseconds. This reduction is significant, particularly in the realm of tactile internet networks that demand high time sensitivity and low latency. Such networks, along with extensive mega-metropolitan networks, necessitate specialized and proficient IoT controllers to ensure classical resilience in traffic scheduling, queuing, and other functions. Additionally, these IoT controllers must maintain security while transmitting data to the associated IoT devices and haptic robotic systems, all while striving to achieve energy efficiency to meet sustainability requirements essential for successful implementation. To enhance resilience through improved security, recent advancements have employed Quantum Key Distribution (QKD) between the controllers and connected devices to potential malicious attacks on the wireless links established between them. Conversely, another initiative has focused on achieving energy savings for IoT devices by integrating QKD with software-defined networking for their fibre-optic connected controllers.

The potential for quantum technology to leverage a different form of quantum key distribution (QKD) is evident in the continuous variable QKD (CV-QKD). In contrast to the previously mentioned discrete variable QKD (DV-QKD), which necessitates the use of coherent sources and single photon sources that are challenging to integrate with current telecommunication systems, CV-QKD employs technologies such as homodyne and heterodyne detectors that are compatible with classical optical communication. This compatibility not only facilitates near-term deployment at reduced costs but also allows for the application of existing resilience studies relevant to classical technologies. However, it is important to highlight that in the context of free-space and atmospheric quantum communication, the use of continuous variable quantum entanglement with squeezed light states is generally discouraged due to significant channel losses that can compromise the integrity of the quantum information. Although initial findings regarding CV-QKD were met with scepticism from the telecommunications sector, recent evaluations of secret key rates (SKR) in multiple-input multiple-output (MIMO) CV-QKD systems under limited eavesdropping conditions—reflecting more realistic scenarios where an eavesdropper lacks complete control over the environment—have revealed substantial improvements in SKR compared to unrestricted conditions. Furthermore, it has been observed that increased squeezing can lead to a degradation of the SKR. These findings underscore the potential for non-squeezed coherent state-based CV-QKD protocols and represent a significant advancement in quantum technology

Confidentiality, regarded as a vital component of trustworthiness, has been defined as “The property that information is not made available or disclosed to unauthorized individuals, entities, or processes. [MRB+24]” This definition can be further refined to signify a commitment to a certain level of privacy within a specific context.

Anonymity, in simpler terms, refers to a situation where it is impossible to ascertain who performed a particular action, meaning the identity of the individual or entity involved remains unknown. More specifically, it involves a demonstrable non-zero degree of indistinguishability within what is termed an 'anonymous set.' In the realm of network communication, anonymity plays a crucial role in enhancing privacy—both of the information transmitted through a channel and the data at the network layer—thereby influencing the overall trustworthiness of the network.

A prevalent example where anonymity is essential in legitimate processes is in voting. The voting system operates on the principle that while the vote's destination is documented during the information transfer, the identity of the voter is not recorded alongside the vote itself. This establishes a framework where source anonymity is standard, with a clear understanding of the vote's ultimate destination, namely the party or entity for which the individual voted. In contrast, our discussion will concentrate on the opposite scenario—emphasizing destination anonymity while the source remains identifiable.

This approach is particularly beneficial in contexts where sub-networks interact within a self-organizing network or in scenarios involving a network of networks (NoN), as frequently outlined in the blueprints for 6G communication networks [MRB+24].

Specifically, consider a virtual network function (VNF) migration network scenario in Figure A-29, it is a network with a central node that when overloaded, wishes to offload VNF1 and VNF2 to two spatially separated physical nodes in the network, assuming each node's virtual machine only implements one VNF at a time (they have the capability to implement multiple types of VNF however). Importantly, the focus is on the case when the central node determines with no preference that there are two eligible nodes that can receive the to-be migrated VNF1 and VNF2. In the classical scenario, it is as simple as flipping a coin to decide which VNF is to be sent to which physical node.

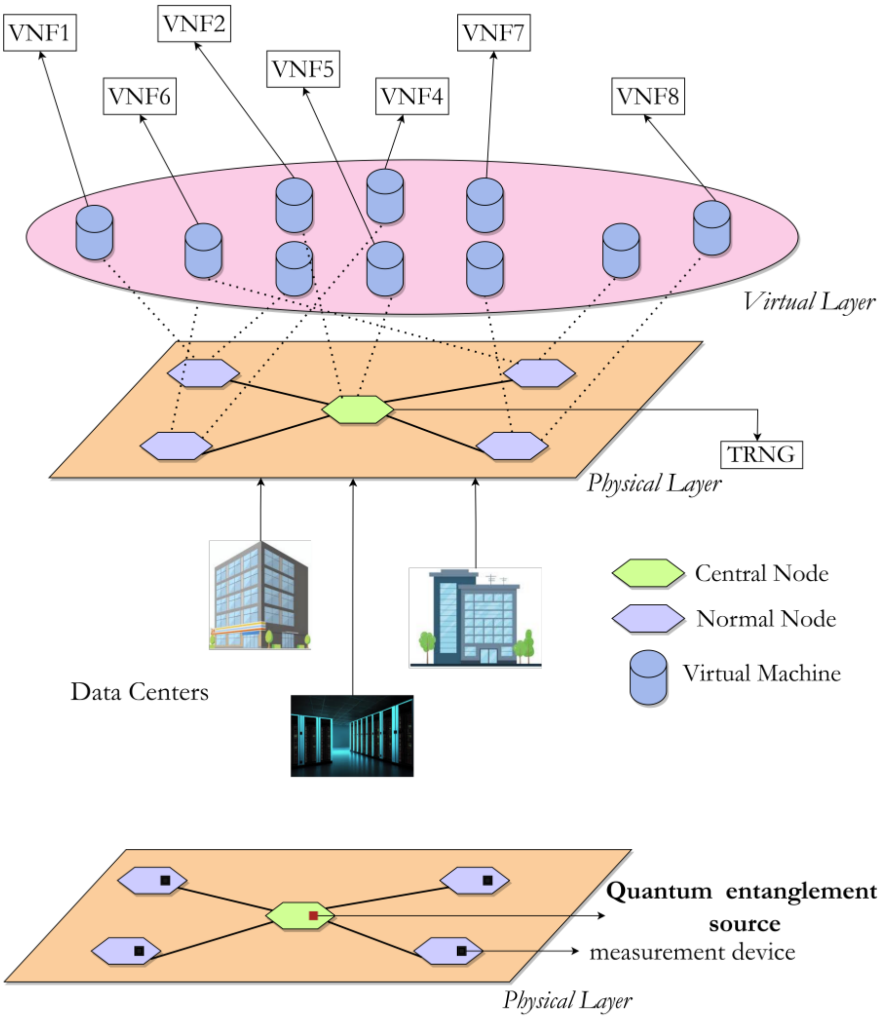


Figure A-29 Illustration of a virtual network function (VNF) migration network scenario [MRB+24].

However, in this process, if there were a malicious party or entity or spy at the central node who wished to orchestrate a VNF2-specific attack (using physical phenomena, on-site agents, etc.) by targeting the weakness of a particular physical node in implementing a particular VNF (WLOG, assume node 1 has a vulnerability in implementing VNF2 that is unknown to the network) with the caveat that they have enough resources to orchestrate only one such attack, then the malicious party has a possibility to gain access to the result of the coin flip to ensure their success. This means that with classical technology, in order to prevent exposure to internal network espionage attacks, encryption or steganography needs to be used thus wasting precious time in low-latency applications, or additional classical resources such as access to a third-party random number generator service is required, thus resulting in additional points of vulnerability in the network. This effect is especially prominent when the network scales in complexity [RBF24].

The quantum solution, see Figure A-30, involves including an entanglement generator at the central node, and detectors to perform measurements at the normal node. Assuming WLOG that a measurement of 1 meaning the node implements VNF2, and a measurement of 0 meaning the node implements VNF1, the nodes need not communicate to each other or the central node to ensure no redundant preparation for the same node - this is because entanglement ensures that when one entity measures the photon they receive from the Bell pair, the other will necessarily measure the other result without fail. This essentially means that no one has guaranteed information of which VNF will be implemented in which physical host until the measurement is made and then the physical node communicates back to the central node that it is ready to receive VNF2 (or VNF1). This is helpful in preventing potential malicious attacks that can result if there is an adversary hidden at the principal node who can compromise the security or functioning of the sub-networks nodes through physical/electronic/software means [RBF24].

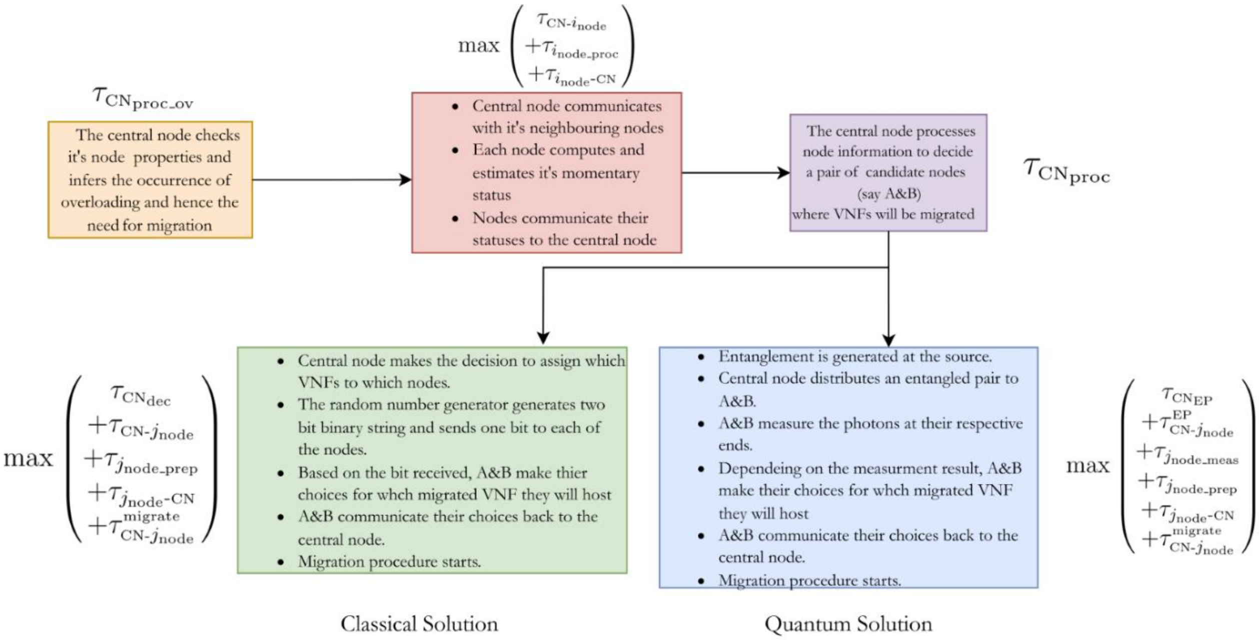

$\tau_{\mathrm{CN}_{\mathrm{proc\_ov}}}$
The central node checks it's node properties and infers the occurrence of overloading and hence the need for migration
$\max\begin{pmatrix}\tau_{\mathrm{CN}\text{-}i_{\mathrm{node}}}\\+\tau_{i_{\mathrm{node\_proc}}}\\+\tau_{i_{\mathrm{node}}\text{-CN}}\end{pmatrix}$
Central node communicates with it's neighbouring nodes
Each node computes and estimates it's momentary status
Nodes communicate their statuses to the central node
The central node processes node information to decide a pair of candidate nodes (say A&B) where VNFs will be migrated
$\tau_{\mathrm{CN}_{\mathrm{proc}}}$
$\max\begin{pmatrix}\tau_{\mathrm{CN}_{\mathrm{dec}}}\\+\tau_{\mathrm{CN}\text{-}j_{\mathrm{node}}}\\+\tau_{j_{\mathrm{node\_prep}}}\\+\tau_{j_{\mathrm{node}}\text{-CN}}\\+\tau^{\mathrm{migrate}}_{\mathrm{CN}\text{-}j_{\mathrm{node}}}\end{pmatrix}$
Central node makes the decision to assign which VNFs to which nodes.
The random number generator generates two bit binary string and sends one bit to each of the nodes.
Based on the bit received, A&B make thier choices for whch migrated VNF they will host
A&B communicate their choices back to the central node.
Migration procedure starts.
Entanglement is generated at the source.
Central node distributes an entangled pair to A&B.
A&B measure the photons at their respective ends.
Dependeing on the measurment result, A&B make their choices for whch migrated VNF they will host
A&B communicate their choices back to the central node.
Migration procedure starts.
$\max\begin{pmatrix}\tau_{\mathrm{CN}_{\mathrm{EP}}}\\+\tau^{\mathrm{EP}}_{\mathrm{CN}\text{-}j_{\mathrm{node}}}\\+\tau_{j_{\mathrm{node\_meas}}}\\+\tau_{j_{\mathrm{node\_prep}}}\\+\tau_{j_{\mathrm{node}}\text{-CN}}\\+\tau^{\mathrm{migrate}}_{\mathrm{CN}\text{-}j_{\mathrm{node}}}\end{pmatrix}$
Classical Solution
Quantum Solution

Figure A-30 The classical and quantum algorithmic steps for binary decision making with their associated latency expressions [MRB+24].

