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 D3.3
Initial analysis of architectural enablers and framework

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:	30/04/2024	Version:	1.0
Project reference:	101095759	Call:	HORIZON-JU-SNS-2022
Start date of project:	01/01/2023	Duration:	30 months

Co-funded by the European Union

6GSNS

Document properties:

Document Number:	D3.3
Document Title:	Initial analysis of architectural enablers and framework
Editor(s):	Mårten Ericson (EAB), Panagiotis Botsinis (APP), Merve Saimler (EBY), Ozgur Akgul (NFI), Sokratis Barmpounakis (WIN), Milan Groshev (UC3)
Authors:	Ozgur Akgul (NFI), Bassem Arar (TUD), Michael De Angelis (NXW), Sokratis Barmpounakis (WIN), Jaap van de Beek (LTU), Giacomo Bernini (NXW), Sonai Biswas (TUD), , Panagiotis Botsinis (APP), Pere Garau Burguera (AAU), Panagiotis Charatsaris (ICC), Panagiotis Demestichas (WIN), Maria Diamanti (ICC), Toni Dimitrovski (TNO), Sameh Eldessoki (APP), Mårten Ericson (EAB), Mohammad Asif Habibi (RPTU Kaiserslautern), Apostolos Kousaridas (NGE), Milan Groshev (UC3), Alperen Gundogan (APP), Hasanin Harkous (NGE), Hamed Hellaoui (NFI), Selim Ickin (EAB), Paola Iovanna (EAB), Grigorios Kakkavas (ICC), Bahare M. Khorsandi (NGE), Slawomir Kukliński (OPL), Gerald Kunzmann (NGE), Hannes Larsson (EAB), Marvin Manalastas (NFI), Beatriz Mendes (UBW), Swaraj S. Nande (TUD), Antonio de la Oliva (UC3), Ece Ozturk (NGE), Torgny Palenius (SON), Symeon Papavasliou (ICC), Ignacio Labrador Pavón (ASA), Janusz Pieczerak (OPL), Bartosz Rak (OPL), Merve Saimler (EBY), Hans D. Schotten (RPTU Kaiserslautern), Erin Seder (NXW), Vivek Sharma (SON), Mohammad Soliman (NGE), Heiko Straulino (NGE), Halina Tarasiuk (OPL), Olav Tirkkonen (AAU), Nassima Toumi (TNO), Vasilis Tsekenis (WIN), Antonio Varvara (TIM), Stefan Wänstedt (EAB), Zi Ye (LTU), Milan Zivkovic (APP), Marcin Ziółkowski (OPL)
Contractual Date of Delivery:	30/04/2024
Dissemination level:	PU[1]
Status:	Final
Version:	1.0
File Name:	Hexa-X-II_D3.3_v1.0

Revision History

Revision	Date	Issued by	Description
0.1	2023-09-25	Hexa-X-II WP3	Template for Deliverables/IRs

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

PU = Public

0.2	2023-12-15	Hexa-X-II WP3	Cleaned-up version
0.3	2024-01-16	Hexa-X-II WP3	Reduced pages, added Appendix
0.4	2024-02-1	Hexa-X-II WP3	Internal full review
0.5	2024-02-15	Hexa-X-II WP3	Addressing review comments
0.6	2024-03-02	Hexa-X-II WP3	External review
0.7	2024-03-26	Hexa-X-II WP3	GA version
0.8	2024-04-15	Hexa-X-II WP3	GA version, minor updates
1.0	2024-04-28	Hexa-X-II WP3	Final version submitted to EC

Abstract

This is the second public deliverable from Hexa-X-II project work package 3 – "Initial analysis of architectural enablers and framework". This deliverable develops and analyses innovative enablers required for a 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 enable ubiquitous coverage.

Keywords

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

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 second public deliverable from WP3, called D3.3 “Initial analysis of architectural enablers and framework”.

The main objective of this deliverable is to analyse the WP3 enablers introduced in previous deliverable and at the same time identify and define additional requirements that are important for the 6G architecture. 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.

The deliverable thereafter analyses the artificial intelligence (AI) enablers for the 6G data-driven architecture. The enablers comprise architectural means and protocols, machine learning (ML) Operations (MLOps), data operations (DataOps), AI as a service (AIaaS), and intent-based management. The AI enablers form a robust framework for seamlessly integrating AI into the compute continuum of 6G networks.

To enable flexibility without increasing complexity, 6G needs an easily deployable architecture of modules (i.e., network functions) that can scale to current needs. The deliverable analyses the network modularisation enablers, which focus on different granularities of a module as well as on the evolution of the modules as a result of different deployments and use cases. The idea is to optimize the 6G network functions or modules according to certain key performance indicators such as latency (end to end), procedure completion time, and control plane signalling, while at the same time reduce complexity and improve sustainability.

The deliverable thereafter analyses the enablers for new access and flexible topologies which consist of network of networks, multi-connectivity and end to end context awareness management. The network of networks enabler aims to develop so-called subnetworks and integrate these sub-networks into a seamless and ubiquitous communication system. One main component for this is to inherently support non-terrestrial networks (NTN) in 6G, with improved interoperability between satellites and integration with the terrestrial networks. Another type of subnetwork is a network of User Equipment (UE) based on mutual trust, where a management node, i.e., a more capable UE, controls other UEs and helps them perform certain procedures. The deliverable also investigates methods to improve the multi-connectivity as well as aggregating different radio technologies, such as 6G cellular and wireless local area network (WLAN). The context awareness enabler aims to allow network components to dynamically adapt to the context to ensure the expected end to end quality of a service (QoS) and use resources in a more efficient manner. One way to introduce context awareness to the transport network is that a resource orchestrator creates an abstracted view of transport resources and employs a software defined transport controller for resource management, ensuring that the QoS associated with a network slice is met.

The beyond communication enabler deals with how to realize new 6G services such as sensing and compute offloading, and how to expose resulting data and relevant service capabilities in a secure, privacy-preserving and efficient manner. To reduce the overhead from data exposure (especially from sensing), the deliverable proposes new functionality that is needed to aggregate and fuse data as well as ensure data privacy and trust.

One way to improve sensing quality is obviously to carry out more radio measurements prior to exposure of the measurement report to the requesting application, preferably measurements that are geographically distributed. Involving more network nodes (UEs or base stations) naturally leads to architectural challenges of centralized vs. distributed inference and processing of the measurements. The provision of sensing services by next generation communication systems necessitates the introduction of a management function of the sensing. The sensing management function controls the sensing process such as which nodes to involve, processing of sensing data and facilitating an efficient coordination of sensing procedures. As far as compute offloading is concerned, the scope includes investigating new procedures for offloading computations and for controlling which compute node will perform the computations.

The off-the-shelf cloud is suitable for a big subset of multimedia human-scale applications, but it has its limitations when it comes to supporting the upcoming latency sensitive 6G use cases. The cloud transformation deals with how to develop the 6G cloud platform for the telecom system. The deliverable proposes an enabler for the integration and orchestration of the compute continuum, including extreme edge resources. The enabler analyses the needed architectural interfaces and components for seamless orchestration and management of the complete compute continuum composed of cloud, edge and extreme edge resources. The analysis results in a proposal of new architectural mechanisms for the Multi-access Edge Computing (MEC) framework to incorporate the extended compute continuum and account for the extreme edge devices. Multi-domain/Multi-cloud federation is the capability to aggregate cloud services provided by multiple domains and providers into a single, coherent cloud. To this end, the cloud continuum should provide intent-based interfaces for cloud services and the core network should provide intent-based interfaces for network services.

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

Table of Contents

List of Tables

List of Figures

Acronyms and abbreviations

Term	Description
5GC	5G Core
5GS	5G System
5QI	5G QoS Identifier
6GC	6G Core
6GS	6G System
A-VIM	Aggregated VIM
ADC	Aggregated Data Centre
AF	Application Function
AI	Artificial Intelligence
AIaaS	AI as a Service
AMF	Access and Mobility Management Function

AMR	Autonomous Mobile Robot
AoA	Angle-of-Arrival
API	Application Programming Interface
ATSSSM	ATSSS processing Module
AUSF	Authentication Service Function
BC	Boundary Clock
BCN	Beyond Communication Network
BCS	Beyond Communication Service
BS	Base Station
C-RAN	Centralized-RAN
CA	Carrier Aggregation
CaaS	Compute-as-a-Service
CCN	Compute Offload Controlling Node
CDF	Cumulative Distribution Function
Cgate	Classical Gate
CioTM	Cellular IoT Module
cMEC	constrained MEC
CN	Core Network
CNF	Cloud Native Network Function
CNFD	Cross-Network Function Distributed
CompN	Computing Node
COTS	Commercial Off-The Shelf
CP	Control Plane
CPU	Central Processing Unit
CSP	Communications Service Provider
CU	Centralized Unit
CUPS	Control and User Plane Separation
D-MIMO	Distributed MIMO
DataOps	Data Operations
DC	Dual Connectivity
DCF	Data Centre Feature
DCME	Data Centre Monitoring Engine
DCN	Data Centric Networking
DFP	Data Flow Programming
DL	Downlink
DLM	Downlink Module
DMM	Distributed Mobility Management
DN	Data Network

DNSSS	Distributed Network Stakeholder Support Service
DP	Data Plane
DPIM	Deep Packet Inspection Module
DRP	Data Centre-Resource Layer Portal
DS	Deployment Service
DSS	Dynamic Spectrum Sharing
DU	Distributed Unit
E-5GC	Evolved 5G Core
E-UTRA	Evolved Universal Terrestrial Radio Access
E2E	End-to-End
ECDF	Empirical Cumulative Distribution Function
EMR	Early Measurements Report
EN-DC	Evolved Universal Terrestrial Radio Access and New Radio Dual Connectivity
EPC	Evolved Packet Core
EPM	Enterprise specific Processing Module
FD	Functional Domain
FDC	Functional Domain Controller
FDD	Frequency Division Duplex
FDF	Federation of Functional Domains
FDFA	Functional Domain Federation Agent
FDFC	FDF Controller
FDFD	FDF Database
FDFM	FDF Manager
FDFRO	FDF Resource Orchestrator
FDFSO	FDF Service Orchestrator
FDM	Functional Domain Manager
FDRO	Functional Domain Resource Orchestrator
FDSO	Functional Domain Service Orchestrator
FEG	Far-Edge Gateway
FFT	Fast Fourier Transform
FIFO	First-In-First-Out
FL	Federated Learning
FOR	Federation Orchestrator
FP	Federation Plane
FR	Frequency Range
FRP	Federation Request Portal
FSM	Federated System Manager
FTM	Fine Time Measurement

FTN	Flexible Topology Node
GA	Genetic Algorithm
GCL	Guarded Command Language
gNB	gNodeB
GTD	Global Topology Database
GUAMI	Globally Unique AMF ID (GUAMI).
HFL	Hierarchical Federated Learning
HPC	High-Performance Computing
I-UPF	Intermediate UPF
IAB	Integrated Access and Backhaul
IDC	Input Data Compression
IFFT	Inverse Fast Fourier Transform
IID	Independent Identically Distributed
INC	Integration of Network and Compute
IPD	Infrastructure Partitions Database
IRN	Infrastructure Registry Node
IRS	Infrastructure Registry Service
ISL	Inter-Satellite Link
ISM	Ingress Steering Module
ISP	Internet Service Providers
ISPN	Infrastructure Status Prediction Node
ISPS	Infrastructure Status Prediction Service
JCAS	Joint Communications and Sensing
KPI	Key Performance Indicator
KVI	Key Value Indicator
LCM	LifeCycle Management
LEO	Low Earth Orbit
LIM	Lawful Intercept Module
LLS	Lower Layer Split
LMF	Location Management Function
LWA	LTE-WLAN Aggregation
LWIP	LTE-WLAN Radio Level Integration using IPSec Tunnel
M-NFVO	Modified NFVO
M-VIM	Modified VIM
M&O	Management and Orchestration
MAC	Medium Access Control
MBSM	Multimedia Broadcast Services Module
MC	Multi-Connectivity

MDAS	Management Data Analytics System
ME	Market Equilibrium
MEC	Multi-access Edge Computing
MEO	MEC Orchestrator
MEP	Multi-access Edge Platform
MEPM	MEC Platform Manager
MgtN	Management Node
MLOps	Machine Learning Operations
MN	Master Node
MNO	Mobile Network Operator
MP	Management Plane
MRSS	Multi-Radio Spectrum Sharing
MST	Minimum Spanning Tree
N6TM	N6 Tunnelling Module
NARD	NS Allocated Resources Database
NF	Network Function
NFV	Network Function Virtualization
NGAP	NG application protocol
NN	Neural Network
NOMA	Non-Orthogonal Multiple Access
NR	New Radio
NRF	NF Repository Function
NS	Network Service
NSA	Non-Standalone
NSCR	NS Charging Record
NSRD	NS Requirements Database
NSTS	Network Slice Topology Selection
NTN	Non-Terrestrial Network
NW	Network
NWDAF	Network Data Analytics Function
O-RAN	Open RAN
ODM	On-Demand Function
ON	Offloading Node
OSR	O-RAN Slice Request
OSS	Operation Support System
OTA	Over-The-Air
PbNF	Procedure-based NF
PCF	Policy Control Function

PCT	Procedure Completion Time
PHY	Physical Layer
PNF	Physical Network Function
PNI-NPN	Public network integrated Non-Public Network architecture
PoC	Proof of Concept
PoI	Partition of Infrastructure
PRTC	Primary Reference Time Clock
PSCell	Primary Secondary Cell
PTP	Precision Time Protocol
QC	Quantum Computing
Qgate	Quantum Gate
QKD	Quantum Key Distribution
QoCS	Quality of Compute Service
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RAT	Radio Access Technology
RB	Radio Bearer
RDAE	Resource Layer Data Analytic Engine
RedM	Redundancy Module
RegNF	Registration NF
RIC	RAN Intelligent Controller
RL	Reinforcement Learning
RLC	Radio Link Control
RNIS	Radio Network Information Service
RNTI	Radio Network Temporary Identifier
ROF	Resource Layer Orchestrated Function
ROP	Resource Layer Operator Portal
RP	Resource Plane
RRC	Radio Resource Control
RSE	Resource layer Security Engine
RSMA	Rate-Splitting Multiple Access
RU	Radio Unit
SBA	Service Based Architecture
SBI	Service-Based Interface
SCF	Sensing Control Function
SDLA	Semantic Deep Learning Analyzer
SDN	Software Defined Network