Note that one could also consider alternate network problems dealing with service function chain (SFC) management in a network of networks scenario, see Figure A-31 that is required to provide some network functionality F comprising 2 SFCs, under some specific constraints such as: the SFCs being implemented by a cyclic chain of 3 distinct VNFs that when rotated forward/backward by 1 position in one SFC and the opposite backward/forward rotation by 1 position in the other SFC, they are able to implement the same network functionality F; these VNFs are implemented in separate physical hosts with each triplet collection being managed by a sub-principal node that then communicates with a principal node of this network.

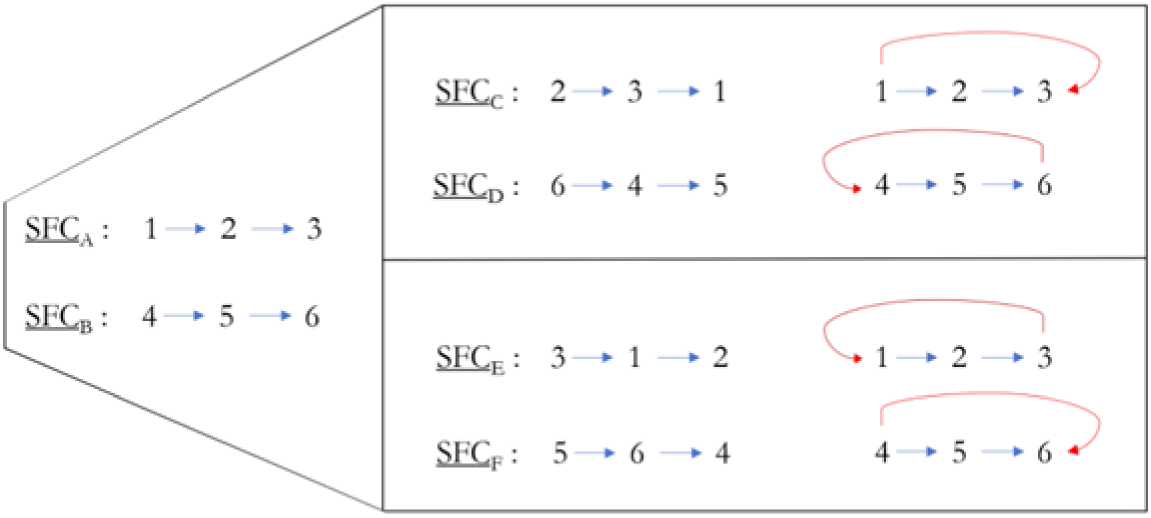


Figure A-31 VNF Chains and their corresponding SFC labelling.

We consider a network within a Network of Networks (NoN), which can be represented as a graph with its standard elements. At the core of this network is a principal node, serving as a central hub responsible for receiving and distributing functionality requests. These requests are communicated to the sub-principal nodes, organized in a star-like topology as illustrated in Figure A-32. These sub-principal nodes oversee the Service Function Chain (SFC) orchestration for their respective sub-networks. Each sub-network comprises physical host nodes, each dedicated to implementing specific Virtual Network Functions (VNFs), as depicted in the aforementioned figures.

As an aside, the quantum solution for the other problem concerning SFC management also utilizes entangled photons distributed by the principal node to essentially enable simultaneous SFC orchestration by the sub-principal nodes. If they measure a 1, they will rotate the VNF chain forward by one, and if they measure a 0, they will rotate it the opposite way. The crucial takeaway here is that only on measurement of the incident photon at the sub-principal node will a particular SFC be decided to be implemented; this means that there is actually some finite-time anonymity of the decision at the principal node provided by the quantum technology, since the principal node only knows which SFCs are implemented and does not know which VNFs are in which physical hosts until the sub-principal nodes communicate back to it.

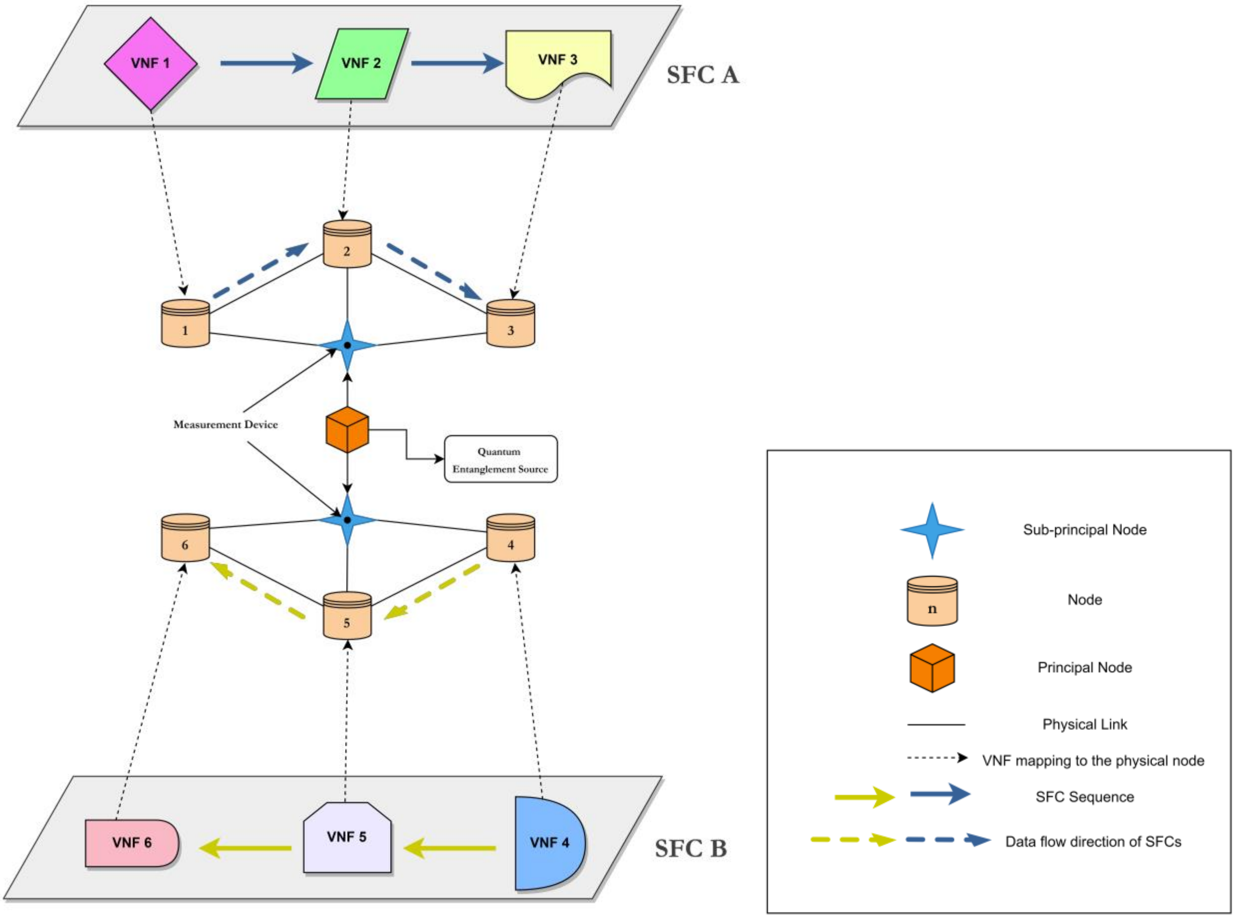


Figure A-32 Network model with classical technology, depicting physical and virtual layers.

A.11.3 Quantum Synchronization

Cross-Correlation Function in TCEP Synchronization

The cross-correlation function is central to TCEP-based synchronization. It provides a quantitative framework to measure and analyze the temporal correlations between photon detection events at spatially separated nodes, Alice and Bob. It enables precise determination of timing offsets by evaluating how the arrival times of entangled photon pairs align under varying time lags τ.

The cross-correlation function, denoted by $C_{AB}(\tau)$ is a mathematical tool used to quantify the degree of temporal correlation between photon detection events at two nodes, Alice and Bob, as a function of a time lag τ. It measures how well the photon arrival times at Alice and Bob align when τ shifts the detection timestamps. The function is defined as:

$$\hat{C}_{AB}(\tau) = \frac{n(\{i|t_{A,i} = t_{B,j} - \tau\})}{n(\{i|t_{A,i}\}) \cdot n(\{j|t_{B,j}\})},$$

Equation 10.8-1

Where each term serves a specific role in the computation:
n(i|t{A,i}=t{B,j}−τ): The number of coincident photons detected at Alice and Bob when a time lag τ shifts their arrival times. This term captures the temporal alignment of the photon pairs.

- n({i|t{A,i}≤T+τ}): Total number of photons detected at Alice during the specified time interval.
- n({j|t{B,j}≥τ}): Total number of photons detected at Bob during the specified interval.
- T : Total duration of the measurement.
- Δt : Time resolution of the detection system.

The value of $C_{AB}(\tau)$ peaks at the lag τ, where the temporal alignment of photon arrivals is strongest, indicating the propagation delay or clock offset between Alice and Bob. The height of the peak reflects the strength of the correlation, while the width of the peak, often quantified as the Full Width at Half Maximum (FWHM), represents the precision of the synchronization process.

In practice, $C_{AB}(\tau)$ is evaluated using datasets collected from TCEP experiments, where photon timestamps are recorded using Time-to-Digital Converters (TDCs) with high temporal resolution. A sharp peak in the function confirms strong temporal correlations, while the normalization ensures the result is independent of the photon detection rates at Alice and Bob. This analysis is critical for extracting precise timing offsets and is often implemented in real-time using FPGA platforms to handle large datasets efficiently. By combining the quantum properties of TCEP with the analytical power of $C_{AB}(\tau)$ this method provides a robust and scalable framework for achieving sub-nanosecond synchronization in quantum networks.

The cross-correlation function quantifies the strength of temporal correlations between detection events at Alice and Bob. The function typically forms a peak at the synchronization offset, corresponding to the propagation delay Θ. The following aspects of the peak provide critical information:

- **Peak Position**: Indicates the clock offset between Alice and Bob.
- **Peak Height**: Reflects the strength of temporal correlations, with higher peaks indicating better synchronization.
- **Peak Width (FWHM)**: A narrower Full Width at Half Maximum (FWHM) indicates higher precision, with sub-nanosecond resolution achievable in optimized systems.

In TCEP Synchronization, the cross-correlation function is used for:

- **Precision Timing**: By aligning clocks based on the peak position, TCEP synchronization achieves sub-nanosecond timing accuracy, essential for quantum networks.
- **Noise Resilience**: The function inherently accounts for noise and timing jitter, making it robust against environmental fluctuations.
- **Scalability**: Normalization terms ensure compatibility with varying photon detection rates, enabling applicability in large-scale distributed systems.

A.11.4 FPGA-Based Implementation of the TCEP

Field Programmable Gate Arrays (FPGAs) are reconfigurable hardware platforms optimized for parallel, high-speed computation, capable of real-time processing in quantum synchronization tasks. In our implementation, the FPGA was used to compute cross-correlation functions on an already generated dataset of photon timestamps from TCEP experiments as depicted in Figure A-33. This approach demonstrated the FPGA's ability to handle high data rates efficiently, achieving sub-nanosecond precision while maintaining scalability and robustness. The design ensures adaptability to varying network conditions and detector resolutions, establishing FPGAs as a critical tool for precise synchronization in quantum networks. The FPGA utilized in this work is the Stratix 10 on the DE10-Pro board. This platform supports Intel's OpenCL Board Support Package (BSP), which provides efficient facilities for the real-time processing of photon timestamp data in TCEP-based synchronization systems.

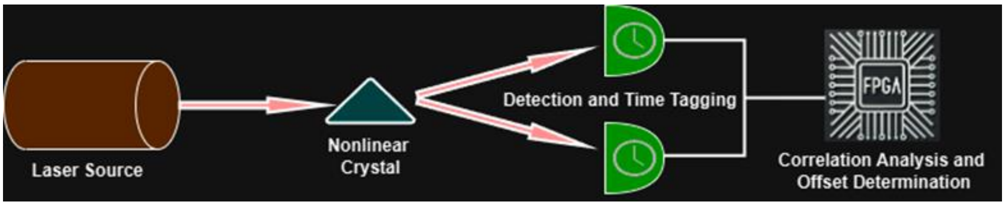

Laser Source
Nonlinear Crystal
Detection and Time Tagging
FPGA
Correlation Analysis and Offset Determination

Figure A-33 Diagram illustrating the process of photon generation and detection using a laser source and a nonlinear crystal.

Figure A-34 shows a bar graph comparing execution times for CPU (blue) and FPGA (red) during correlation calculations based on varying numbers of events per bin (1000, 5000, and 10,000). The data illustrates that while CPU execution times increase significantly with more events, FPGA execution times remain consistently low, highlighting the FPGA's superior efficiency for high-throughput computational tasks.

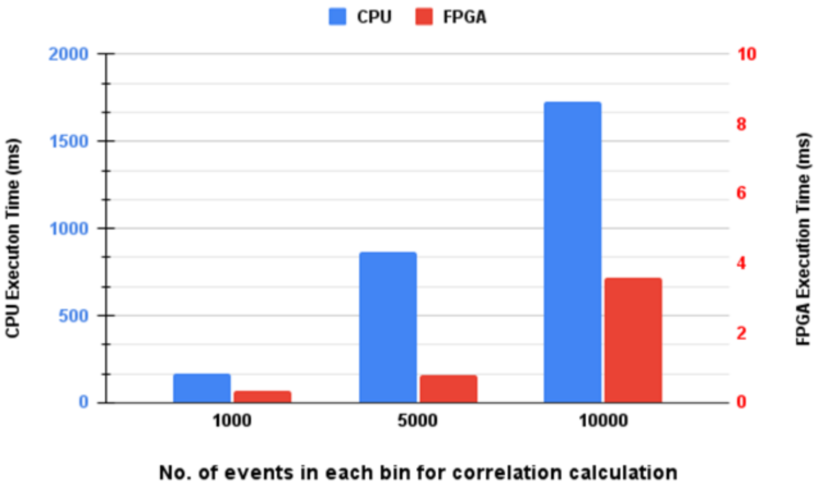


Figure A-34 Comparison of execution times for CPU (blue) and FPGA (red) during correlation calculations.

Synchronization Accuracy: The TCEP-based system demonstrated exceptional timing precision:

- Timing Jitter: Consistently below 100 ps, outperforming classical methods like Precision Time Protocol (PTP).
- Correlation Peaks: Experimental results confirmed that the Full Width at Half Maximum (FWHM) for correlation peaks is below 100 ps, validating sub-nanosecond synchronization accuracy.

Scalability and Throughput:

- High Data Volume Handling: Real-time processing of large datasets with minimal latency.
- Efficient Resource Utilization: FPGA implementation consumed 31,557 ALMs and 67,688 registers, showcasing adaptability to larger quantum networks.

Further current research is going on to investigate the integration of quantum synchronization into 5G networks and evaluates its impact on the Time Error Budget (TEB). Classical synchronization protocols such as Precision Time Protocol (PTP) and White Rabbit (WR) achieve nanosecond-level accuracy with a TEB of approximately 1.5 µs, but they face limitations from clock drift, propagation delay, and noise. Quantum synchronization, leveraging femtosecond (fs) and picosecond (ps) precision through methods like confined atoms and correlated photons, demonstrated a reduction in TEB to 1.1 µs by mitigating holdover and noise components. Femtosecond accuracy exhibited a time deviation uncertainty of 10^{-15}, while picosecond-level systems achieved, enabling enhanced stability and synchronization precision. Maximum Time Interval Error (MTIE) and Time Deviation (TDEV) analysis revealed that noise averaging effects contributed to femtosecond stability, while picosecond systems faced higher variability in multi-node networks. Although the findings highlight quantum synchronization's transformative potential for reducing latency and enabling ultra-reliable low-latency communications (URLLC), challenges remain, including noise interference, clock drift, asymmetric propagation delays, and implementation costs. A hybrid approach integrating quantum synchronization at key points, such as the Primary Reference Time Clock (PRTC), while maintaining existing infrastructure, offers a practical pathway for future networks. This study underscores quantum synchronization's significance in 6G networks, where achieving sub-nanosecond accuracy will be crucial for applications such as extended reality (XR), distributed computing, and secure communications, while also emphasizing the need for advancements in noise filtering, oscillator stability, and cost-effective deployment strategies.

Annex B Quantified targets

The Hexa-X-II project objectives are defined by Hexa-X-II from the beginning of the project, and the work packages were assigned to investigated certain objectives that fit the work package description. For each objective, there are one or more so called Quantified Targets (QTs), which the work packages should try to fulfil. WP3 were assigned objective 3,4 and 5. The QTs for these objectives are shown in Table B-1.

Table B-1 The Hexa-X-II project objectives assigned to

Quantified target	Objective
	Objective 3: Enhanced connectivity for 6G services: Develop and describe radio access solutions for communication considering the requirements on 6G services
QT3.1	Low end-to-end (E2E) latency (<1 ms and <0.1 ms in critical sub-networks).
QT3.2	Provide to >99% global population & world area with at least one basic 6G use case (when/where needed) at sustainable cost levels.
	Objective 4: Network sensing, compute, and AI for novel digital services Develop and describe solutions for an expanded scope of wireless networks, for creation and processing of data, considering the requirements on 6G services
QT4.1	<1 m at 90th percentile Radio/Communication based sensing precision in mid-band and <10 cm at 90th percentile Radio/Communication based sensing precision at 100 GHz to detect a moving human-sized object at 10 m distance
QT4.2	>20% improvement in performance in at least one of energy efficiency, latency, bit rate or area capacity through use of sensing, localisation, traffic data, or mobility patterns for AI-based optimisation in selected use cases
	Objective 5: Efficient network realisation, implementation, and management: Develop and describe solutions for building the 6G platform considering the requirements of 6G services
QT5.1	Reducing energy consumption per bit in networks by >90%.
QT5.2	>25% Operational Expenditure (OPEX) reduction by using zero-touch automation.

B.1 Objective 3: Low Latency Support (QT3.1)

Objective 3 includes the target QT3.1 (see Table A-5) to Network supporting low latency (<1 ms E2E and <0.1 ms in critical subnetworks). In mobile network, we distinguish between control plane and user plane latency. While user plane latency is critical for the user experience when running applications over mobile network, the control plane latency is also important because it impacts the time to setup up the user plane connection. This section addresses objective 3 on supporting <1 ms latency.

Control Plane Latency

The control plane latency in the 6G system is defined as the time needed to complete control plane procedures such as UE registration, PDU Session establishment, etc. The procedure-based functional decomposition of Core Network functions described in section 5.2.1.3 proposes a design where each system procedure is consolidated into a single NF. The evaluation results of this study showed that the PCT of UE Registration and PDU Session Establishment procedures is reduced by 42% and 50% respectively compared to 5G core design.

Further, the 5G SBA architecture can be improved by using DataFlow programming to tackle data-centric Service-Based Architecture (SBA) and the higher decomposition of core Network Functions (NFs) as described in section 5.2.2.4. Dataflow-based service composability is proposed as a solution to integrate these approaches seamlessly, fostering innovation and customization for emerging applications [BQG+24]. The results emphasize the performance benefits of redesigning the network modules interactions based on DataFlow programming, especially for time-sensitive 5G workflows. The workflow completion time, depending on the protocol of choice, may be reduced by up to an order of magnitude.

Moreover, as the networks are comprised of multiple dynamic and ephemeral elements, reliability becomes harder to guarantee. Failure prediction module of 5GCOP – 5G Core Observability Platform for modular networks, presented in section 3.1.1.4, offers identifying bottlenecks (latency between pair of network functions) and guaranteeing reliability for 5G/6G Core performance evaluation of parameters such as procedure completion time.

User Plane Latency

The user plane latency is estimated following the methodology presented in [HEX23-D53]. Figure B-1 shows the E2E user plane path with the individual components that contribute to the total delay.

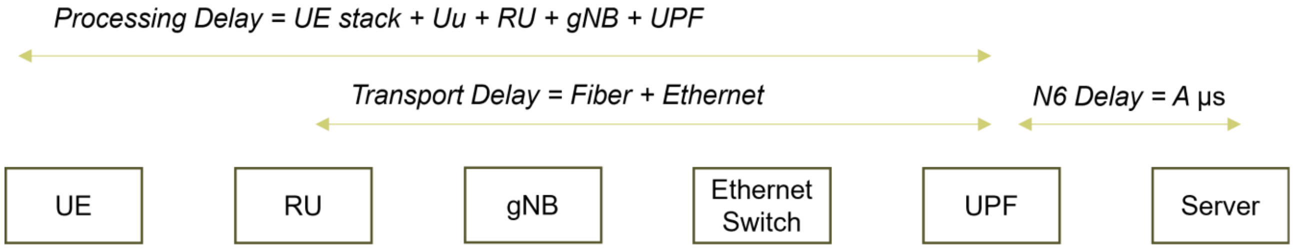

Processing Delay = UE stack + Uu + RU + gNB + UPF
Transport Delay = Fiber + Ethernet
N6 Delay = A µs
UE
RU
gNB
Ethernet Switch
UPF
Server

Figure B-1: Processing and transport delay components in E2E UP path.

Table B-2 summarises the processing delay of the different layers in the UE protocol stack, the air interface delay (Uu) (see HEX23-D23] for more details), the processing delay in RU, the processing delay of gNB layers, and the UPF processing delay. The transport delay when using Fiber over 50km distance to connect the RAN and Core domains as well as the ethernet switching delays is calculated to be equal to 196 µs. Accordingly, the total E2E delay is equal to 380 + A µs calculated as the summation of the processing delay of mobile network elements shown in Table B-2, the N6 delay, and the total transport delay, i.e., 196 µs. For more details on how these values were derived, please refer to [HEX23-D53].

Table B-2: Mobile Network elements processing delays.

UE stack					Uu	RU	gNB				UPF	Transport
PHY	MAC	RLC	PDCP	SDAP	Uu	PHY	MAC	RLC	PDCP	SDAP	N/A	N/A
8 µs	8 µs	8 µs	8 µs	8 µs	100 µs	8 µs	8 µs	8 µs	8 µs	8 µs	4 µs	196 µs

The study on orchestrating the extreme edge, presented in section 5.4.1.4 enables the use of end-user terminals in the vicinity of the UE, to host ETSI MEC applications. This certainly reduces the latency required to transverse the network towards the application server, i.e., the N6 delay, therefore reducing the end-to-end latency.