SDNDC	SDN Domain Controller
SE	Satisfaction Equilibrium
SeMF	Sensing Management Function
SeRS	Sensing Radio Signals
SESM	Semantic Edge Slicing Module
SFC	Service Function Chain
SLA	Service Level Agreement
SMF	Session Management Function
SN	Secondary Node
snCP	Subnetwork Control Plane
SPF	Sensing Processing Function
SR	Service Request
SRN	Service Registry Node
SRS	Services Registry Service
SSB	Synchronisation Signal Block
SubNW	Subnetwork
T-GM	Telecom Grandmaster
TCO	Total Cost of Ownership
TDD	Time Division Duplex
tMEC	traditional MEC
TN	Terrestrial Network
TNL	Transport Network Layer
ToA	Time of Arrival
ToD	Time of Departure
ToF	Time-of-Flight
TS	Traffic Source
TSN	Time-Sensitive Networking
UALCMP	User Application LCM Proxy
UAV	Unmanned Aerial Vehicle
UDM	Unified Data Management
UDR	User Data Repository
UDSF	Unstructured Data Storage Function
UE	User Equipment
UL	Uplink
UL-CL	Uplink Classifier
ULM	Uplink Module
UP	User Plane
UPA	User Plane Adapter

UPF	User Plane Function
URLLC	Ultra-Reliable Low-Latency Communications
VIM	Virtual Infrastructure Manager
VM	Virtual Machine
VME	VNF Migration Engine
VNO	Virtual Network Operator
WCA	WLAN-Cellular Aggregation
WFQ	Weighted Fair Queuing
WLAN	Wireless Local Area Network, also referred to as WiFi in the report
WN	Worker Node
WPO	Work Package Objective
WT	WLAN Terminal
XR	Extended Reality

1 Introduction

The Hexa-X-II project is a flagship initiative bringing together key stakeholders in Europe for 6G research, continuing the Hexa-X project work. 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. Furthermore, the Hexa-X-II project comprises several work packages that spans over important parts of the 6G ecosystem. 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 as well as new access and flexible topologies for improved reliability. This is the second public deliverable from WP3, called D3.3 “Initial analysis of architectural enablers and framework”.

1.1 Objectives

The main objective of this deliverable is to analyse the WP3 enablers and at the same time define new requirements that are important for the 6G architecture.

The long-term objectives of WP3 are presented in Table 1-1. The three Work Package Objectives (WPO) for WP3 are the 6G architecture for AI and beyond communications, how to combine the cloud technology for a modular, scalable and extendable architecture and an architecture for flexible topologies.

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 and 6
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 4 and 7
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 5

1.2 Enabler definition and Methodology

In this deliverable, the term enabler is used extensively. In WP3, enabler is defined as a technical area with common objectives. The technical area (or enabler) may contain several different types of solutions (or components) aiming for the same objective. For example, the enabler Network of networks in Section 5.1, aims to develop a seamless and ubiquitous communication system. 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. These solutions applied together aim to increase the coverage and reliability of the networks.

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 need to be modified in order to implement and introduce the enabler in the 6G architecture. For some of the enablers there is also a dedicated evaluation section 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 in the final deliverables, we will also aim 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 AI enablers for a data driven architecture. Chapter 4 describes the network modularisation, i.e., how to build an architecture of modules that can scale based on current needs. Chapter 5 describes new 6G radio access for flexible topologies and local communications. Chapter 6 describes the "beyond communications" services in 6G, such as sensing and computing. Chapter 7 describes the virtualization and cloud transformation. In Chapter 8, the status of the proof of concepts activities within WP3 are described. Chapter 9 summarizes and concludes the deliverable. Chapter10 contains the references and Chapter 11 provides an Annex with details from some of the enablers and studies.

2 6G Architecture overview

2.1 WP3 objectives mapping to the E2E 6G system blueprint

In [HEX223-D22], the Hexa-X-II view of the 6G E2E system blueprint was defined, see Figure 2-1. The blueprint consists of four layers:

- Application,
- Network-centric application,
- Network functions, and
- Infrastructure layers.

In addition to this, is also shows the so called “Pervasive functionalities” which can reside in any of the four layers.

Figure 2-1 will be used throughout this deliverable to indicate where different enablers are placed on the 6G End-to-End (E2E) system blueprint. In this section, we will make a similar assessment for the objectives listed in Table 1-1.

The objective WPO3.1 “6G architecture for AI and beyond communications” includes data driven functionalities such as Machine Learning Operations (MLOps), Data Operations (DataOps) and exposure of AI services. It also contains the Beyond Communication Network (BCN) functions such as sensing and compute offloading and how to expose these services. As can be seen in Figure 2-1, this objective belongs to the Network function layer, and will have new functionality in the Radio Access Network (RAN) and in the Core Network (CN).

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 and the E2E placement of functions (including the devices in some cases). The objective should also address 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 function layers.

The last objective is WPO3.3 “Architecture for flexible topologies”. This objective maps to the Network function layer (see Figure 2-1) and develops existing and new 6G subnetworks, including NTN, mesh networks and device networks. The objective also develops new multi-connectivity solutions.

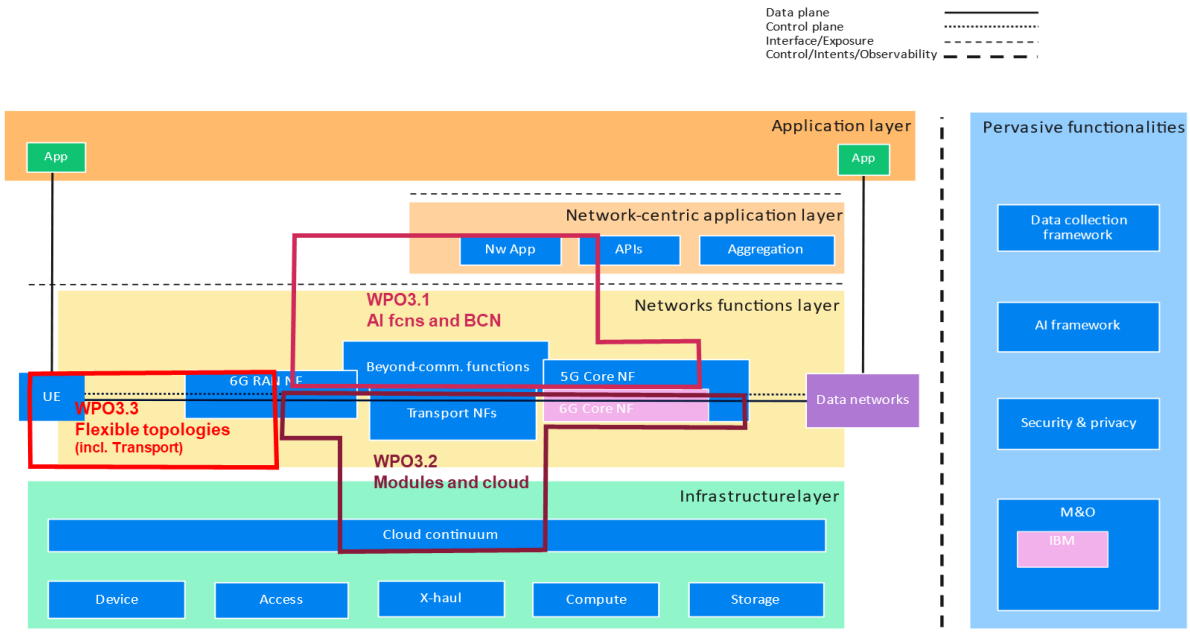


Figure 2-1 Illustrative mapping of WP3 objectives to the system blueprint

2.2 5G to 6G Migration

2.2.1 Introduction

For the migration from 4G to 5G, the industry was concerned that the envisioned deployment of the 5G RAN could be delayed due to a delayed introduction of the 5GC. Consequently, 3GPP agreed to deliver an "early drop" of New Radio (NR, i.e., 5G RAN) in Rel-15 which supports only the so-called Non-Standalone (NSA). The NSA option meant that the UE could connect to a 5G RAN i.e., either connected to the 4G CN (i.e., Evolved Packet Core (EPC)) or to the 5GC. Further on, several different options of dual connectivity between 4G and 5G was also standardised [38.300], where 4G RAN and 5G RAN can be used in any combination connected to either EPC or 5GC. One of these options that also is implemented and deployed in the network is Evolved Universal Terrestrial Radio Access (E-UTRA) and New Radio (NR) Dual Connectivity (DC) (EN-DC), where the 4G RAN is the master node and the 5G RAN the secondary node, and EPC is used as core network.

During the remainder of release 15 (Rel-15), focus was shifted so that the complete Rel-15 specifications also specify SA. So far, several years after 5G's launch, only a small fraction of networks has yet transitioned to the full SA 5G architecture, i.e., using both 5G RAN and 5GC.Therefore, to reduce the 5G-6G interworking and deployment complexity, the goal should be to have fewer architecture options specified for the deployment of 5GC as a 6G requirement. This would also avoid delays introducing key 6G services. As a result, the CN for 6G could be an evolution of the 5G CN so that the networks can gradually extend the support for new 6G services without the need to replace the CN.

The 6G radio access should support a single-RAT architecture only, i.e., a 6G UE that connects via the 6G radio interface establishes a connection to the CN for 6G without any complex inter-RAT multi-connectivity. Interoperability with 5G and older standards could be managed via existing core network handover or reestablishment. The above considerations aid in enabling the 5G-6G migration with a focus on a selected set of key architecture options to be specified, that will later be developed, and deployed. The 6G RAT will support operation on newly assigned 6G spectrum but also on all bands supported by 5G which then enables dynamic spectrum sharing with 5G. Thus, operators can use existing deployments both for 5G and 6G and dynamically share the spectrum resources between 5G and 6G as needed.

The main architecture options evaluated for 6G migration (cf. Figure 2-2) are as follows.

- **Option 1**: 6G RAN is connected to 5G RAN using DC. One of the objectives of 6G RAN is to enhance capacity, while in this option base capacity and coverage is provided by 5G. The core network is based on 5GC. Note that some updates to 5GC are still needed to support the introduction of 6G RAN as secondary RAT. However, option 1 is less likely to be selected for standardization since it requires 6G to support more than one RAT and will also slow-down the introduction of the new 6G services.
- **Option 2**: 6G RAN is deployed standalone and connected to an Evolved 5GC (E-5GC) (see section 2.2.2). 6G intra-RAT multi-connectivity using an enhanced carrier aggregation (CA) and/or 6G-6G DC (see more in section 5.2) can be used to combine capacity and coverage bands that are dynamically shared with 5G via Multi-Radio Spectrum Sharing (MRSS) [HEX223-D43].
- **Option 3**: 6G RAN is deployed in stand-alone mode with a 6G Core (6GC) (cf. Section 2.2.2), which allows for non-backward compatible changes such as a completely new RAN-Core or NAS interface or significant refactoring between RAN and Core functions. Depending on the amount of refactoring, network functions may be alike (e.g., 5G and 6G AMF), while the 6GC would still have many commonalities with the 5GC such as the Service Based Architecture (SBA) framework or shared network functions like the Unified Data Management (UDM), PCF, Network Repository Function (NRF), etc., can exist.

Mobility/inter-working between 5GS and 6G System (6GS) is core-based and realized via inter-RAT hand-over. Note that although the different general architecture options are presented here for the sake of completeness, a decreased set of options considering the 6G architecture E2E blueprint is presented in [HEX223-D22].

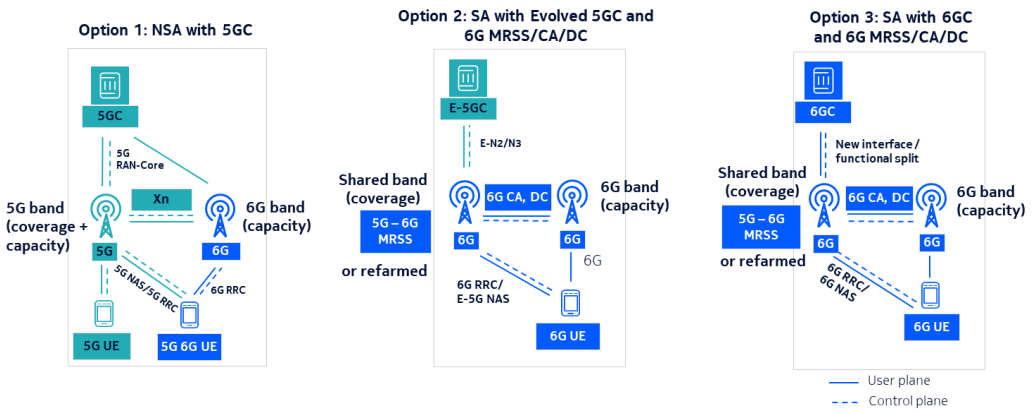


Figure 2-2 General architecture options for 6G.