B.2 Objective 3: Enabling global coverage (QT3.2)

Service coverage [HEX223-D12][HEX223-D13] is one of the main KPIs for 6G, playing a key role in all identified 6G use cases. This section argues how the aforementioned enablers contribute to achieving the QT3.2 (see Table A-5).

The ubiquitous network use case aims to ensure basic services with the same level of experience everywhere. To achieve this, connectivity shall be provided not only to urban areas but also to remote areas, relying on both TN and NTN. The target is for (>99%) of the global population and (>99%) of the world area to have essential service coverage, when and where needed, at sustainable cost levels. The above two goals (one based on earth *population*, and one based on earth *area*) spring from a desire to include and enable more and new 6G markets. Of the above goals, coverage of 99% of the earth's surface is by far the most challenging, since the world' population exhibits a strong spatial imbalance: in 2009, the European Commission’s Joint Research Center published a map in the World Bank’s World Development report, showing that 95% of the world’s population is concentrated in just 10% of the land surface [WDR09]. However, the same report also shows that only 10% of land on Earth was considered ‘remote’ – more than 48 hours travel time from a large city [UN10] (in other words, 90% of the earth's land surface is not ‘remote’). This suggests that it is most meaningful to focus on the *areal* coverage goal above.

It is widely agreed that fulfilment of this goal requires presence of (a) global NTN(s). Thus, we assume the presence of an NTN that provides very basic connectivity (possibly with low data rates, with high latencies, and only outdoors) on at least >99% of the earth's surface. The focus of a quantitative target study then turns into a *QoS-coverage* target, "provision of connectivity with a certain quality of service in a certain location".

An additional assumption is that even in the presence of NTNs, there will be a *residual inequality* between connectivity based on TNs and that based on NTNs. For instance, NTNs will not be able to provide the TN-quality in terms of latencies. Recognizing that *areal service inequalities* are the main motivation of the above coverage target, a relevant performance measure captures these inequalities. To this end, a novel *coverage inequality index* can quantify urban-rural coverage fairness (or rather, unfairness) and identify which remote area networks should be adapted to improve essential service coverage. The developed coverage index takes two maps of a larger region (e.g. country, or region of validity of a spectrum license) as its input, a rurality map, and a TN coverage map. The rurality map quantifies for each location on the map its remoteness. Rurality measures have been proposed in the literature, and here the weighted distance measure based on the ARIA index [CDHAC01] is adopted. The coverage map is a binary map of the type regularly published by most national operators, indicating for any location on the map, whether the location has connectivity provided by either a terrestrial network (with a certain minimum data rate) or, alternatively, by a non-terrestrial network.

The inequality index is then computed with mathematical tools from economics: notably the Lorentz-concentration curve is created and the area enclosed by this curve and the line-of-equality provides the inequality index. The index reflects the extent to which non-terrestrial coverage (of lower quality) is concentrated to rural regions of the map. A number close to 0 indicates that connectivity is distributed equally over the map (urban areas are not privileged with more high-quality terrestrial connectivity than rural areas), while a larger index value (closer to 1) indicates a larger inequality and terrestrial coverage being structurally skewed to the urban regions. Figure B-2 shows an example based on the coverage map for Sweden. The yellow locations enjoy terrestrial connectivity of at least 10 Mbps, while the dark locations are covered by non-terrestrial networks. The right-hand figure illustrates how the inequality (y-axis) develops over the years, when terrestrial network coverage is expanded (example terrestrial data from 2013-2020 used for illustration).

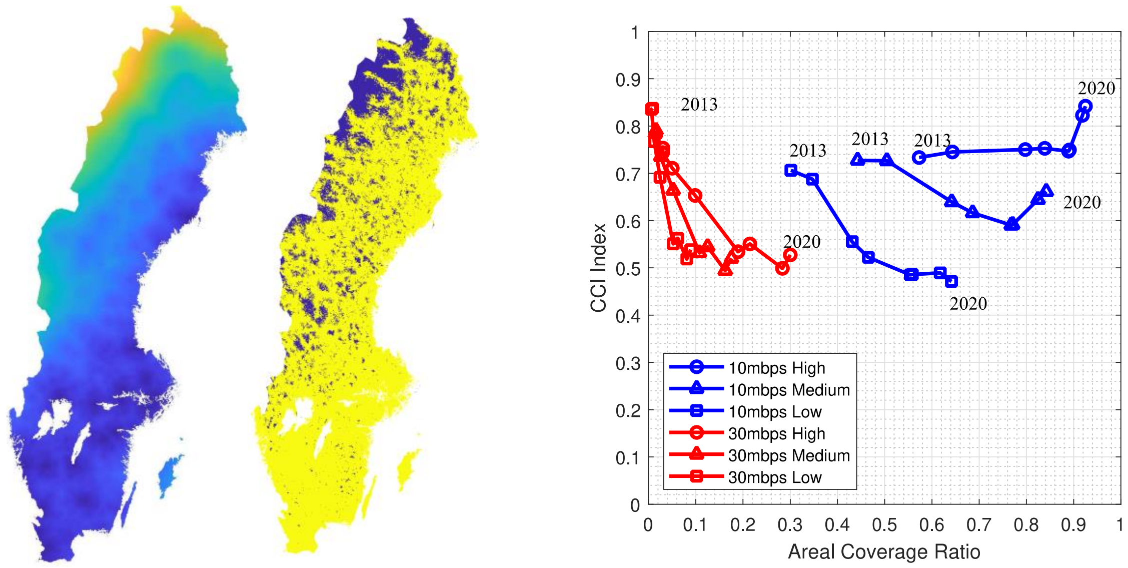


Figure B-2: Example inequality index for Sweden. Left: the two input maps (rurality map and TN coverage map locations covered with >10Mbps). Right: The cellular coverage inequality (CCI) index (y-axis) versus the areal

coverage fraction of the TN. The coverage maps from years 2013-2020 are used to illustrate the index comparison over time.

Even though NTN in remote areas will be essential for providing the required basic coverage, it will have the trade-off of additional delay. One of the reasons is that ISL hops may be needed to connect to a ground station [HEX23-D53]. To evaluate the target of 99% coverage for 99% of the population means that there is a need to perform an evaluation of a very rural area, not usually covered by terrestrial network due to the limited population density. [HEX23-D53] had a similar target and evaluated the performance of a LEO satellite constellation for very few users over Atlantic Ocean for a given number of satellites in orbit. The results showed that at least 600 satellites are required to support very few users (maximum one UE per beam) and a user throughput of at least 1 Mbps. However, these simulations did not consider any interference between the beams (which occur due to the beam overlap). The simulations in [HEX23-D53] also showed that the delay due to ISL hops to the ground station was in the order of additional 40-50 ms latency.

These simulations model the beam overlap and interference in more detail than [HEX23-D53]. The simulations are performed for one LEO satellite for a number of beams, where each beam is one cell. The simulations are performed with both frequency reuse factor (FRF) 1 and 3 used, the total bandwidth used is 30 MHz (for both FRF=1 and FRF=3), see Figure B-3. The carrier frequency is 2 GHz and channel Model is TR 38.811.6.6 with LOS probability = 1. The users are placed randomly over the area, using a full buffer traffic model. Note that the simulations are divided into two regions: one central region where the statistic is collected and one interference region. The central region contains 19 beams, and each beam is treated as a cell here. Each cell (beam) covers an area of 1400 km^2, i.e., each cell is covering a relatively large area of a size of a very large city.

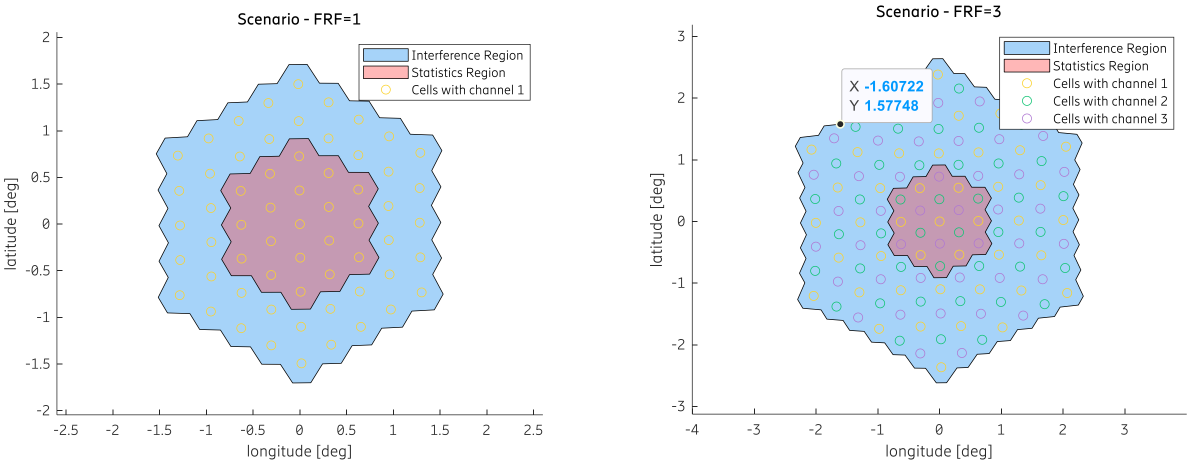


Figure B-3: The simulations use two regions: one central region where the statistic is collected and one interference region. The left figure shows frequency reuse factor (FRF) of 1 and the right for FRF=3.

The results are performed for different number of users per beam. Figure B-4 shows the normalized (per Hertz) user throughput for 1 and 10 users. The FRF = 3 case performs best, with slightly larger user throughput. The DL normalized user throughput for FRF=3 is around 0.5 bps/Hz when the number of users per beam is one and around 0.05 for 10 users per beam. This corresponds to around 15 Mbps for FRF=3 when the bandwidth is 30 MHz. The UL throughput is noticeably lower compared to the DL due to the lower transmit power of the UEs. For one user per beam and FRF =3, the normalized user throughput is in the order of 0.01-0.02 bps/Hz, which corresponds to a user throughput of around 0.2 and 0.4 Mbps.

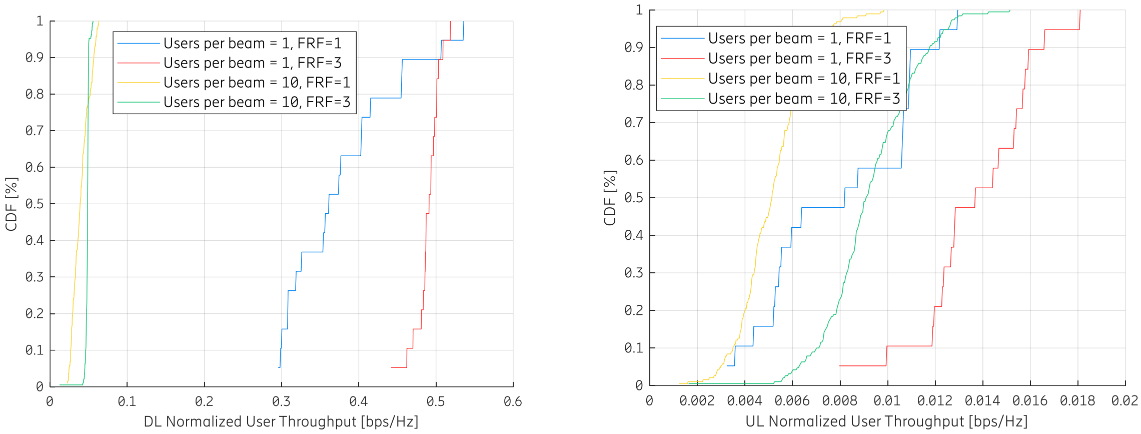


Figure B-4: Normalised user throughput for frequency reuse factor of 1 and 3 for UL and DL.

The target here stipulates that 99% coverage for 99% of the populations for a basic 6G service. Figure B-4 shows that it possible to support a 6G service of at least 1 Mbps in DL and 0.2 Mbps in UL. In the context of NTN, it should be noted that TN-NTN dual connectivity may also enable finding a better balance between building a TN network and relying on NTN, to minimize the footprint. At the same time, it would ensure that the connection is available and there is service continuity even in areas where there are coverage holes from the TN or where only one of the two networks provides coverage.

The focus so far has been in remote areas. However, even in urban environments there are spots where direct line-of-sight is blocked, for instance by buildings. The geometry of such a scenario is shown in Figure B-5. A deployment where 50 simultaneous users are uniformly distributed on the streets of the crossing is considered. All channels are modelled as Ricean fading (Ricean factor 5), and a proportional fair scheduler accommodates the scheduling of resources.

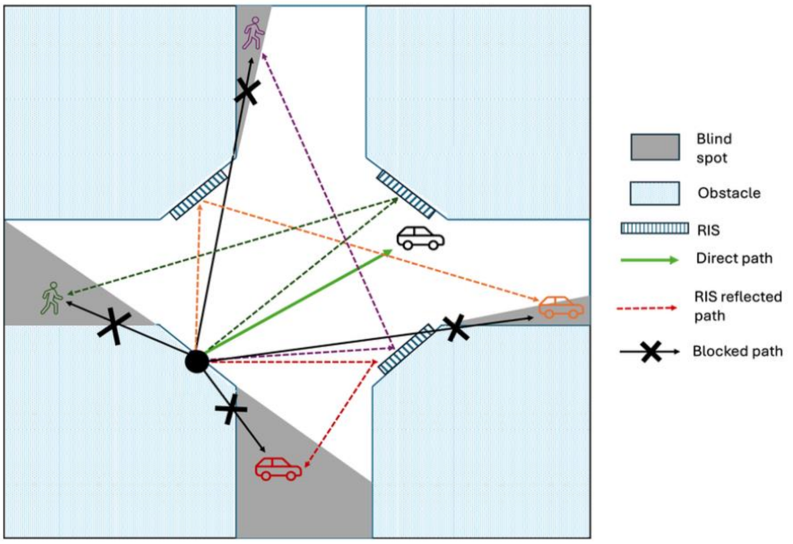


Figure B-5: An urban scenario. The road-crossing with one base station is also equipped with three strategically located RIS-boards improving SNR in locations without line-of-sight to the base station.

The three RISs are configured according to three protocols/scenarios. Scenario 1 ('Autonomous RIS') is fully autonomous (the RISs are not controlled by external network or nodes) and provides fully opportunistic beamforming by randomly assigning narrow beams. The second and third scenario require the dynamic configuration by the network. Scenario 2 ('controlled RIS') actively schedules, jointly with the user scheduling, a beam from a codebook of beams. Scenario 3 ('coherent RIS'), provides an upper bound of the performance in that the beamforming assumes knowledge of the UE position and provides optimal beamforming gain each time a user is scheduled. We compare the results against a 'no-RIS' scenario.

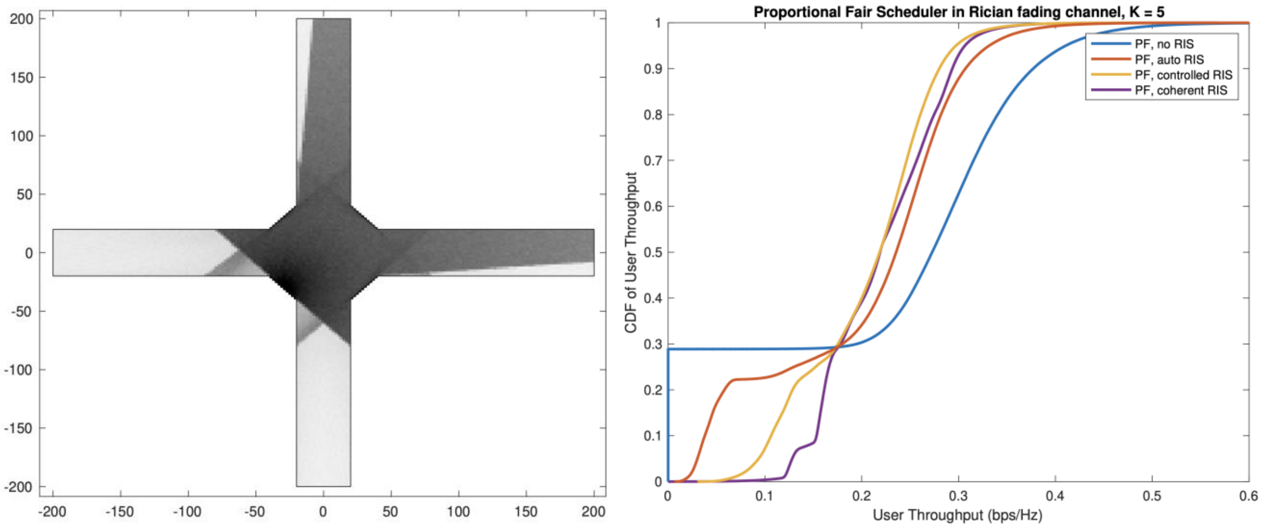


Figure B-6: Simulations results of the urban scenario. Left: the average SNR distribution in the autonomous RIS scenario. Right: the cumulative distributions of the user throughput for the 3 RIS-employed scenarios, along with the "no-RIS scenario.

Figure B-6 illustrates the results. The left-hand figure illustrates for three autonomously operating RISs how the distribution of the average SNRs now also provides non-zero SNR in regions that are blocked from a line-of-sight to the base station. The right-hand figure shows the extent to which the three RIS protocols improved the data rates of the 'cell-edge' users (29% of the users that do not have coverage at all when no RISs are deployed). The purple curve provides an upper bound in that the RIS configurations are optimally directed to each scheduled user. Not however, that also the autonomous RIS scenario now provides coverage and connectivity to users that do not have line-of-sight to the base station. Clearly this comes with a scheduling sacrifice by the high-SNR users: a more evenly distributed data-rate distribution is the result.

As far as the cooperating mobile robots use case and in general the network-assisted mobility use cases are concerned, a 6G network should be capable to provide coverage in critical scenarios via drone operation, traffic management, collision avoidance. Flexible topologies are required for providing 3D coverage in these critical scenarios where, for example, the TN does not operate anymore. The approach leverages trustworthy flexible network topologies enhanced by autonomous drones to ensure robust 6G coverage, particularly in remote or underserved regions, by dynamically deploying UAVs as relay nodes. This strategy addresses the ambitious target of providing over 99% of the global population and 99% of the world area with at least one basic 6G use case whenever and wherever needed, all at sustainable cost levels. Central to this solution is the optimization problem, formulated with a constraint of 100% coverage, ensuring network access across all target areas. Key features of this approach include adaptive deployment through AI/ML-driven algorithms that optimize drone placement for maximum coverage with minimal resources, continuous trust evaluations to maintain network reliability and security, and cost efficiency by balancing the number of drones against coverage needs to sustain operational costs. For instance, in the context of use cases such as warehouse inventory audits supported by cooperating AMRs and autonomous systems, the flexible topology ensures seamless network connectivity for all the nodes even in remote spots where the signal is compromised. This way, critical application functions like path planning and collision avoidance maintain their operation without any interruption increasing this way the reliability of the system. Moreover, in network-assisted mobility use cases, a 6G network must provide coverage in critical scenarios through drone operations, facilitating essential services such as traffic management. Flexible topologies are essential for delivering 3D coverage in these scenarios, especially when the TN is non-operational. Simulation results, illustrated in Figure B-7, validate the effectiveness of the framework, demonstrating that the proposed flexible topology achieves 100% inclusion with an optimal number of drones, significantly reducing deployment and maintenance costs compared to static infrastructures. By integrating trustworthy flexible topologies with optimized drone deployment, this contribution not only meets the current demands of diverse 6G applications but also paves the way for a resilient and scalable network infrastructure capable of supporting critical operations in dynamic and challenging environments.

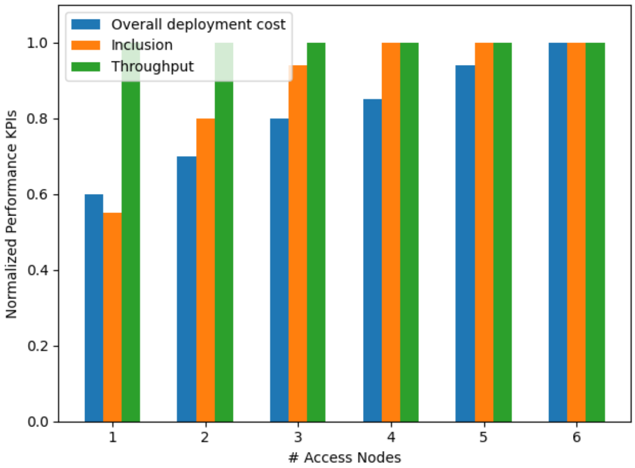


Figure B-7: Optimized drone placement to ensure 100% inclusion, with performance KPIs normalized across deployment cost, inclusion, and throughput.

Shifting the focus to the immersive reality use case, it is critical for the overall user experience that service continuity is guaranteed. Similarly, human centric services [HEX223-D12], such as public safety services and precision healthcare, require the use of trusted environments and highly depend on the communication to be responsive, stable and reliable. Subnetworks and the newly defined UE role of MgtN would enable this. The MgtN, which is a user-owned node, would be able to extend the coverage and provide access to lower-cellular-capability devices, such as XR devices, to the overlay network. In addition, since the MgtN would be a user-owned node it would ensure the user's privacy. The MgtN's main role would be to support the UEs in the subnetwork with certain procedures, either related to cellular communication with the overlay network or to their applications, and to enhance their access towards the overlay network, which would increase the coverage for UEs in a subnetwork in an energy-efficient manner as well as provide better service to the user. The concept of subnetworks and MgtN could also be used for extending outdoor coverage or improve the user experience in areas with poor service coverage, where the battery of the mobile devices could drain faster due to cell search and measurements. One of the trusted devices could assume the role of a MgtN and provide coverage to the rest of the trusted devices, removing their need for frequency scans and measurements. Finally, the coverage of 3GPP Network could be extended to non-fully compliant 3GPP devices, e.g. Non-StandAlone (NSA) devices that have reduced cellular capabilities and require an anchor device to establish a cellular connection [6GS23-D42]. Access to the cellular network for those devices would be provided in a controlled way, via the subnetwork.

A complementary enabler for guaranteeing service continuity and responsive, stable and reliable communication is the WLAN and cellular aggregation. That enabler could provide enhanced coverage in certain scenarios, such as in-home connectivity. A user-owned WT could be connected to both the cellular BS, as well as to the UE, ensuring the required service even when the UE is not in cellular coverage (e.g., the user is in the basement, where there is only WLAN coverage).

Multi-domain/Multi-cloud federation is the capability to aggregate cloud services provided by multiple domains and providers into a single, coherent cloud. This is enabled by aggregation of cloud resources into a cloud continuum, made available by cloud continuum nodes. The federation concept is not only about the capability of spanning across the administrative domains, but it is also strictly related to the multi-cloud capabilities, i.e. the aggregation of underlying cloud resources built on different cloud technologies, including private and public cloud. In the context of service coverage to assure availability of at least one 6G use case on the 99% of the world area, the critical matter is related to distribution of cloud continuum nodes over the area with the objective to enable placement of the service in the distance from the user, that assures the same level of experience in the whole area. Currently physical resources forming cloud resources are concentrated in the areas with high population density, what differentiates the quality of experience for users located in the areas of resource concentration in contrary to those located outside.