2.2.2 Evolved 5GC and 6GC

The E-5GC is built upon 5GC NFs which can be enhanced to handle 6G requirements. Moreover, new 6G NFs can be defined within E-5GC when justified. The exact list of the shared/dedicated functions is subject to further study, e.g., use of evolved or shared 5G Network Functions (NF) and definition of new 6G NF(s). The E-5GC serves both legacy 5G RAN and new 6G RAN while supporting the 5G and 6G features at the same time (with the latter requiring 6G UEs and 6G RAN to be used). In the design, 5G SBA framework is continued to be used in both the E-5GC and the 6GC. 5GC and 6GC will have shared network functions such as UDM, NRF and User Plane Function (UPF). Additionally, new network functions or new services for existing 5G NFs (i.e., evolved 5G NFs or 6G NFs) can be added to support new 6G use cases and applications. Figure 2-3 depicts how this extensibility can be utilized for an evolution of the 5GC to introduce 6G functionalities in a backward compatible manner.

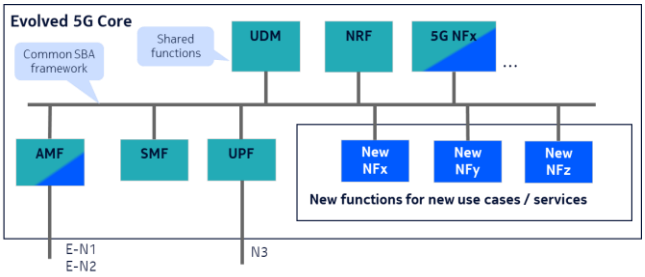


Figure 2-3 Evolved 5G core architecture (Option 2)

New 6G functionality is introduced by adding new NFs (e.g., NFx in Figure 2-3) or by enhancing 5GC NFs with new functionality, i.e., “shared NFs” (illustrated as blue triangles). The new 6G NFs are accessed only by new 6G UEs, 6G RAN, or new 6G services, while new functions / services can access and reuse the services provided by 5G network functions (e.g., UDM, NRF) through the common SBA framework. However, larger refactoring of functional splits (e.g., changing RAN-Core split) between network functions would render E5GC infeasible and break backward compatibility. In this case, a clear separation between 5GC and 6GC should be preferred as depicted in Figure 2-4 (Option 3), where only few NFs would be shared. Regardless of the migration option, the granularity at which core components interact with each other will continue to stay at NF level definition.

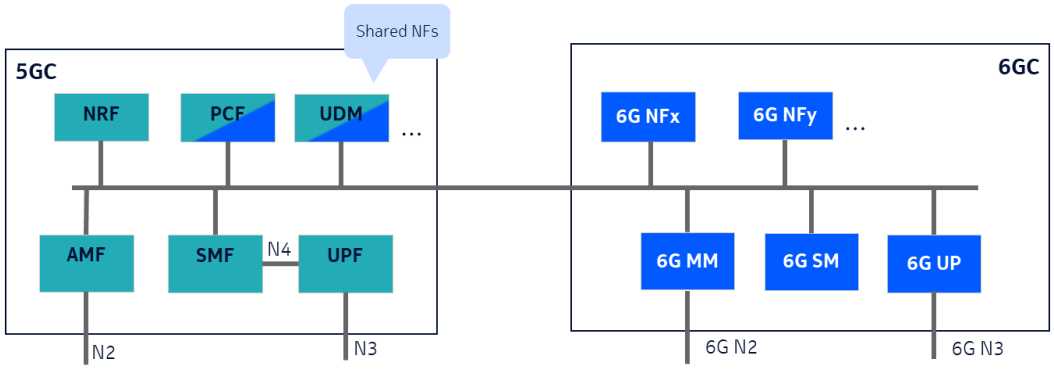


Figure 2-4 6GC architecture (Option 3)

2.2.3 5G-6G Multi-Radio Spectrum Sharing

While traditional static spectrum refarming is, in theory, always an alternative, it might not be justifiable in practice due to the disruption of existing services during the initial stage of 6G device adoption. The real-time spectrum sharing is advantageous as it adapts seamlessly to RAT-specific traffic variations and a gradual 6G uptake. Due to the lean design of 5G, which will also be inherited by and further improved for 6G, spectrum sharing [HEX223-D42] will come with even lower signalling overhead than 4G/5G Dynamic Spectrum Sharing (DSS) [38.300] and it will in addition to Frequency Division Duplex (FDD) bands also support Time Division Duplex (TDD) bands.

When first deployed, 6G cell coverage should at least be on par with today's 4G/5G networks. One option to achieve this is to use an NSA option 1. However, as stated earlier, an NSA option will be complex and hinder the development of 6G services. Another option is to use MRSS between existing 5G bands and 6G. Efficient MRSS [HEX223-D42] will be required not only for coverage bands (lower frequency bands), but also for the 5G TDD capacity bands in FR1. MRSS and (core) architectural aspects can be understood as largely orthogonal issues from a purely technical point of view, as MRSS does not impose restrictions to the architecture options and the MRSS design could support both SA and NSA options.

In other words, MRSS and 6G CA are key technology components for the migration to 6G. Thereby, MRSS shall not require any changes to 5G UEs and 6G UEs shall support MRSS through 6G radio design and basic 6G radio functionalities.

2.2.4 Lower layer split

In 5G the gNB was split into 3 separate logical nodes (Centralized Unity (CU) – Control Plane (CP) (CU-CP), CU – User Plane (UP) and Distributed Unit (DU)), i.e., a higher layer split, with standardized interfaces in between (e.g., F1 between CU and DU, E1 between CU-CP and CU-UP). Those interfaces were assumed to be latency-insensitive and simple and therefore suitable for multi-vendor deployments. However, the consequence of the split between the tightly coupled CU-CP and DU is that neither of the two nodes have all required information to choose an appropriate configuration for the UE. In other words, splitting the control plane between CU-CP and DU increased the complexity, adding additional signalling messages and latency while decreasing the possibility to choose an "overall best" configuration for each UE [HEX23-D53].

To enable a lean, efficient, and fast control plane for 6G, each UE should be controlled by a single RAN control function at any point in time. This will simplify standardization and the implementation of UEs and networks, as well as reduce the number of signalling messages required to acquire and utilize all relevant information for each UE.

Instead of a higher layer split one option is to 6G to allow a lower layer split (LLS), cf. Figure 2-5. This involves separation of baseband and Radio Unit (RU) by a fronthaul interface, typically based on a functional split below the Medium Access Control (MAC) layer (MAC–PHY) or inside the physical layer (PHY, L1).

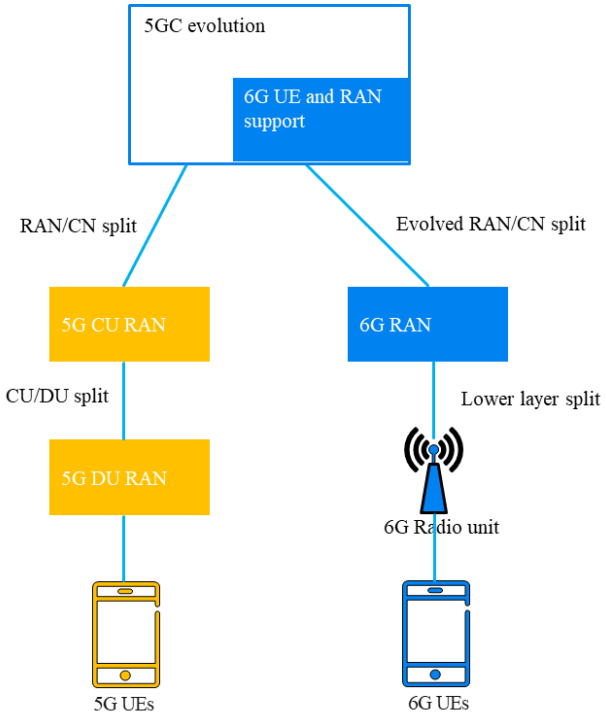


Figure 2-5 Possible 5G to 6G migration path for the Core network. Notice the possible use of Low Layer Split (LLS) in 6G instead of the High Layer Split (HLS) used in 5G.

3 AI enablers for data-driven architecture

In this chapter, Artificial Intelligence (AI) enablers are explored within a data-driven architecture, addressing critical aspects of data-driven decision-making, operational excellence, scalability, innovation, data governance, service delivery optimization, enhanced user experiences [IJR+22] and intent-based automation [NSZ+22]. The integration of architectural means and protocols, MLOps, AI as a Service (AIaaS), and DataOps, and Intent-Based Management creates an efficient technological ecosystem, promoting interoperability and adaptability. AI-driven architectures enhance operational excellence through automation, resource optimization, and agility in response to business needs. Architectural means and protocols provide scalability and flexibility, while AIaaS enables adaptation capabilities. Intent-Based Management ensures alignment with ethical and compliance standards, with Intent-Based Automation and MLOps facilitating adaptability to changing dynamics.

Data-driven architecture is a structural framework designed to support the efficient extraction, transportation, processing, and utilization of data within telecommunication networks or other systems. It serves as the infrastructure that enables various use cases, such as network optimization, customer experience analytics, and service assurance, by providing mechanisms for accessing, handling, and analysing data. Key components of a data-driven architecture include probing and exposure, data pipelines, network analytics modules, and environments for AI and machine learning (ML). This architecture facilitates the implementation of advanced functionalities, including predictive analytics, automated decision-making, and continuous improvement, by leveraging vast amounts of data generated within the network ecosystem [R+20]. On the other hand, the scalability of the architecture meets evolving data demands while addressing risk and regulatory concerns. In this case, security and privacy drive the need for robust measures, with AI automating insights extraction and adapting to changing data patterns [STK+23]. AI enablers like architectural means and protocols, MLOps, AIaaS, DataOps, and Intent-Based Management enhance data-driven architectures, optimizing data flow and aligning systems. In telecommunications, AI together with a data-driven architecture improve network performance, enhance Quality of Service (QoS) through analytics, and bolster security. AI optimizes network traffic, resource allocation, and personalized services, particularly in the 5G era [R+20], reshaping efficiency, and customer service.

Overall, this exploration of AI enablers within data-driven architecture is crucial for navigating the digital era, fostering efficiency, compliance, innovation, and user-centricity. In order to provide further insights, each architectural enabler is mapped to the 6G E2E system blueprint of [HEX223-D22] in Figure 3-1.

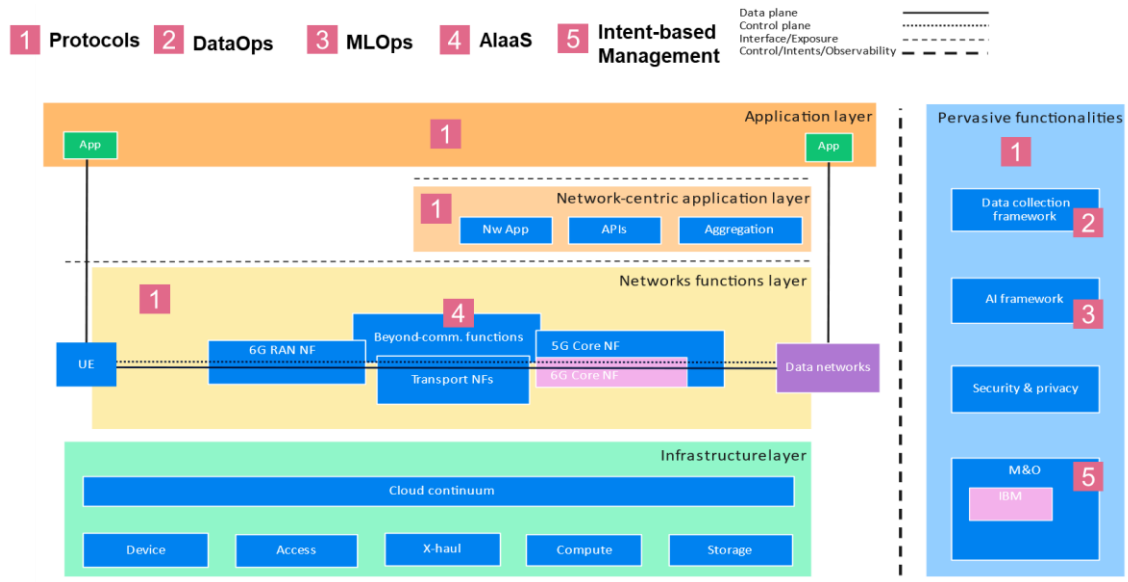


Figure 3-1 AI Enablers–- 6G E2E System Blueprint Mapping

3.1 Data-driven architectural means and protocols

3.1.1 Introduction

In the context of AI enablers within data-driven architecture, architectural means and protocols serve as the foundational framework that supports the integration and functioning of AI. Architectural means refer to the overall structure, design, and principles governing how AI components interact within the broader data-driven infrastructure. This includes defining how data is collected, processed, and disseminated, as well as determining the communication channels between various AI modules and data sources. Protocols, within this context, establish the standardized rules and formats for data exchange and communication, ensuring seamless interoperability between different AI systems and data-driven components. Together, architectural means and protocols create an organized and efficient ecosystem, allowing AI to operate cohesively within the data-driven architecture.

Protocols within the 6G landscape play a pivotal role in facilitating this evolution by laying the groundwork for efficient data transmission and network management. 6G protocols are envisioned to support an array of revolutionary technologies. AI and ML-driven protocols optimize resource usage and prioritize data transmission, maximizing benefits while minimizing energy consumption.

In addition to speed and reliability enhancements, the architectural means, and protocols of 6G also emphasize sustainability and environmental consciousness. Efforts are underway to develop energy-efficient solutions that mitigate the ecological footprint of wireless networks. This involves leveraging advanced power management techniques to minimize energy usage during intensive computational tasks, which includes optimizing algorithms to reduce unnecessary computation. By intertwining sustainability practices into the architectural fabric and protocol frameworks of 6G, the aim is to create an eco-friendly network infrastructure capable of meeting escalating connectivity demands while minimizing its impact on the environment.

The Architectural means and protocols envisioned for 6G not only promise enhanced data speeds but also bring forth a collaborative paradigm where network nodes actively share data and models to achieve common tasks. This collaborative sharing leverages the knowledge and experience embedded in each node, fostering a synergistic approach that can significantly improve network decisions and connectivity procedures. Importantly, the architecture integrates a robust framework for data privacy preservation through common data classification and a tailored approach to selecting data and model sharing methods.

The benefits of this innovative approach extend further into the realm of ML model lifecycle management. The architecture and workflows developed for 6G enable a more efficient and automated ML model lifecycle by considering dependencies between different models and datasets. Moreover, the framework facilitates a

reduction in model training time by strategically reusing pre-trained models and datasets. Additionally, the architecture efficiently manages Digital Twins, playing a crucial role in generating synthetic data for AI/ML model training and augmenting datasets. These Digital Twins are integrated into the training sandbox, ensuring a safe and efficient training environment for models. Overall, the 6G framework stands to revolutionize connectivity and ML management.

3.1.2 Architectural Implications

The transition to cooperative learning in 6G networks brings forth significant architectural implications, necessitating the development of novel control structures, discovery methods, and signalling mechanisms for effective capability sharing among UEs and gNBs. A key consideration is the recognition of diverse privacy sensitivity levels, prompting the implementation of privacy-preserving measures for both data and model sharing. The framework in Section 3.1.2.1 emphasizes that UE data can be split into different privacy levels and shared with the network by preserving UE privacy.

The AI-native architecture proposed in Section 3.1.2.2 integrates AIaaS functions to manage AI/ML models, employing well-defined protocols and means for data exchange between components. Programmable network monitoring and telemetry assess AI model performance, while DataOps efficiently stores and manages collected data. Cooperative learning among cellular nodes introduces challenges, requiring coordination mechanisms in DataOps and MLOps for synchronization. This interconnected framework weaves AIaaS, DataOps, and MLOps, creating a robust privacy-aware, cooperative learning environment.

Key Performance Indicators (KPI) focus on optimizing communication latency, primarily influenced by data transport, and computation latency, which refers to the time taken for actual calculations. This optimization is achieved through innovative control structures designed to enhance efficiency in both aspects. Emphasis on privacy and security includes privacy-aware data classification, aligning UE data sharing with privacy needs. Another critical KPI involves balancing data accuracy and privacy in collaborative learning. A Key Value Indicator (KVI) is the continuous performance monitoring of ML models, considering their foundational role in training other models. The architecture underscores the significance of data quality and freshness, necessitating regular data collection and updates for reliability and data freshness in the cooperative learning environment. In summary, the architectural means and protocols prioritize optimizing latency, ensuring privacy, and maintaining performance and data quality within the cooperative learning framework.

3.1.2.1 Architectural support for cooperative learning

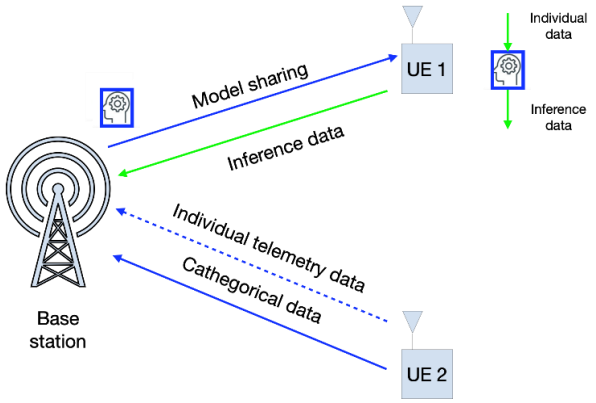
Cooperative learning requires simultaneous participation of multiple nodes and requires multiple iterations. Based on the privacy-aware data classification and trust levels between UEs and network, UE data will be shared either partially or totally or owned locally.

Therefore, different types of data can be characterized by its privacy sensitivity level that determines the data sharing cooperative model:

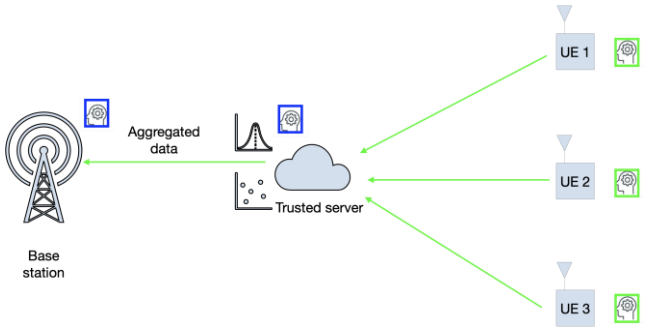
- **Privacy level 0: Individual user private data** can contain precise UE location, exact used application, exact behaviour (e.g., sleeping), personal information (age, sex, etc..). This information is not shared with the network. The network shares with the UE the inference model, trained in a privacy-preserving manner, such that no private data can be revealed by inference. The UE then performs inference on it using private data, and sends the inferred result back to the network, as shown in Figure 3-2a (UE 1).
- **Privacy level 1: Individual telemetry data** capturing statistical information about non-private connectivity activity patterns (QoS categories, traffic prediction) of the UE. The UE might conditionally share this data with the network for inference as depicted in Figure 3-2a (UE 2).
- **Privacy level 2: Categorical data** containing contextual information of the UE (high/low/no traffic) that can be shared as a class/category instead of exact UE information, as shown in in Figure 3-2a (UE 2).
- **Privacy level 3: Aggregated multi-user data.** As illustrated in Figure 3-2b, by using privacy-preserving methods, individual private data can be aggregated across multiple-UEs in such a way it

contains statistical and analytical information about UEs without relieving individual private information.

Moreover, data and models can either be shared periodically, upon request, or when triggered by an event. The data can be attached with a valid time, with the scale ranging from milli/micro-seconds to days/weeks.



a)



b)

Figure 3-2: Cooperative data and model sharing: a) Inferring the individual private data on network-shared model (User Equipment 1 (UE1) and sharing of individual non-private data (UE2); b) Sharing of aggregated multi-user data and models.

3.1.2.2 AI -Native Architecture

The proposed AI-native architecture for 6G networks comprises a set of functions that support MLOps CLs that perform end-to-end AI/ML model lifecycle management for automated data collection and model (re-) training, (re-) deployment, and monitoring while considering different model and data dependencies. The architecture can be employed to provide AI/ML models for different parts of the 6G network (i.e., core, radio, or the deployed services). To this end, the AI-native architecture includes a new repository function which stores dependencies and relationships between different models, and between models and data sources. When performance degradation is observed for a model, the cause and scope of the degradation is identified, and mitigation actions are determined. Then, using those dependencies, the model orchestration framework can identify the models and/or data sources that have direct or indirect dependencies with the degraded model, and perform the appropriate mitigation actions.

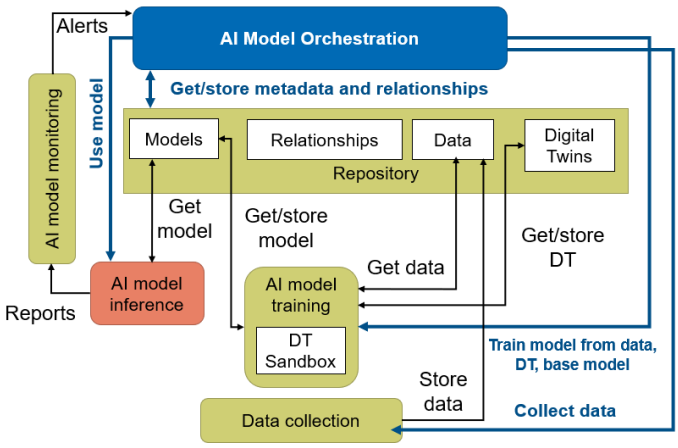


Figure 3-3 AI-Native architecture for efficient AI/ML model orchestration

Figure 3-3 shows the AI-native architecture and components to support MLOps CLs for AI/ML model orchestration:

- **AI/ML model training:** This function fetches data from the repository function and/or the data collection function to create the input dataset, then performs AI/ML model training and validation. Optionally, if the new model training relies on another model (e.g., in case of Transfer Learning or Continual Learning), the function retrieves the base model from the repository. This function may also use a sandbox environment including Digital Twins for safe AI/ML model training and validation.
- **AI/ML model inference agent:** This function performs inference using the trained model retrieved from the model repository. The agent also provides model performance reports to the model monitoring function.
- **AI/ML model monitoring:** This function is responsible for collecting and aggregating different performance reports from the AI/ML model inference agents and determining the model's average accuracy across agents. The function sends alerts to the AI/ML orchestration function when model performance drops below a certain threshold.
- **Data collection function:** This function performs data collection and pre-processing prior to storage. It interfaces with different data sources to collect data on demand, or periodically to maintain data accuracy and freshness (e.g., for keeping Digital Twins up to date).
- **Data and model repository:** This repository serves to store and provide AI/ML models, Digital Twins, datasets and their related metadata, such as accuracy and dependency relationships.
- **AI/ML model orchestration:** This function is the main decision component of the AI-Native framework. Similar to a service orchestrator, it manages the lifecycle of the model and its different versions by collecting monitoring data, making decisions and communicating with the other functions to trigger different operations with the aim of maintaining model performance in an automated manner.

3.1.3 Evaluation

The set of principles outlined for the 6G system in [HEX223-D21] serves as a comprehensive framework that intricately intertwines with the architectural means and protocols. These principles, ranging from support for advanced digital services and full automation to flexibility in various network scenarios, scalability, resilience, and considerations for security, privacy, and sustainability, lay the foundation for an innovative and adaptive architecture. The relationship between these principles and architectural means and protocols becomes evident in the way they shape the fundamental design decisions. Each principle addresses specific aspects of network functionality, service management, security, and environmental impact, providing a holistic approach that influences the protocols, interfaces, and overall architectural structure. This synergy ensures that the resulting 6G architecture not only meets the requirements of advanced communication services but also aligns with principles of automation, flexibility, resilience, and sustainability.

Cooperative learning, with its demand for the classification of data with varying privacy sensitivity levels, represents a paradigm shift in the landscape of wireless communication. Currently, cooperative learning operates primarily on the application level, lacking a tailored network architecture, protocols, and procedures

that cater specifically to stringent requirements of cooperative learning on privacy/security and data accuracy. Section 3.1.2.1 has the following impacts on the design principles:

- **Principle 1, Support and exposure of 6G services and capabilities:** Section 3.1.2.1 supports efficient incorporation of 6G services, such as computational offloading, when network nodes with low capability, having to perform a certain task, need to offload data and/or computation to more capable collaborative node.
- **Principle 6, Persistent security, and privacy:** The proposed collaborative learning model supports a persistent security and privacy principle.

The proposed framework and workflows in Section 3.1.2.2 not only align with the principles but also play a crucial role in automating ML lifecycle management. Moreover, it brings forth efficiencies by allowing early prediction, detection, and prevention of ML model performance degradation, emphasizing the symbiotic relationship between the overarching principles and the intricate details of architectural means and protocols. The study has the following impacts to the principles:

- **Principle 1, Support and exposure for 6G services and capabilities:** The solution developed within Section 3.1.2.2 will enhance support for future 6G services and capabilities by providing and maintaining AI/ML models that can be used by the 6G system (e.g., Zero-Touch network and service orchestration) and the services deployed on it.
- **Principle 2, Full automation and optimization:** The output of Section 3.1.2.2 will support full automation and optimization by providing and maintaining optimized AI/ML models that support 6G system management automation.
- **Principle 4, Network scalability:** The AI-native architecture will support network scalability by training, updating, or re-training AI/ML models and Digital Twins, and deploying new model instances as needed, in a dynamic and automated manner.
- **Principle 5, Resilience and availability:** The trained AI/ML models and Digital Twins can also be used to predict or detect events that might affect the system's availability or performance. The trained AI/ML models and Digital Twins also provide decisions for preventing or mitigating those events, and for the overall optimization of the system and its relevant KPIs.

3.1.4 Summary

The shift to cooperative learning in 6G networks necessitates novel architectural frameworks to enable effective capability sharing among UEs and gNBs. This includes the development of control structures, discovery methods, and signalling mechanisms. Integral to this transition is the implementation of privacy-preserving measures to address varying sensitivity levels. Proposed AI-native architectures integrate AIaaS functions, emphasizing protocols for data exchange and management. Key components such as programmable network monitoring, DataOps, MLOps, and Cooperative Learning governance are crucial for synchronization and performance optimization. Ensuring reliability within the cooperative learning environment demands regular data collection and updates.

Data-driven architectural means and protocols are instrumental in integrating immersive reality technologies [HEX223-D12] seamlessly, optimizing network performance, prioritizing data traffic, and dynamically allocating resources. They ensure security, privacy, and enhance user experiences while enabling predictive maintenance. Additionally, they support autonomous robots [HEX223-D12] by facilitating local ad hoc connectivity, efficient resource allocation, and collaborative decision-making. These protocols also enable the implementation of digital twins [HEX223-D12] across industries, ensuring real-time data ingestion and executing specialized algorithms. In the context of 6G networks, they prioritize privacy protection, establish trusted environments, and enable precision healthcare, ensuring safety and reliability in critical scenarios.

Moreover, the mapping of this enabler to the 6G system blueprint is illustrated in Figure 3-1. Table 3-1 summarizes the main benefits and implications of the data-driven architectural means and protocols enabler.