Fulfilling of objectives of 6G network requires appropriate planning of resource distribution. Resources are located in data centres, which can serve the area surrounding the data centre location. Aggregating more data centres into logical aggregated data centre, under the condition that type of resources and network connectivity assure, that all cloud services based of the resources provided by aggregated data centre can be offered with the same quality of experience, allows to expand the area of the service availability. [HEX223-D33] provided detailed analysis of the distribution of the data centres over the area of country (in this case Poland), with the assumption that the aggregated data centre (i.e. cloud continuum node or simply network node) will serve the area of diameter of 50 km and 150 km. For every assumption 7 scenarios were analysed to identify number of aggregated locations on the basis of number of available locations (see Figure B-8).

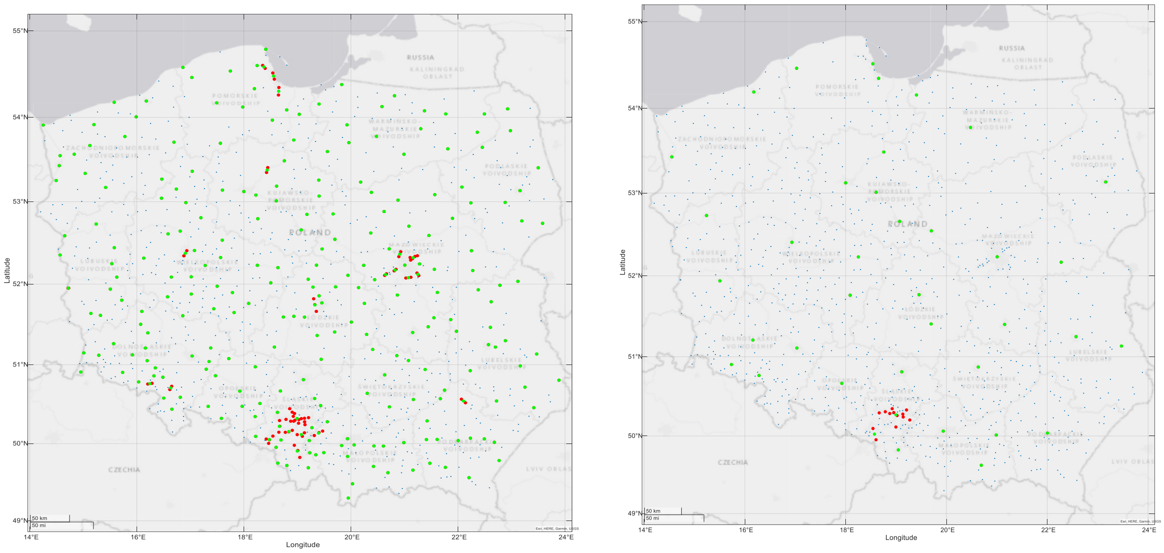


Figure B-8: Illustrative comparison of network nodes distribution for diameter of 50 km (left) and 150 km (right) for providing the same level of experience for users located in the whole area.

The achieved results showed the dependency between core network nodes density and level of user experience (e.g. end-to-end latency) for services built on multi-provider cloud continuum resources distributed over the region (on the example of country). The calculated network nodes density can be the basis for locating additional network nodes in the migration existing TN towards 6G network to ensure that basic services are provided with the same level of experience everywhere within the region. Conclusions from this analysis can be further expanded for the planning in the areas broader than a country, finally covering whole world, where 6G users can be offered with services at the same level of experience.

To achieve a fully connected world in the future 6G standard, it is necessary to extend network access and service coverage towards the entirety of the globe. From urban metropolitan centres all the way to remote and rural areas, one of the goals of 6G is to achieve a Ubiquitous Network [HEX223-D12], by supplying a high-quality network connection to mobile users, as well as access to important 6G network services and functions, in the realm of governmental crisis management, digital health and autonomous supply chains. Enabling this Hexa-X-II use case can be done by upgrading the current frameworks and architectures to better utilise the available network capabilities, but it can also be done by upgrading the surrounding infrastructures that support the network. For the latter case, enhancing 6G coverage can be done by integrating new domain paradigms, such as NTN, like High Altitude Platform Stations (HAPS) and Satellites, or Cloud-Edge-IoT (CEI) Continuum, into the 6G platform. CEI Continuum refers to the development of mechanisms and services to harmonise the infrastructures in all three of these domains, Cloud, Edge, and IoT. The IoT era fully cemented itself with the development of 5G, as the improvements of the network for massive machine type communications enabled the massive deployment of small and cheap mobile IoT devices, like sensors, capable of collecting and/or processing data in a resource-efficient manner. In addition, with enhancements such as those for ultra-reliable low latency communication, supported devices are capable of offering much faster and more reliable access to data in urban cities, enabling services like assisted driving, cobots and public safety applications. As coverage is increased, extending internet connectivity to rural areas, edge computing takes advantage of this distributed computing framework, using IoT devices as nodes of the network, to use the data collected in order to orchestrate specific workloads, tapping into a greater pool of available resources and using these resources optimally, while reducing implementation costs. On top of this, the Cloud domain enhances available computing and storage capabilities of the network, by offloading the effort required to run these workloads to available public and private cloud resources; this resource management is automatically done by the network, through intelligent AI/ML algorithms, that determine the most cost efficient way to deploy applications and services that require network resources. The 5G rollout was slower than anticipated by most Telco operators, leading to the development of services that don't rely on 5G Standalone (5G SA) technologies; this hurt the adoption of applications that rely on the capabilities of these specific network domains. The next telecommunication generation is focused on a unified 6G SA deployment, fully supporting from day one the applications available on the network, while also increasing overall connectivity, resilience and continuity of the offered network services. Multi-cloud federation is key towards achieving this unified 6G network. Different cloud domains have a higher resource complexity, as different Telco providers offer different computing, storage and network resources; 6G networks must be able to federate these available resources located in different clouds on-demand, allowing them to be integrated in an agnostic way and used for any workload requested by the network. By offloading network tasks towards the cloud, we can improve the performance and resource-efficiency of the deployed applications and services. While the cost of extending current infrastructures for the purposes of a Ubiquitous Network is prohibitively high, by tapping into CEI technologies and mechanisms, we can reach the edge and far-edge, which will increase the availability and reliability of the 6G network, while decreasing end-to-end latency of the services offered. In the case of Smart Cities, we can increase the digital inclusion of remote and rural areas, using this increased coverage to support a wider range of data collection, providing local governments better and more analytics. Stakeholders can, then, vendor their cloud resources to the benefit of more people, while distributing the Smart City applications and services that are running on the network in a cost-efficient way.

B.3 Self-sensing concept (QT4.1)

This section dals with the QT4.1 "(<1 m at 90th percentile) Radio/Communication based sensing precision in mid-band and (<10 cm at 90th percentile) Radio/Communication based sensing precision at 100 GHz to detect a moving human-sized object at 10 m distance", see Table A-5. The details of this research is outlined in Section 3.2.4. It contributes to achieving centimeter-level precision in 6G sensing by leveraging mmWave sensing integrated with traditional LiDAR systems. The approach, termed waveSLAM, utilizes 60 GHz mmWave radios with a bandwidth of 2.1 GHz that perform simultaneous self-sensing and environmental mapping. This technique enhances the mapping precision through the integration of Time of Flight (ToF) and Angle of Arrival (AoA) calculations with advanced algorithms like mD-track.

ToF via Fine Time Measurement (FTM) enables distance estimation between devices without requiring clock synchronization, providing robust performance even in challenging conditions. Meanwhile, AoA estimation using a Uniform Rectangular Array (URA) ensures accurate directional sensing by iteratively isolating and analyzing multipath signals. These methods ensure precise localization, a critical requirement for 6G applications.

The research also showcases how mmWave sensing compensates for the limitations of LiDAR in environments with low visibility or high translucence, such as glass surfaces or foggy conditions. By merging data from both systems, waveSLAM ensures reliable point clouds, filtered and refined for accuracy. The experimental results indicate that mmWave-based SLAM maintains sub-10 cm (see Figure 3-47) error margins in distance and angle estimations over a range of conditions, demonstrating its potential to meet the quantitative targets of centimeter-level precision vital for 6G technologies.

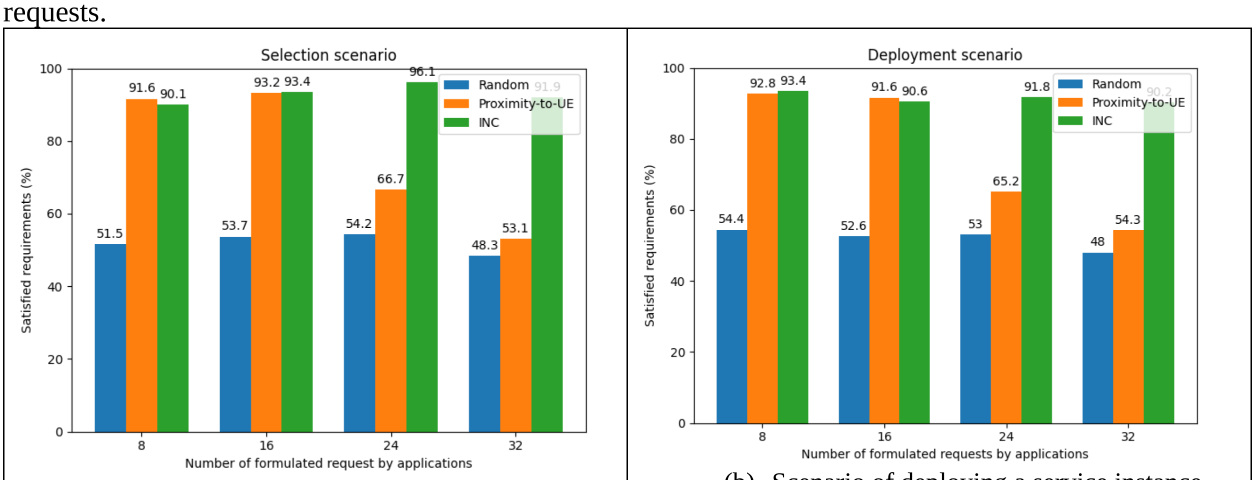
B.4 Objective 4: Energy efficiency, latency, bit rate or capacity (QT 4.2)

This section deals with the objective 4 and the QT4.2 (see Table A-5): "(>20%) improvement in performance in at least one of energy efficiency, latency, bit rate or area capacity through use of sensing, localisation, traffic data, or mobility patterns for AI-based optimisation in selected use cases".

Service placement/ISAC-related E2E latency.

Different applications, where a user is accessing a service deployed in a compute site, are associated with stringent requirements in terms of network and compute. Addressing these requirements would require a new approach to jointly optimize these two resources. We have introduced the INC approach that aims to jointly optimize network and compute processes in a way to meet application requirements (see section 3.4.1 for more details). This approach implies the introduction of a new functionality that is responsible for collecting network and compute metrics, receiving application request (including delay & compute requirements), and deciding the optimal service or deployment location. Quantitative evaluation has been performed against Proximity-to-UE approach (i.e., assigning a service that is close to the UE, based on their IP addresses), which reflects the current specification in Edge Computing. We have evaluated the ratio of the satisfied latency requirements considered different number of formulated requests by the users. A satisfied latency requirement would mean that the UE has been associated with a service that has a delay that is less or equal of the requested delay. The obtained results are depicted in Figure B-9. As we can see from this figure, both INC and Proximity-to-UE outperform random approach. In addition, for smaller numbers of requests, Proximity-to-UE and INC provide similar results. This is valid for the service selection scenario as well as for service deployment scenario. This is explained by the fact that the amount of compute resources available at the edge clouds that are in the proximity to UE is enough to accommodate all requirements of the applications (i.e., delay is always small). However, as the number of formulated requests increases, the proposed INC approach outperforms the Proximity-to-UE solution. Indeed, as the edge clouds that are in the proximity to UE cannot satisfy all the requirements, a global optimization that takes into account both network and compute requirements is needed. Unlike the Proximity-to-UE solution that selects the nearest locations, the INC approach collects network and compute metrics from the different locations and perform optimal decision.

In summary, the evaluation showed that as the number of formulated requests increases, the INC approach outperforms Proximity-to-UE and can achieve more than 35% enhancement of the number of satisfied latency requests.