Table 3-1: Data driven Architectural means and Protocols enabler

Description	The structure, design, and principles governing AI component interaction and data handling.	
Benefits	**KPI improvement**	Interoperability ensures seamless communication among architecture components, integrating diverse systems effectively for efficient data flow. Scalability accommodates data growth and processing demands, allowing expansion without performance loss. API Response Time directly affects user experience and process efficiency, ensuring quick data processing for timely decision-making.
	Design principles [HEX223-D21]	Principles #1, #2, #4 and #5
	Dependencies / Basis for another enabler	MLOps, DataOps, and AIaaS depend on architectural means and protocols for seamless integration and efficient operation. Architectural means provide structure, while protocols establish communication standards, ensuring compatibility and scalability.
Implications	**Requirements**	Requirements for a data-driven architecture within an architectural means and protocols perspective include standardized protocols for integration, scalability, robust security, metadata management, and interoperability. Additional requirements cover version control, monitoring, as well as protocols for cross-functional collaboration, API design, and adaptability to emerging technologies.
	Standard relations & regulations	3GPP TR 23.700-82 [23.700-82], TS 23.288 [23.288], TR 38.817-01 [38.817-01] and TR 38.817-01 [38.817-02]

3.2 MLOps

3.2.1 Introduction

The arrival of 6G is aims to revolutionize connectivity and data processing. This next frontier in wireless communication holds the promise of unparalleled speed, ultra-low latency, and a diverse range of applications. At the core of this transformative shift lies the integration of MLOps within the 6G framework, a strategic union that reshapes how ML is developed, deployed, and managed. MLOps, a comprehensive set of tools, addresses the ML development lifecycle, including data preparation, model training, deployment, and monitoring. The MLOps in the 6G landscape becomes particularly crucial due to the inherent distribution of datasets and the need to train ML models where data is collected, to uphold privacy considerations. This distribution mandates the efficient management of distributed AI functions across the telecommunications system, a challenge that MLOps tackles by minimizing communication, computation, storage, and energy costs during data collection, training, and inference. The collaborative and decentralized nature of MLOps algorithms facilitates model training and inference across various network functions while preserving privacy.

The proposed MLOps solution within the 6G framework brings forth several notable benefits. Firstly, it introduces Input Data Compression (IDC) through a split neural network, which efficiently compresses input datasets with multiple attributes into a reduced dimensionality, reducing communication overhead during the transmission of model outputs or activations. Additionally, the MLOps enables Cross-Network Function Distributed (CNFD) ML model training and inference, allowing the fusion of output from ML models trained on different network domains, such as RAN, core network, or applications, while safeguarding privacy. Furthermore, the architecture facilitates Distributed ML Model Generalization by deploying multiple output neural network models that simultaneously serve different tasks, thus reducing the number of models to maintain and train, along with associated memory and compute requirements. The solution also supports Domain Adaptation, making the generalization node agnostic to different data distributions, further contributing to the reduction of required ML models. Lastly, the integration enables Distributed ML Model

Layer Offloading, allowing for dynamic adjustments in the computation environment by seamlessly offloading certain neural network model layers between computation nodes during training or inference. These benefits collectively preserve user privacy through private Federated Learning (FL) and enable distributed data-driven network decisions by learning from UE, RAN, core network, and application data within the 6G ecosystem.

3.2.2 Architectural Implications

The architecture experiences a paradigm shift as MLOps seamlessly integrates into the existing infrastructure, potentially requiring the introduction of novel network functions rather than modifications. The orchestration of intricate ML models, especially in a distributed setting, becomes a central challenge, demanding adaptability to dynamic changes in compute, memory, and energy capacities. Continuous monitoring and feedback loops are imperative to ensure the freshness of datasets and trained models across a network of distributed nodes. Furthermore, the integration of privacy-preserving architectural components becomes crucial, requiring additional considerations for the management of privacy-sensitive data exchange between nodes. MLOps not only enhances the robustness and efficiency of ML systems but also poses architectural challenges that mandate a holistic and adaptive approach to accommodate the evolving landscape of AI.

In the context of distributed ML settings, establishing communication links between computation nodes is imperative, especially considering the iterative nature of information exchange during multiple rounds of model parameter exchange. This ongoing exchange of information is vital not only during training but also in the collaborative inference phase of split neural networks. Unlike the conventional centralized ML model training, where decentralized nodes acted as data sources, in distributed ML training and inference decentralized compute nodes take on the additional roles of ML model trainers and aggregators. This shift necessitates effective orchestration to dynamically adapt, place, and offload ML tasks based on the changing capacity of the execution environment, considering factors such as compute, memory, energy, and communication. The orchestration process must optimize various KPIs, including model efficacy, training time, and resource utilization.

The integration of privacy-preserving architectural components within the network introduces modifications or novel network functions to facilitate privacy-preserving data collection and learning. This adds complexity to the network architecture, requiring thoughtful consideration of privacy implications and the incorporation of mechanisms to ensure secure and private handling of sensitive data. Additionally, the deployment of intricate ML models in a distributed manner poses challenges, necessitating continuous monitoring and feedback loops to ensure the freshness and reproducibility of diverse datasets. As the network scales with an increasing number of distributed nodes engaged in ML model training, robust infrastructure becomes essential. Integrating MLOps practices further complicates the scenario, demanding compatibility with existing systems and careful management to streamline the deployment and management of distributed ML models. Overall, the implications from the MLOps perspective underscore the need for sophisticated orchestration, privacy preservation measures, and robust infrastructure to navigate the complexities of distributed ML in the evolving network landscape.

MLOps plays a pivotal role, acting as an enabler for both AIaaS and Intent-based Management. Within MLOps, tools and processes are designed and employed to manage the entire lifecycle of AI services, translating user intents into actionable models and services. This emphasizes the significance of MLOps in operationalizing and optimizing AI services throughout their lifecycle. Moreover, the synergy between MLOps and DataOps becomes apparent, forming a crucial dependency to efficiently manage ML models and the diverse datasets used for training. The integration of MLOps and DataOps practices ensures a harmonized approach to handling data and models, contributing to the seamless execution of privacy-aware and collaborative ML initiatives within the network architecture.

KPIs and KVIs play a pivotal role in assessing the effectiveness and efficiency of MLOps within the context of distributed ML, particularly in the split neural network setting. These metrics provide a comprehensive view of the operational aspects and impact of MLOps tools and mechanisms. The identified KPIs and KVIs include robustness, flexibility, communication cost, computation and memory cost, fault tolerance, model security, ML model efficacy, improved privacy/security, data accuracy, power consumption and resource utilization.

3.2.2.1 *Distributed Model Training and Inference*

This section presents a distributed AI-enabled technology called split learning on different scenarios. The research questions addressed in this section are related to:

- *cross-network function ML:* how to perform training and inference on a distributed ML model consisting of input data of different attributes from different network elements such as RAN, core network, or from different network functions, without moving original local datasets from where they are located?
- *model generalization with joint training:* how to train minimal number of ML models with minimal computation and memory requirements that serve different use cases and tasks simultaneously?
- *model generalization with domain adaptation:* how to train an ML model for multiple data domains, i.e., datasets with different distributions, simultaneously? (Note on terminology: the well-known domain adaptation in ML (e.g., data distribution) should not be confused with the same terminology used in communication networks (e.g., RAN, core network), and here we refer to ML terminology).
- *model layer offloading:* how much reduction in CPU utilization and memory allocation be achieved by offloading model layers in between split learning compute nodes?

The distributed ML in this section consists of three main types of nodes: input node, generalization node, and output node. The global Neural Network (NN) model is split into these node types, and every node type contains portion of the global NN model. A node type in this demo is defined as the group of at least one neural network model. The neural network models that are deployed at the input and output of the global neural network are called input and output nodes, respectively. The input node contains the input dataset, i.e., ML features; and the output node contains the target labels. The intermediate NN model is deployed in the generalization node. The nodes communicate in full synchronization. These nodes are illustrated in Figure 3-4.

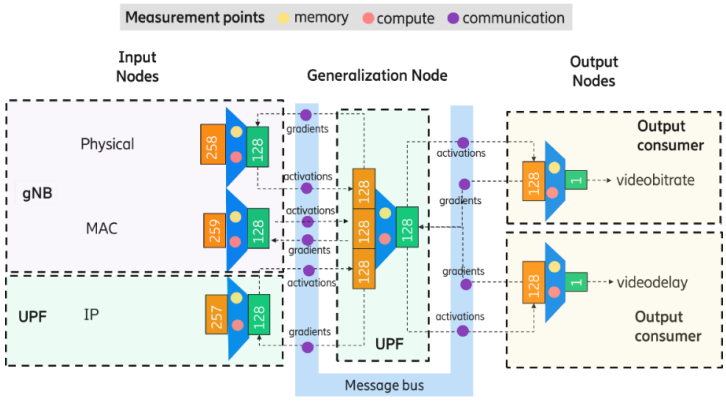


Figure 3-4 Illustration of a split learning setting.

Input node consists of three entities that each contains input data and compute capability. The three input entities are Physical, MAC, and IP, and each entity has one local NN model that trains on the local input data. The generalization node does not have access to raw dataset; it is rather an intermediate entity with a NN model (serving as a compute element) with the goal of taking high compute tasks related to the training of a split learning model. It also serves with the goal of generalizing the intermediate learned representations from input nodes, so called encodings or activations, to multiple use cases deployed in the output node. The output node consists of two NN models serving for two use case tasks, i.e., video bitrate estimation and video delay estimation. The Physical and MAC entities can be co-located in gNB; the IP entity and the generalization node can be co-located in UPF; and the output consumer entities that contains video bitrate and video delay data can be co-located in output consumer such as external application or some other external entity. More detailed description on the proof of concept (PoC #B.2) and results are presented in Chapter 8.1.

3.2.2.2 *Privacy-aware data collection and learning*

Private federated learning and private data aggregation enable privacy-preserving learning on user data. Privacy is preserved by combining the local noise addition with secure aggregation techniques in such a way that correct statistical information of data can be still learned without revealing personally identifiable information.

Being put in the future cellular context, aggregated data originated from several UEs is collected from a defined sampling area and over a defined sampling time window. The data shared with the network does not capture individual private data, but it could consist of statistical (mean, variance, distribution, etc.) and/or analytical insights (embedding, graph, correlations, etc..) of the users in the network. Some examples of such data could be a capture of cellular metrics (e.g., signal strength), Quality of Experience (QoE) metrics, location, motion information, etc.

The granularity of this sampling could be agreed between UE vendor and network vendor such that it respects the user privacy and serves the intended utility. If the network needs private individual data of a UE to infer the best config/parameters, e.g., location for handover, it can share the computation model (inference model) to the UE who can run the model to decide which configuration the NW should use. Such inference models can be trained by private federated learning that preserve user privacy.

Privacy preserving architecture proposed in D3.2 introduced new architectural elements such as UE aggregation unit (performs privacy-preserving data aggregation by leveraging secure aggregation techniques and/or data anonymization to collect statistical information of UE data) and data-driven network control unit, (an entity at the network side that configures the base station by performing automation, optimization, and intelligence services, based on both network and UE data).

This architecture is based on privacy-preserving cryptographic protocols like Prio [CB17], depicted in Figure 3-5, currently being standardized in IETF Privacy Preserving Measurement Workgroup [GPR+23]. It consists of a Collector, an entity that obtains the aggregated UE data, and Aggregators (Leader and Helper), which are responsible for UE data collection. This approach is already efficiently implemented in Apple's ecosystem [TWM+23] [CCC+23].

Leader (main) is an aggregator responsible for coordinating the data aggregation. It receives the encrypted UE data and orchestrates the process of computing the aggregate result as requested by the **Collector. Helper** is an aggregator assisting the Leader with the computation. Privacy is preserved under the assumption that the Leader and the Helper do not collide, i.e., they do not contain the same data shares. The Leader and the Helper send aggregated shares to the Collector that computes the final aggregation result.

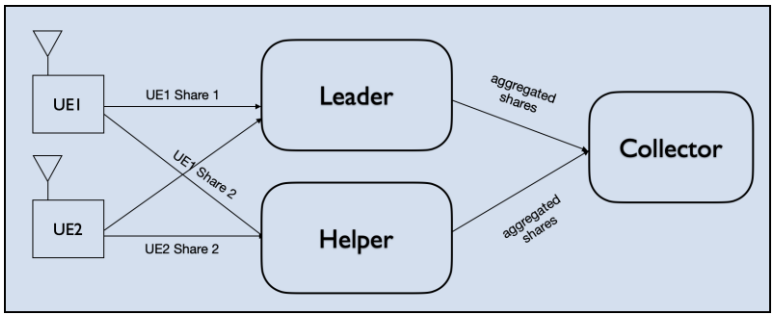


Figure 3-5 Prio-based aggregation system architecture.

The physical realization of the different Prio entities in the cellular network architecture can have different variants. The Leader and Helper can be realized as a Network Data Analytics Function (NWDAF) service or reside in Data Network (DN), while Collector can reside in DN to preserve UE ownership of the data. An example of potential privacy preserving architecture in 6G network is shown in Figure 3-6. The Leader is implemented in NWDAF, thus leveraging mostly CN resources for aggregating encrypted UE data shares, while Helper resides in DN, assisting the Leader. Application function can be in core network (AF1), e.g., for network optimization/OAM, or in data network (AF2), e.g., for analytics.

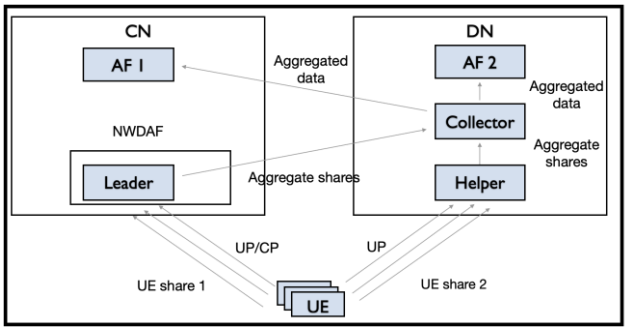


Figure 3-6 An example of Privacy preserving architecture in 6G network.

3.2.2.3 Incentive mechanism design for wireless federated learning networks

Consider a wireless Hierarchical Federated Learning (HFL) network where edge servers facilitate the aggregation and transmission of end user'' model parameters to a server in the cloud. It is evident that the performance of the HFL procedure mainly depends on the end users' discretion to invest their resources (e.g., battery, communication, computing resources), and that is when the problem of providing sufficient incentives arises. Several incentive mechanisms have been proposed so far for the typical FL setting, based on game, auction, contract, and matching theories [TZL+22]. Depending on the objective pursued (e.g., latency requirement, model accuracy, transmission feasibility), different commodities are traded between the FL network parties, such as Central Processing Unit (CPU) resource, power resource, or size of end users' dataset used.

Especially, considering an HFL network, where multiple edge servers exist, then the potential that these servers belong to different providers makes the problem of incentive provisioning even more challenging. Apparently, inherent competition arises between them in how to attract end users that yield uniform data distributions under budget constraints, where the budget may regard resources or monetary rewards offered by the servers to the end users. The term uniform refers to Independent and Identically Distributed (IID) data distribution, which is crucial for maintaining statistical consistency across the edge servers that, in turn, minimizes bias and variance, thereby enhancing the convergence and generalization performance of the federated learning system. The allocation problem can be modelled as a Fisher market [BCD+14], where a set of buyers (i.e., edge servers) aims to purchase multiple goods in the sense of employing end users to maximize its utility under constrained budget availability. In a Fisher market, the prices and allocation of goods are derived to clear the market, meaning that the goods are optimally allocated to maximize the buyer'' utility and fairly priced such that the corresponding sellers would not deviate from the resulting allocation. The latter equilibrium point is referred to as Market Equilibrium (ME).

Motivated by the notably limited literature on the joint client association and incentive provisioning problem in HFL networks, a price-driven client association based on the Fisher market modelling is proposed. The proposed framework manages to model and account for the competitive behaviour of both the edge servers and the end users when performing user-to-edge-server association and pricing, respectively. In this context, the edge servers play the role of the buyers that invest a budget in the form of monetary rewards to motivate end users to participate in the HFL process. The user-to-edge-server association and the pricing are determined based on the ME, at which point the edge servers attain maximum utility from associating with end users that yield balanced data distributions to address the challenging issue of non-IID data. The end users, in turn, are priced fairly so that no other association would be more beneficial for them.

3.2.2.4 Federated learning approach between different city verticals

While FL addresses some issues regarding data privacy by removing the need to transfer raw data to the central server, it does not make it impervious to attacks on the privacy or security of data. Malicious devices on the edge network can not only intercept the model uploaded by the central server and use that information to infer raw data on other device'' uploaded models, but they can also upload data poisonous to the global model aggregation. Some solutions to this are vulnerability detection of edge devices (using Generative Adversarial Networks to protect model parameters, weight-based anomaly detection against poisoning attacks and privacy-preserving data, data filtering in dealing with poisoning attacks), creating a trust system to label the edge

devices on their performance, protecting against backdoor attacks introduced by data points with unusual features, among others.

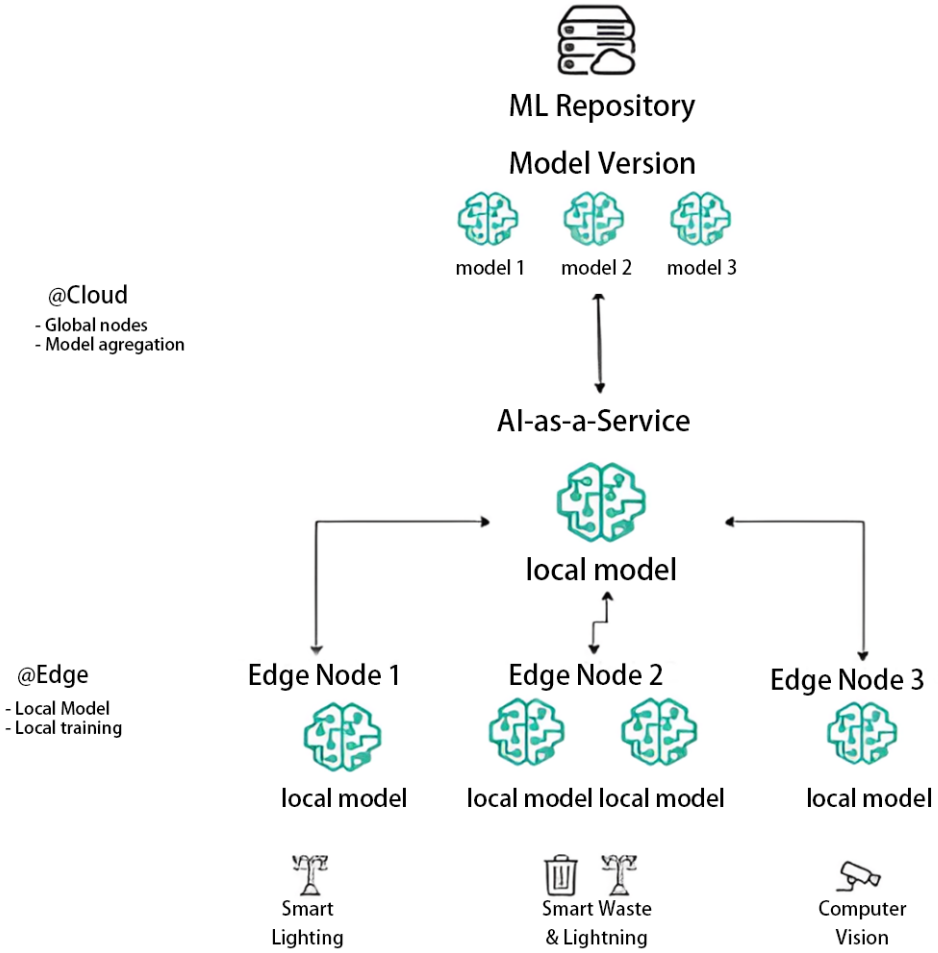


Figure 3-7: Collaborative edge-computing model

In vertical FL, which is our case, the data are complementary; parking and traffic data, for example, are combined to predict someone's route preferences. Finally, in federated transfer learning, a pre-trained foundation model designed to perform one task, like detecting parking availability, is trained on another dataset to do something else, like identify routes with minor traffic.

The Collaborative edge-computing model, see Figure 3-7, is using the edge nodes for local learning, will address the issues of data privacy and resilience. All data concerning city A will remain locally. What is updated on the central node are the algorithms used without all the private information, and therefore without the concerns of what information types and privacy, and without all the amount of data locally present on each edge. These will be managed by a centralised Machine Learning Catalogue, shared by all, and all these communications will be encrypted in transit.

Regarding network requirements needed to ensure communications, Heterogeneous communication systems that can support various innovative wireless technologies, services and applications are required. To do so, the 6G Radio Access Network is deployed standalone. This solution will ensure shared communication between the nodes, enabling the exchange of information in a secure and reliable medium.

All these services must be integrated and standardised to achieve a global service, with model aggregation, directly fed from all the nodes after all the local modelling and training. The solution will present AIaaS, due to this architecture that will provide an edge to the cloud continuum with high availability and resilience.
FL and Edge Computing integration become critical in the context of 6G. The real-time communication needs of FL setups are well-suited to the ultra-low latency and high bandwidth capabilities of 6G networks. Nearer

to the end users, edge devices allow for collaborative model learning and improvement without sacrificing data privacy.

3.2.3 Evaluation

By embracing the principles of full automation and optimization, flexibility to diverse scenarios, and persistent security and privacy, MLOps becomes a strategic enabler. It ensures that AI models seamlessly adapt to dynamic network conditions, address diverse use cases, and uphold stringent security and privacy standards. The relationship between these principles and MLOps reflects a collaborative effort to infuse intelligence and adaptability into the 6G system, harnessing the potential of ML within a framework defined by overarching principles for a robust and responsive wireless communication network.
The distributed nature of ML models, deployable across nodes with computational capabilities, introduces considerations such as the potential need for additional secure hardware. Furthermore, ML model training may necessitate increased bandwidth to share parameters without degrading existing communication link quality. The integration of privacy-preserving architectural elements, driven by MLOps, mandates corresponding changes in architecture and novel signalling protocols to accommodate varied data privacy levels. On the positive side, MLOps significantly enhances the efficiency of the ML lifecycle, streamlining, and automating various processes. This not only accelerates model development, training, and deployment but also underscores the importance of version control and reproducibility, ensuring heightened reliability of ML models.

The mention of a split 6G data architecture reflects adherence to Principle 3, which emphasizes the design's adaptability to various network scenarios, including subnetworks and non-public networks, ensuring optimal performance across diverse environments. The utilization of split neural networks for distributed ML resonates with Principle 6, which highlights the importance of persistent security and privacy. This is achieved by addressing challenges such as data protection regulations and bandwidth limitations, showcasing a commitment to safeguarding user data.

The integration of privacy-aware data collection and a privacy-preserving architecture in support of security and privacy aligns with Principle 10, which underscores the aim of minimizing the environmental footprint and promoting sustainable networks. This principle emphasizes the need for justifying any increased environmental footprint with added value, cost efficiency, and societal benefits. The acknowledgment of distributed data-driven network decisions contributing to sustainable development aligns with the broader goal of environmental consciousness, reinforcing the importance of responsible and eco-friendly technological advancements.

Lastly, the emphasis on collaborative and decentralized model training through MLOps corresponds to the overarching objective of optimizing and automating network operations, aligning with the principles of full automation and optimization (Principle 2). This principle underscores the importance of utilizing distributed AI/ML agents to manage and optimize the system without human interaction, promoting efficiency and autonomy in network and service management operations.

3.2.3.1 Wireless hierarchical federated learning: On the accuracy-energy trade-off

HFL is an extension of FL that introduces a hierarchical structure to the model aggregation process. HFL suggests adding an extra layer of edge model aggregation where edge servers facilitate the aggregation and transmission of end user'' model parameters to a server in the cloud. In this way, better resource utilization across the network can be achieved while reducing backhaul network traffic and thus the required convergence time of the FL procedure and the consumed energy at the end-user devices. Nevertheless, an inconvenient assignment of users to the different edge servers can cause an unequal distribution of data among the edge servers and an imbalanced traffic load on the RAN, bringing exactly opposite effects in the HFL procedure. To reap the maximum benefits of such a hierarchical structure, several research works, e.g., [MAM+22], [LYC+22], [LCW+20], [ZLH23], are devoted to the appropriate user association to the available edge servers and the allocation of the radio resources.

Consider a wireless HFL network as the one considered in Section 3.2.2.3. The users associated with the same edge server transmit their local models over the same time and frequency resources by multiplexing through the power-domain Non-Orthogonal Multiple Access (NOMA) technique. The joint problem of user association

and uplink transmission power allocation is formulated as a non-cooperative game in satisfaction form [PTL+10] to achieve a trade-off between training model accuracy and energy consumption for the users in a distributed manner. The solution concept pursued is the so-called Satisfaction Equilibrium (SE) point, where each end user's minimum acceptable trade-off value is satisfied. The SE point is determined by a Reinforcement Learning (RL) algorithm. The users act as agents that explore their action space, i.e., possible associations and transmission power levels, by observing their achieved trade-off value provided as feedback from the cloud server. The details regarding the system model, problem formulation, algorithm design, and performance evaluation can be found in [CDP23].

The proposed SE solution concept is evaluated by comparison against (i) the "Random States", according to which each user randomly selects its associated edge server and transmission power from a feasible set of values, and (ii) the "Satisfaction Game - Energy" that refers to a satisfaction game where the users aim to solely reach a maximum energy consumption threshold without accounting for the achieved training model accuracy.

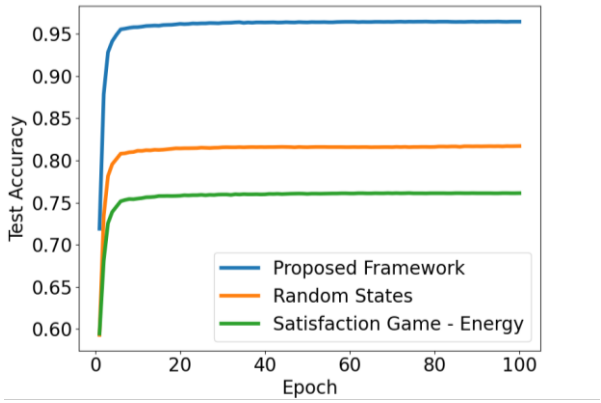


Figure 3-8 Convergence behaviour and test accuracy of the global HFL model

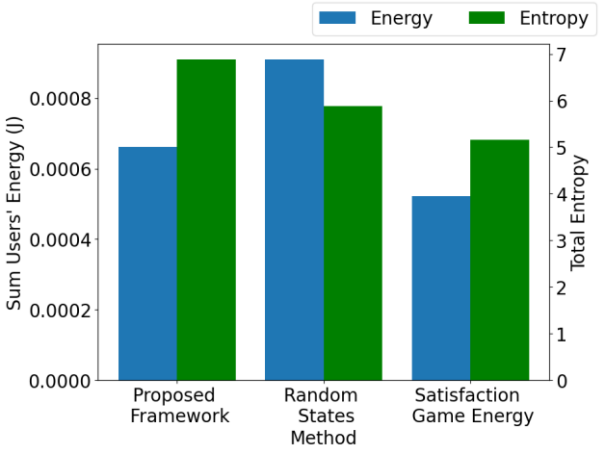


Figure 3-9 Total UE energy consumption and network's entropy

In Figure 3-8 the variation of the global model's accuracy achieved over the test set is presented as a function of the cloud iterations, i.e., epochs, under the proposed and the two comparative approaches. The proposed framework outperforms the two comparative approaches that do not explicitly consider the performance of the learning process striking to achieve a uniform distribution of the data to the different edge servers in the HFL network via proper association. Figure 3-9 depicts the total energy consumed by the users (left axis) along with the total information entropy achieved in the system (right axis) after the convergence of the proposed and the two comparative approaches. The information entropy expresses the users' data distribution among the different edge servers. The highest the total entropy is, the closer the network is to the IID case that results in high training model accuracy. It is shown that the proposed framework provides low energy consumption for the users and the highest information entropy.