(a) Scenario of selecting a service instance among pre-deployed services

(b) Scenario of deploying a service instance

Figure B-9: Evaluation of ratio of satisfied requirements considering INC approach

Device compute offloading energy savings

As explained in section 3.3, an Offloading Node (ON) may offload the computation of specific processes to a Compute Node (CompN) in order to save power or due to its limited compute capabilities. Given that the ON has decided to offload a computation, the ON can avoid unnecessary monitoring tasks or even unnecessary transmissions, if it is aware of the size of the compute results (Z [bytes]) and of the maximum affordable delay (T [ms]). DL transmission of compute results can be planned before they are ready, i.e., via configured scheduling. CompN can schedule the compute response accordingly even before the actual response is ready and informs ON about the planned DL transmission immediately, to increase the ON's sleep time, as shown in Figure B-10. ON can then enter the power saving state, and, upon planned DL transmission, wake up and receive on the planned DL resources directly. If the maximum affordable delay of T [ms] is satisfied, there may be several HARQ transmissions of that planned DL. In the case of compute results are not ready at CompN within the T [ms], the offloading fails and CompN indicates to ON that the response is not ready, discarding any late compute results. The proposed computation offload procedure enables increased sleep time of the ON resulting in energy efficiency.

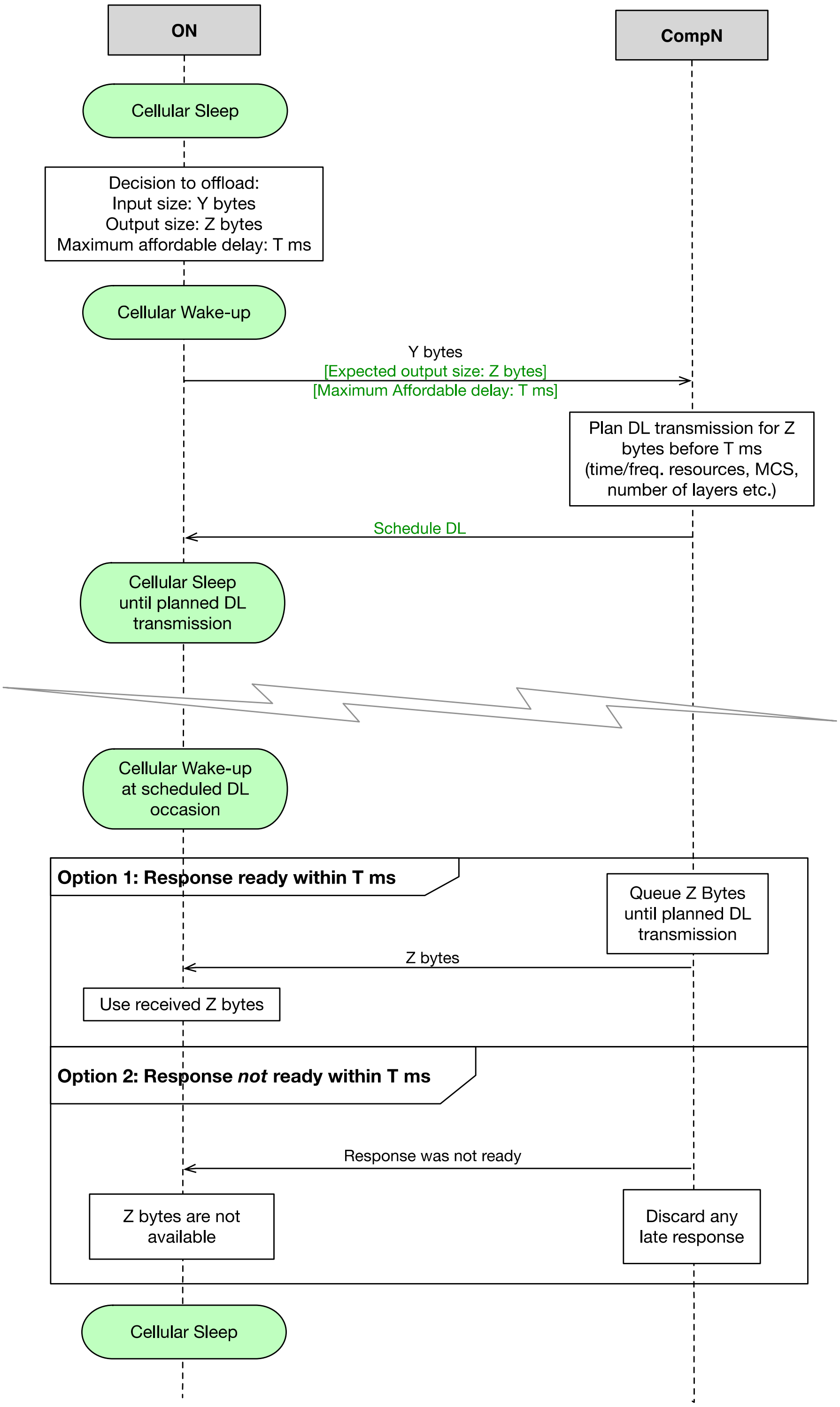

ON
CompN
Cellular Sleep
Decision to offload:
Input size: Y bytes
Output size: Z bytes
Maximum affordable delay: T ms
Cellular Wake-up
Y bytes
[Expected output size: Z bytes]
[Maximum Affordable delay: T ms]
Plan DL transmission for Z bytes before T ms (time/freq. resources, MCS, number of layers etc.)
Schedule DL
Cellular Sleep until planned DL transmission
Cellular Wake-up at scheduled DL occasion
Option 1: Response ready within T ms
Queue Z Bytes until planned DL transmission
Z bytes
Use received Z bytes
Option 2: Response not ready within T ms
Response was not ready
Z bytes are not available
Discard any late response
Cellular Sleep

Figure B-10: An example of flow and message content exchange for Compute offloading with planned DL.

In [HEX224-D33], a dynamic device offloading experiment was performed. The purpose of the demonstration is to examine the performance characteristics of offloading, especially the power consumption, execution time and network utilization. As can be seen, the power consumption in the mobile device is sharply reduced for the offloading periods, with roughly 50% less power consumption in the remote offloading case.

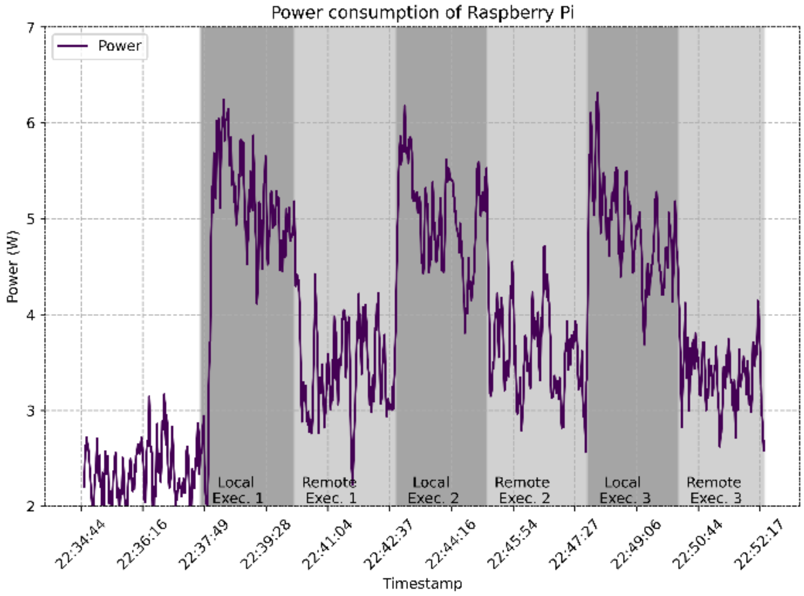


Figure B-11: Dynamic offloading of a CPU demanding operation from the device to the network.

The Figure B-11 shows the consumed power when there is local (when there is no offloading) and remote (offloading to the network). However, this comes at the cost of the use of network communication resources, to ensure that the module running in the execution environment in the network is synchronized with the remaining application running on the device.

BCS consumer application placement optimisation

The optimization of Beyond Communication Services (BCS) consumer application placement in 6G networks significantly enhances overall network performance by strategically leveraging network traffic data. Utilizing AI-based Genetic Algorithms, the placement of application functions is meticulously optimized by analyzing factors such as latency sensitivity, energy usage, and resource availability across various network layers. This data-driven approach ensures that application functions are deployed at the most suitable nodes within the 6G compute continuum, from the extreme edge to the core, thereby minimizing end-to-end latency and reducing power consumption. Real-time network traffic and sensing information are integrated to this process, facilitating informed decisions that optimize node selection for application deployment.

Key performance improvements include up to a 50% decrease in end-to-end latency, which is critical for maintaining real-time responsiveness in latency-sensitive applications, and up to a 55% reduction in power consumption, supporting the overarching goal of sustainable and energy-efficient network operations. These advancements are achieved by strategically positioning application functions closer to data sources and end-users, minimizing communication delays, and selecting energy-efficient nodes to reduce unnecessary data transmissions. The impact of these optimizations extends beyond mere performance enhancements, aligning seamlessly with 6G sustainability goals and meeting stringent Quality of Service (QoS) requirements across diverse use cases. This holistic approach not only boosts overall network performance but also ensures scalability and adaptability, enabling the 6G network to effectively support a wide array of applications and industries with enhanced reliability and efficiency. The impact of this optimization is evident in simulation results where a genetic optimization is compared to a greedy algorithm, as illustrated in the Figure B-12, and as shown, the genetic algorithm consistently outperforms the greedy placement strategy. Latency is significantly reduced, achieving near-linear performance with the number of applications, while energy consumption shows a marked reduction, supporting sustainable operations. Moreover, the genetic algorithm minimizes data exposure and improves resource utilization, enhancing both security and efficiency.

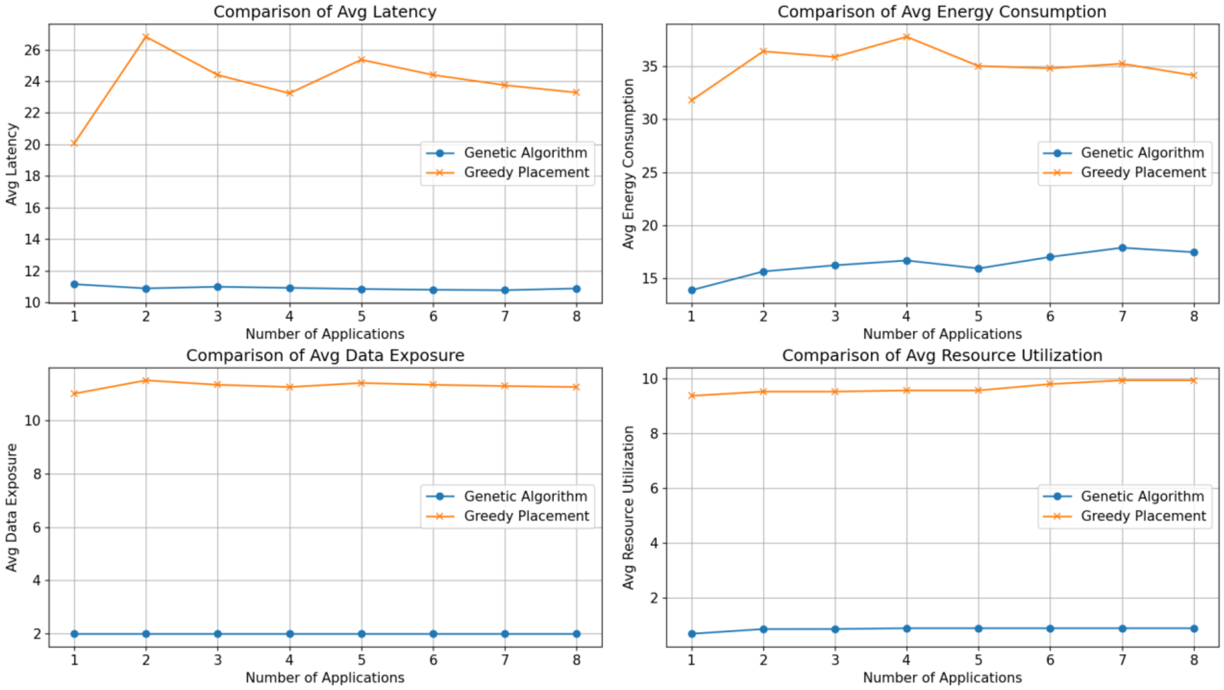


Figure B-12: Performance comparison of Genetic Algorithm vs. Greedy Placement for latency, energy consumption, data exposure, and resource utilization.