Figure 3-10 illustrates the total energy consumed by the users for communication along with the total entropy achieved in the HFL network, under different minimum trade-off values. It is observed that as the pursued trade-off value increases and the user'' requirements get stricter, the total user'' energy decreases, whereas the network's entropy increases. The data are more efficiently distributed to the different edge servers while maintaining interference levels low. Figure 3-11 shows that the presence of more users results in higher energy consumption by each of them due to increased interference levels. On the contrary, the higher number of users provokes an increment in the network's entropy, since the additional users joining the HFL network bring a wider variety of samples, allowing for better distribution of the dataset's classes among the edge servers.

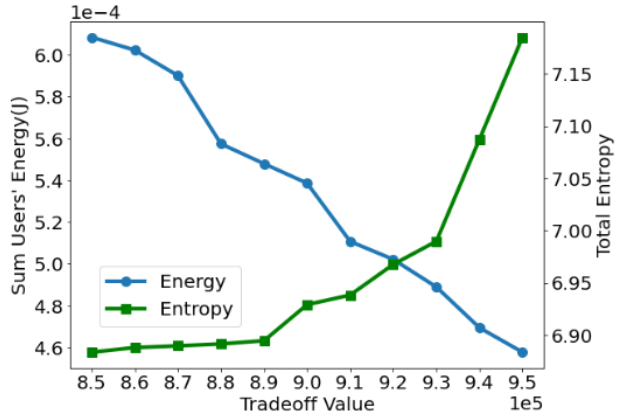


Figure 3-10 Total end users' energy consumption and network's entropy under different trade-off values

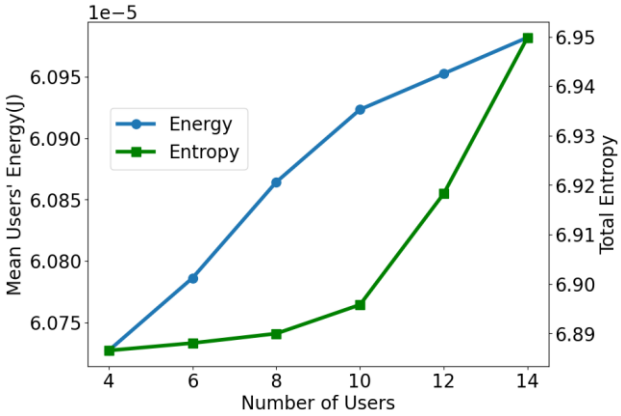


Figure 3-11 Mean end users' energy consumption and network's entropy under different numbers of users

3.2.4 Summary

The integration of MLOps brings transformative changes to system architecture, streamlining the ML lifecycle and posing challenges in orchestration, especially in distributed settings. Continuous monitoring ensures data and model freshness, while privacy-preserving measures become essential for secure data exchange. Effective orchestration optimizes KPIs, with MLOps and DataOps synergizing for streamlined data and model management. KPIs and KVIs provide insights into MLOps effectiveness, emphasizing robustness, flexibility, and security. Overall, MLOps serves as an enabler for AIaaS and Intent-based Management, highlighting its role in operationalizing and optimizing AI services.

MLOps streamlines the development and deployment of applications like Seamless Immersive Reality by efficiently managing data, optimizing machine learning models, and ensuring continuous integration. It supports autonomous robots in local ad hoc networks by managing data, training models, and fostering collaboration for robust operation. Additionally, MLOps facilitates network services for vehicles by optimizing algorithms, ensuring seamless updates, and monitoring performance. It also aids in the creation and operation of digital twins by managing data and ensuring real-time performance monitoring for accuracy and reliability across various applications.

Moreover, the mapping of this enabler to the 6G system blueprint is illustrated in Figure 3-1. Table 3-2 summarizes the main benefits and implications of the data-driven architectural means and protocols enabler.

Table 3-2: MLOps Enabler

Description	Streamline of the entire life cycle of ML models during development, automating deployment through CI/CD pipelines.	
Benefits	**KPI improvement**	In MLOps, key priorities include ensuring model accuracy and effectiveness through continuous monitoring and improvement, maintaining the relevance of models by automating (re)training and deployment with new data, and implementing robust monitoring systems for early issue detection, ensuring model reliability and performance.
	Design principles [HEX223-D21]	Principles # 2, #3, #6, #10
	Dependencies / Basis for another enabler	MLOps is pivotal in integrating DataOps, architectural means, protocols, and AIaaS within the architecture. It ensures smooth deployment and monitoring of ML models, while also establishing production operation protocols. This integration enables seamless deployment and monitoring throughout the data lifecycle. Architectural means and protocols support collaboration and data integration, crucial for MLOps. AIaaS relies on MLOps for

		seamless integration and operation, aligning communication protocols and data exchange formats for scalability and cost-effectiveness.
Implications	**Requirements**	Key requirements for a data-driven architecture within MLOps include high-quality data, scalable storage, communication between computation nodes, data processing, model training and inference, data and model alignment, and data versioning. Automation via CI/CD pipelines and robust monitoring are crucial, as are model explainability and interpretability. Security, compliance, feedback loops for continuous improvement, and resource optimization are also essential.
	Standard relations & regulations	3GPP TR 23.700-82 [23.700-82], TR 38.817 [38.817] for RAN intelligence

3.3 AIaaS

3.3.1 Introduction

6G stands as the forthcoming milestone, promising advancements beyond the capabilities of its predecessor, 5G. 6G technology envisions a future where communication is not just fast and reliable, but also incredibly immersive and responsive.

AIaaS complements the evolutionary trajectory of 6G, introducing a paradigm where AI capabilities are made available and accessible as a service. AIaaS represents a paradigm shift where the mobile network becomes a catalyst for innovative use cases by providing accessible AI capabilities through a variety of APIs, thereby eliminating the need for application developers to build and manage their AI infrastructure. It envisions the mobile network as a provider of pre-built AI models, datasets, algorithms, and tools accessible through user-friendly APIs. This concept goes beyond MLOps by incorporating additional APIs that enhance the network's capabilities, turning it into a platform for innovation, and is considered as a superset of MLOps-aaS, encompassing all MLOps-aaS APIs and introducing additional functionalities.

The synergy between AIaaS and 6G networks yields numerous benefits. Simplifying access through user-friendly APIs, AIaaS seamlessly integrates pre-built AI models, datasets, algorithms, and tools into applications. The highly adaptable APIs provide customization for individual Communications Service Providers (CSP) and across CSP federations, addressing specific customer needs. Beyond convenience, AIaaS tackles data privacy concerns, offering unparalleled services that hyperscale cloud providers cannot replicate, enhancing trust in data security.

Furthermore, the integration of AIaaS into 6G networks expands its capabilities to support unique use cases. Custom AI/ML models are crafted for specific scenarios, ensuring applications powered by these networks are not constrained by generic solutions. Additionally, as AIaaS providers, mobile networks offer distinctive enhancements such as device mobility information, QoS support, and diverse network-related data and analytics.

The focus is set on understanding the services essential to diverse scenarios, the transformative role of AI in enhancing these services, and the imperative for network-wide reconfiguration to meet stringent QoS specifications. With Section 3.3.2.1 the exploration extends into the domain of “interacting and collaborating robots” where the 6G network becomes the catalyst for delivering vital AI services to multiple robots, facilitating human interaction, and empowering drones in tasks such as object detection and obstacle avoidance. The proposed AIaaS framework is not just a theoretical construct; Section 3.3.2.2 integrates both centralized management functions and on-demand per AI service distributed functions to support heterogeneous cloud-native deployments and facilitate decentralized and cooperative AI functions. The existing AIaaS approach [HEX223-D32], emphasizing the need for dedicated AI management and coordination functions for handling different AI lifecycle aspects. The evolved framework includes components such as AI/ML Service Manager, Training Manager, Runtime Manager, AI/ML Catalogue, and AI/ML API Orchestrator, providing a comprehensive solution for managing, deploying, and consuming AI services in cloud-native environments.

3.3.2 Architectural Implications

AIaaS becomes a crucial enabler for advanced capabilities, emphasizing API-driven innovation with user-friendly APIs simplifying integration. This allows for the deployment of pre-built AI models, datasets, and algorithms, offering high customization. The proposed evolved AIaaS framework targets distributed, cooperated, and federated ML techniques, aiming to hide complexity from end-users while managing varied AI service requirements through Runtime and Training Managers. The framework also strives for unified APIs, avoiding unnecessary data models or languages.

The AIaaS approach is connected to DataOps and MLOps, where DataOps ensures efficient, secure, and trusted data management crucial for on-demand functions, while MLOps facilitates the lifecycle management of ML models. Intent-based management synergizes with AIaaS, interpreting high-level intents in the network. The success of AIaaS relies on well-designed architectures and protocols for seamless integration and optimization within the network.

The evaluation of AIaaS in 6G involves KPIs covering latency, throughput, accuracy, scalability, availability, resource utilization, model training time, data privacy compliance, interoperability, adaptability, customer satisfaction, and security metrics. Sustainability considerations are measured through KVIs focusing on environmental, social, and economic dimensions, including energy efficiency, resource utilization, green computing, waste reduction, and digital inclusion. Trustworthiness KVIs assess reliability, ethical considerations, transparency, user feedback, compliance, and robustness to adversarial attacks, shaping overall trustworthiness in 6G networks.

Architectural implications revolve around designing flexible, scalable, and sustainable architectures that prioritize responsiveness, correctness, security, and trustworthiness. These architectures should accommodate distributed, cooperated, and federated ML techniques, provide unified APIs, and integrate seamlessly with DataOps and MLOps for efficient data management and lifecycle support. The success of AIaaS in 6G hinges on well-thought-out architectural means, ensuring optimal integration and performance across diverse dimensions.

3.3.2.1 AIaaS Operation

The transition to 5G Advanced is marked by extensive AI integration within the network for diverse tasks. Looking ahead to 6G, we envision the network evolving into an AI-native platform, termed AIaaS. This concept involves the network serving as an AIaaS provider, offering pre-built AI models, datasets, algorithms, and tools through APIs. AIaaS allows applications to access AI functionalities without constructing and managing their own infrastructure, transforming the network into an innovative platform for various use cases [SDR+23].

Generally, support for MLOps is provided through a set of MLOps tools that can be implemented within the CSP domain as seen in Figure 3-12. The MLOps toolset(s) can be internally exposed to any of the network domains within the CSP domain. Arrow (1) in Figure 3-12 is an example of exposure to the RAN to support the training of AI-enabled network functions. The toolset(s) can also be made available to applications running on top of the CSP network. We refer to the latter as MLOps-aaS.

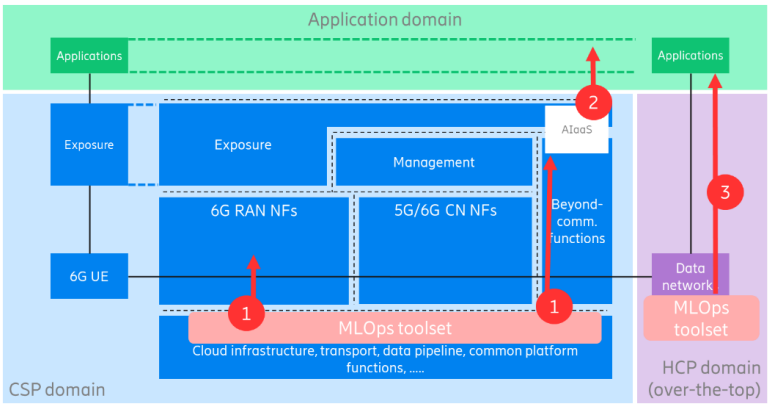


Figure 3-12 The architecture with internal exposure of the MLOps toolset(s) to any of the network domains in arrow (1) and external exposure of AIaaS to applications in arrow (2)

AIaaS APIs are realized by API orchestration of the MLOps toolset (1) and of network functionalities in network domains such as RAN, CN, and management. The AIaaS in Figure 3-12 includes API orchestration but also other functionalities such as authentication and access management. By implementing these functions, higher-level AI services can be realized and exposed to the application domain (2). We emphasize that an application can utilize MLOps-aaS also directly from an HCP — (3) in Figure 3-12.

AI significantly enhances Robots-becoming-cobots by enabling perception, comprehension, and adaptability. ML empowers robots to learn from human feedback, optimizing performance. AI facilitates predictive maintenance by analysing sensor data for issue identification and proactive scheduling.

While the mentioned functionalities can be implemented and embedded as stand-alone capabilities within a robot application, a question arises: How can such robots leverage AI services provided and exposed by the network?

1. AIaaS APIs handle requests beyond existing capabilities, such as creating custom AI/ML models for cobot applications. For instance, a model predicting cobot locations may use GPS, mobility events, and CSP domain sensing. Inference is then consumed through the exposed API.
2. AIaaS APIs fulfil application needs by invoking CSP domain functions, enabling computational offload, sensing capabilities, and resource orchestration. For instance, QoS guarantees may trigger new bearers or allocate edge resources for low-latency connections in cobot applications.

Generalizing from the robot-to-cobots use case we envision a list of API families characterizing AIaaS. The MLOps-aaS API family manages the entire life cycle of ML models, providing tools and environments for training, deployment, and monitoring. The exposure of network services API family includes functionalities like sensing and compute offload. The Exposure of network data API family offers diverse network-related data for applications, such as mobility events, network load levels, performance insights, and energy consumption details. The Life-cycle management of customer models API family oversees customer-specific model lifecycles, including training with network and application data and exposing inferencing APIs. QoS APIs can be combined with other families; for instance, an application with specific latency requirements may package its model in a container (from the network service API family), request compute capability, and specify latency constraints through the QoS API.