Split learning and energy savings

The energy saving in a collaborative QoE estimation use case is quantified via model layer offloading and generalization. In component PoC #B.2, the estimator model is a split neural network. One part of the neural network is deployed in the network while the other part in the application function in a split learning setting. This allows both network and application functions to train the local portion of the global NN estimator model while keeping both data and model locally. Moreover, this setting allowed offloading model layers from application function to the network or vice versa to adapt the dynamic changes of the compute resources in both data domains. We demonstrated offloading 2 out of 4 layers from application function to the network. The offloaded layers are aggregated, meaning that if there are 2 layers offloaded from *N* devices, there will still be 2 additional layers in the network. We observe 73% reduction in energy consumption at the application Additional 2 layers increased the energy consumption at the network. However offloading layers from multiple tail NN models are performed via aggregation of the offloaded model layers at the network, hence the increase in the energy consumption does not increase with the number of tail nodes, i.e., applications. In Figure B-13 the estimated overall energy consumption before and after layer offloading with the increasing number of devices is given. The assumption here is that there are 2 tasks running in every device.

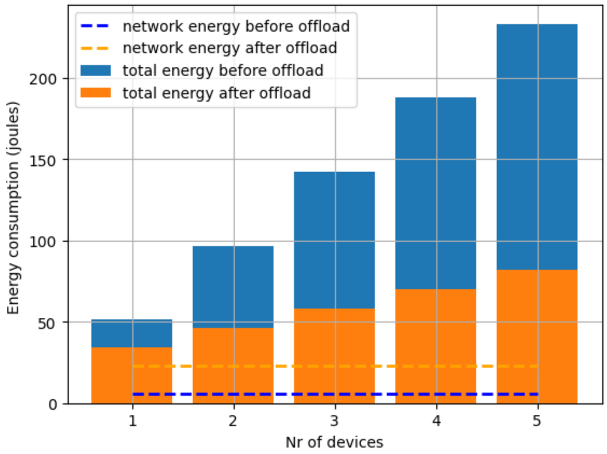


Figure B-13: The device, network and overall energy consumption before and after offloading model layers from application to the network are depicted.

B.5 Objective 5: Reducing OPEX by using zero-touch automation (QT5.2)

This section describes how to achieve the QT5.2 in Table A-5: "(>25%) reduction in OPEX by using zero-touch automation.

Zero touch automation

The developed framework and procedures will enable zero-touch automation by providing fully automated and flexible management of the AI/ML models. Indeed, the developed solution automates AI/ML model training, deployment, performance monitoring and update and re-deployment. Those procedures help maintain the performance of AI/ML models that can be used in the network system (e.g. Zero-Touch network and service orchestration) as well as the services deployed on the network infrastructure.

The architecture and workflows also enable an improved automated ML model lifecycle management by considering, for each model, different dependencies with other models, Digital Twins and datasets. Consequently, some model degradations can be predicted in advance and proactively mitigated based on events occurring for other dependent models or datasets (e.g. performance degradation or update).

Therefore, the response time to AI/ML model performance degradation can be reduced, or in best case scenarios the performance degradation can be completely avoided, and AI/ML model overall performance can be maintained over its lifecycle to adapt to dynamic scenarios in the network infrastructure. Further, the model training time can be reduced by re-using available pre-trained models and datasets whenever possible.

Federated Learning

This section examines the impact of effective radio resource allocation on the energy consumption of wireless Hierarchical Federated Learning (HFL) networks, analysing numerical results from the related study titled "On the Accuracy-Energy Trade-off in Wireless Hierarchical Federated Learning" in Section 3.1.2.3. In this study, to optimize wireless HFL networks, joint user-to-edge-server association and uplink transmission power allocation is performed using a distributed game-theoretic approach. Various solutions are analysed: (a) **Satisfaction Equilibrium (SE)**, where a target accuracy-time-energy trade-off is achieved; (b) **Minimum Efficient SE (MESE)**, which ensures SE while minimizing energy costs; (c) **Random Association**, with randomly determined user associations and transmission powers; (d) **Closest Server**, where users are associated with their nearest server, and power allocation is optimized to achieve SE; and (e) **Closest Server MESE**, where the same setup targets MESE.

Notably, as seen in Figure B-14, the Closest Server MESE achieves up to a 90% reduction in energy consumption (which can be seen as an OPEX reduction), highlighting the significance of selecting an appropriate radio resource optimization strategy. However, these results must be balanced with other critical metrics, such as HFL convergence time and global model accuracy, to determine the most effective and holistic resource allocation solution.

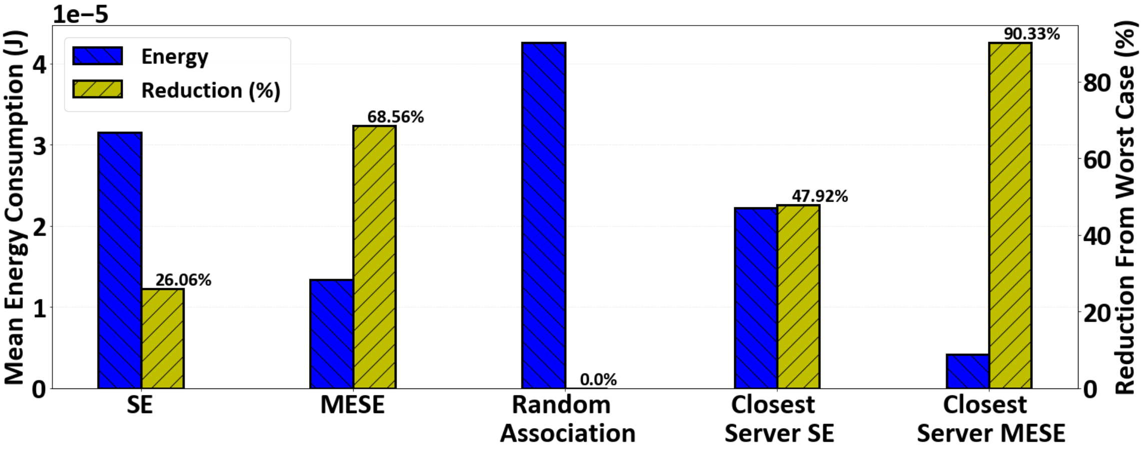


Figure B-14: HFL network energy consumption in Joule and percentage reduction in energy consumption achieved under different scenarios. The percentage reduction is calculated relative to the worst-case energy consumption observed in Random Association scenario.

OPEX reduction via improved Orchestration of the Cloud Continuum

The evolution towards 6G is bringing applications and services to be deployed and operated across highly distributed and dynamic infrastructures of the cloud continuum. This introduces new challenges in the area of management and orchestration of end-to-end resources, that need to take into account multiple available technologies, multi-stakeholder infrastructures, and heterogeneous resource capabilities, including resource-constrained environments in the extreme-edge domains. On the other hand, it opens opportunities to improve the efficiency and cost effectiveness of network and services operations, especially when introducing advanced cloud continuum orchestration integrated with zero-touch automation mechanisms.

The solutions proposed for managing multi-cluster resources, decentralized orchestration, and constrained devices in ETSI MEC scenarios go in the direction of bringing a transformative approach that addresses the increasing complexity and cost challenges faced by network operators. The widespread adoption of these technological enablers is indeed expected to bring substantial benefits for the economic and operational aspects of next-generation networks. Indeed, quantitative studies evidence that resource optimization techniques in distributed cloud architectures can reduce energy consumption by 30% to 40%, which directly translates to cost savings, particularly as energy expenses account for a significant portion of network OPEX in telco infrastructures. Similarly, zero-touch enabled cloud continuum resource orchestration mechanisms improve provisioning, performance optimization and fault management processes, which in next-generation 6G networks has the ability to reduce manual operational tasks by 70%, directly contributing to substantial OPEX reductions [TAL+22]. Moreover, a report from Capgemini indicates that network operators that introduced zero-touch autonomous mechanisms and transparent continuum orchestration achieved during 2022-2023 a 18% reduction of OPEX with a 20% increase of operational efficiency [CAPG].

Distributed Computing Reduction of OPEX

Operational Expenditures (OPEX) represent the set of costs required, from the maintenance procedures, labour and resources, among others, required to support 6G networks. The development of 6G enablers is expected to bring improvements towards overall network capabilities such as speed, latency and bandwidth throughput over the current 5G means. While most of these enablers are still in development, these innovative technologies and architectural changes promise significant benefits towards reducing OPEX for the coming 6G framework.

These new network capabilities of 6G plan to open up these new service types. However, supporting the extension of the current physical infrastructures towards fulfilling 6G requirements is prohibitively expensive for public and private entities. To that end, a shift towards Distributed Computing and Edge Computing is taking place, by sending this critical role of processing data and communications closer to the user, reducing the load on costly data centres and cloud servers. This is supported by shifting the network paradigm towards multi-access edge computing (MEC) devices and cloud resources, enabling these devices to process, store and deliver services at the edge of the network, reducing latency times in communications, reducing the costs of local physical data centres that would be required to support these computational requirements and improving network efficiency, by dynamically adjusting resource usage to the requirements of the network.

5G brought enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC) and massive Machine Type Communications (mMTC), innovations that were central to the transition from 4G to 5G, enabling the development of applications towards Smartphones, IoT Sensors and Drones. 6G plans to extend these innovations into the paradigm of the new device types and new services provided under the 6G umbrella:

- Mobile Broadband Reliable Low Latency Communication (mBRLLC), superseding eMBB and URLLC, as these technologies are not sustainable for the high volume of reliable and low latency data that 6G use cases like Extended Reality (XR) and Connected Robotics and Autonomous Systems (CRAS) require. This new service provided by 6G will be able to deliver high performance within the data rate, reliability and latency requirements of the applications.
- massive Ultra Reliable Low Latency Communications (mURLLC), given that URLLC in 5G was limited to specific applications that were operated considering a close proximity between IoE devices, mURLLC must scale up this service availability to the requirements of 6G, expanding the reliability and latency capabilities towards a broader range of network topologies. Human Centric Services, a new class of services that require a reliable and low-latency high data rate towards improving Quality of Personal Experience (QoPE) targets, such as Brain Computer Interface and Haptic devices, in which performance is closely determined by the specificity of their human users. Multi Purpose 3CLS (Communication, Computing, Control, Localisation and Sensing) and Energy Service
- To further extend the benefits of Edge Computing towards the Smart City vision of the future, our framework intends to implement a Federated Learning system, that takes the data gathered on the distributed IoT devices scattered along the city verticals to train AI/ML models; by using a Federated Learning scheme, we enable model training that can be used not only to optimise network services, but also enable other shareholders to use the network to train their workloads, all while preserving the privacy of the raw data collected on these edge devices.

Federated Learning brings great benefits towards the reduction of OPEX, by reducing transmission costs, as batches of raw data will not be uploaded, and instead only model gradients used for model aggregation and re-training cycles; the size of these model gradients is also lower, which makes storing the information easier for the available infrastructures. The nature of IoT devices allows for horizontal scalability, making it cheaper to extend the coverage of the network while reducing the latency for model updates and connected devices; smart device management algorithms can be deployed to optimise energy consumption, by maximising device efficiency, and reduce necessary costs of maintenance, as human intervention is reduced by automatically optimising device efficiency.