3.3.2.2 Strategies and mechanisms for distributed AI and AIaaS functions' management

The identification and definition of common services and functions for AI and AIaaS lifecycle management is crucial to enable an AI-native 6G network architecture and specifically to facilitate AI integration and use. Similarly, it is fundamental to consider heterogeneous cloud-native deployments of AI services and functions across the continuum, while enabling different decentralized and cooperative AI techniques. Starting from this, the AIaaS approach defined in Hexa-X and initial functional split has been revisited to consider AIaaS as a comprehensive framework exposing unified APIs for AI services management and consumption. With the aim of achieving a stand-alone and self-consisting AIaaS framework, dedicated AI (and APIs) management and coordination functions are required to properly handle different AI lifecycle aspects, including training, deployment, inference, etc.

Therefore, as depicted in Figure 3-13, the AIaaS framework functional split is evolved to accommodate two different categories of functions: (logically) centralized management functions (in blue in the figure), and on-demand per AI service (distributed) functions (for serving/inference and training capabilities). In particular, the AI/ML Service Manager wraps and embeds the logics for managing and coordinating the deployment, configuration and execution of on-demand AI training and AI runtime services. The Training Manager is responsible to orchestrate the execution of the various pipelines on top of the cloud continuum, deploying and configuring the on-demand training functions, and managing re-trainings based on the monitoring and validation of ML models performances. The Runtime Manager orchestrates the deployment and configuration of the on-demand runtime functions, taking care of their constraints in terms of execution (e.g., for distributed/cooperative ML techniques). An ML model storage is kept as part of the (logically) centralized management functions to collect, maintain and track the whole set of ML models trained and available within the AIaaS framework. With the aim of exposing AI services towards different type of consumers, the AI/ML Catalogue encapsulate the available ML models with the required metadata for their description (in terms of capabilities, type of model, description of output, etc.), execution and consumption (e.g., with metadata for model invocation and inference execution). At the top of this evolved AIaaS framework, the AI/ML API Orchestrator represents the main entry point, responsible to glue and hide all the internal logics. Its main aim is to expose towards both internal and external consumers a set of APIs which allow to: query and consume available AI services, provision on-demand AI training and runtime services. Specifically, internal consumers refer other functions within the CSP domain, e.g., for network or service management, while external consumers refer to the application domain, according to the terminology presented in Figure 3-12. These exposed APIs are intended to provide access to the various AI services available in the framework, and which depend on the given use cases to be supported, with their constraints in terms of performance, execution and deployment, and models that have been trained and maintained in the AI/ML Catalogue and ML model storage. Indeed, the proposed AIaaS framework is intended to support heterogeneous AI/ML techniques and services (in terms of training, serving, validation), specifically targeting distributed and cooperative ML solutions, with application in different network domains, including RAN, edge, core, transport to achieve a full integration in the 6G compute continuum.

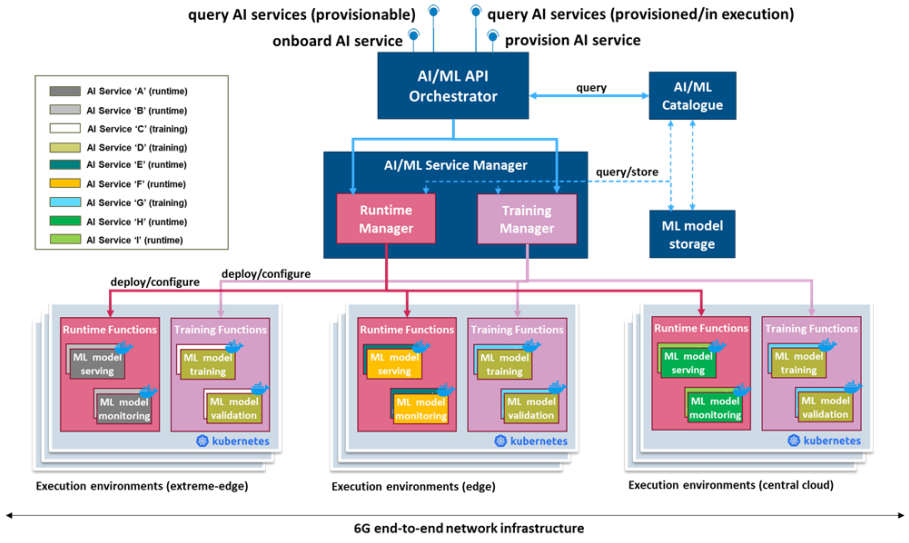


Figure 3-13 AIaaS functional split for distributed AI services

3.3.3 Evaluation

The design principles for 6G architecture lay the groundwork for the seamless incorporation of AI services, leading to a new era of intelligent and adaptive communication networks. These principles are strategically crafted to support the convergence of innovative technologies. They emphasize the support and exposure of 6G services and capabilities, full automation, and optimization. Additionally, they prioritize flexibility in accommodating diverse network scenarios and scalability to handle the demands of AI workloads. They also emphasize resilience and availability for a robust user experience, persistent security, and privacy measures. Furthermore, they focus on cloud-optimized internal interfaces, separation of concerns for efficient network

functions, and simplification compared to previous generations. Lastly, they commit to minimizing environmental impact. The integration of an AIaaS framework within the 6G end-to-end architecture has profound implications for the availability and consumption of AI capabilities. This comprehensive framework, enriched with APIs for advanced exposure of AI capabilities, acts as a fundamental building block, enabling in-network ML functionalities to be readily consumed by various actors. The principles governing this integration are designed to support and seamlessly incorporate diverse AI services into applications, automate ML model lifecycle management, ensure adaptability to different data sources and network scenarios, address scalability challenges associated with AI workloads, and enhance resilience, security, and privacy. The principles also advocate for cloud-optimized internal interfaces, modular separation of concerns, network simplification, and minimizing environmental footprint, thereby creating a robust foundation for the realization of new business propositions, AI and data monetization, and innovative AI-driven vertical services and network applications.

Principle 1–- Support and Exposure of 6G Services and Capabilities:

For AIaaS in the context of 6G systems, it is vital to ensure robust support across AI services, covering model inference, training, and other ML operations. Seamless integration is key, emphasizing the architecture's ability to effortlessly incorporate diverse AI services into applications. This requires a flexible framework accommodating various AI functionalities, enhancing the overall capabilities of the 6G system.

Principle 2–- Full Automation and Optimization:

To enhance this AIaaS principle, it is crucial to explicitly recognize the automation of AI model deployment, scaling, and optimization. The emphasis should be on automated AI model lifecycle management, encompassing versioning, monitoring, and retraining mechanisms. This ensures continuous optimization and updates, aligning with the evolving needs of applications and users.

Principle 3–- Flexibility to Different Network Scenarios:

Expanding this AIaaS principle involves ensuring adaptability to diverse data sources, formats, and AI frameworks, promoting compatibility with a variety of ML models. From the AIaaS perspective, it is crucial to guarantee compatibility with popular frameworks and deploy various models, ensuring flexibility for different applications and use cases.

Principle 4–- Network Scalability:

To enhance this AIaaS principle, scalability considerations must extend to include both AI model inference and training workloads. The AIaaS perspective requires a comprehensive approach to tackle scalability challenges, especially with handling concurrent AI requests and dynamic AI workloads. This ensures the architecture can seamlessly scale to meet the growing demands of AI services.

Principle 8–- Separation of Concerns of Network Functions:

Improving this principle for AIaaS entails extending the separation of concerns to include AI-specific functions such as model serving, monitoring, and retraining. A modular design is crucial from the AIaaS perspective, allowing for the flexible and independent development of various components within the AI model lifecycle.

The benefits arising from the implementation of AIaaS principles are substantial. AIaaS facilitates the efficient utilization of AI capabilities, liberating users from the complexities of managing their own AI infrastructure. Acting as catalysts for innovation, mobile networks, as AIaaS providers, become platforms that foster creativity for various use cases, allowing users to focus on application innovation. The user-friendly APIs simplify the integration of pre-built AI models, datasets, algorithms, and tools into applications, providing customization and flexibility through adaptable APIs to address specific customer needs. AIaaS also addresses data privacy concerns, enhancing trust in data security, and supports specific use cases beyond existing offerings by creating custom AI/ML models tailored to unique scenarios. Additionally, mobile networks, as AIaaS providers, bring network-specific enhancements, offering added value compared to hyperscale cloud providers through features like device mobility information, QoS support, and diverse network-related data and analytics.

3.3.4 Summary

AIaaS emerges as a crucial enabler for advanced capabilities, driven by user-friendly APIs and customizable models. The framework targets distributed ML techniques, managed through Runtime and Training Managers. Integration with DataOps and MLOps ensures efficient data management and lifecycle support. Evaluation metrics encompass various KPIs and KVIs, emphasizing performance, sustainability, and trustworthiness. Architectural implications prioritize flexibility and scalability, aiming for seamless integration and optimal performance in 6G networks.

AIaaS enhances various technological applications. For autonomous robots, it provides capabilities like computer vision, task planning, natural language processing, anomaly detection, and adaptive learning, ensuring effective navigation and task performance in dynamic environments. In the realm of smart transportation, AIaaS improves vehicle capabilities for object detection, navigation assistance, and context-aware communication, enhancing safety and efficiency in urban areas. Additionally, it strengthens digital twin applications by offering data processing, monitoring, predictive maintenance, and simulation, improving decision-making and efficiency across industries. Furthermore, AIaaS supports human-centric 6G services by providing privacy protection, security enhancement, personalized healthcare, and safety monitoring, ensuring compliance with regulations and enhancing trust among users.

Moreover, the mapping of this enabler to the 6G system blueprint is illustrated in Figure 3-1. Table 3-3 summarizes the main benefits and implications of the data-driven architectural means and protocols enabler.

Table 3-3: AIaaS Enabler

Description	AIaaS represents a paradigm shift where the mobile network becomes a catalyst for innovative use cases by providing accessible AI capabilities through a variety of APIs, thereby eliminating the need for application developers to build and manage their AI infrastructure.	
Benefits	**KPI improvement**	From an AIaaS perspective, the most critical KPIs are model efficacy, scalability, and API availability/reliability. These metrics ensure accurate model performance, the ability to handle growing demands, and seamless access to AI services, respectively.
	Design principles [HEX223-D21]	Principles #1, #2, #3, #4 and #8
	Dependencies / Basis for another enabler	DataOps, architectural means and protocols, and MLOps rely on **AIaaS** for seamless integration and operation within the architecture. AIaaS provides access to AI capabilities without the need for infrastructure management, enabling these components to leverage advanced AI functionalities. This integration allows DataOps to streamline data workflows by incorporating AI capabilities for enhanced data processing and analysis. Architectural means and protocols can be designed to seamlessly integrate AIaaS, aligning communication protocols and data exchange formats to accommodate AI-driven processes efficiently. Additionally, MLOps can utilize AIaaS for model deployment, monitoring, and optimization, ensuring the operational efficiency and effectiveness of ML models within the architecture.
Implications	**Requirements**	Key requirements for a data-driven architecture within the MLOps perspective include ensuring high-quality data, implementing scalable storage, and processing, and facilitating data versioning. Automation through CI/CD pipelines and robust monitoring is crucial, along with a focus on model explainability and interpretability. Security measures and compliance with regulations, feedback loops for continuous improvement and resource optimization are essential.
	Standard relations & regulations	Standards: ETSI GS ZSM 012 [ZSM012], 3GPP TR 23.700-82 [23.700-82] Regulations: EU AI Act

3.4 DataOps

3.4.1 Introduction

6G and DataOps represent cutting-edge technologies poised to shape the future of communication and data management. Expected to be fully operational by the end of the decade, 6G could revolutionize industries supporting technologies like advanced AI applications. With potential speeds reaching terabits per second, 6G aims to create a digital landscape that enables immersive experiences and seamless connectivity.

On the other hand, DataOps is a collaborative and agile approach to data management. It emphasizes the streamlined management of data throughout its lifecycle, from acquisition to analysis. By integrating DevOps practices into the data domain, DataOps fosters automation, continuous integration, and collaboration among data professionals. This methodology enhances the speed and accuracy of data analytics, reducing the time it takes to derive insights and ensuring a smooth flow of information across different departments. The convergence of 6G and DataOps holds promise, as the high-speed, low-latency capabilities of 6G can enhance real-time data exchange, offering transformative possibilities for innovation, efficiency, and data-driven decision-making.

The synergy between DataOps and 6G brings along with several benefits. Efficient data utilization is facilitated allowing the aggregation of insights from distributed devices without the necessity of transferring the complete dataset to a central entity, which proves advantageous for 6G networks, designed to manage extensive data, as it significantly reduces network traffic and bandwidth usage. Furthermore, empowering edge devices and end-users to actively participate in the learning process fosters distributed intelligence. This not only promotes a collaborative and decentralized learning environment but also enables real-time decision-making at the edge of the network. Such capabilities align with the evolving needs of advanced technologies like 6G, where the ability to process and act upon data locally is increasingly crucial.

The focus is set on understanding the distributed intelligent management solutions, a federated approach facilitates seamless collaboration between data-producing components and systems within a management framework. These components deliver either cleansed data or locally trained AI/ML models to a centralized management entity through standardized interfaces. This approach prioritizes efficient data sharing and model training while safeguarding sensitive information and enables secure data sharing only when necessary or at specified intervals, maintaining the integrity of sensitive information.

3.4.2 Architectural Implications

The correct functioning of AI is related to data quality. Data shall be delivered, pre-processed, and stored where and when required. This imposes requirements on a flexible data ingestion architecture, which can refer to DataOps. DataOps serves as a pivotal force orchestrating efficiency and collaboration. Its central role involves attention to data quality assurance, employing processes to cleanse and validate data, ensuring the precision essential for effective AI model training. DataOps aligns the efforts fostering seamless communication and a shared vision within AI initiatives. Through adept orchestration of automation, continuous integration, and monitoring, DataOps accelerates the AI development lifecycle while incorporating scalability, flexibility, and robust security and compliance measures.

From a DataOps perspective, in Section 3.4.2.1 critical KPIs and KVIs guide the assessment and optimization. Model Convergence measures the efficiency of the learning process, ensuring models reach stable and accurate states. Network Efficiency evaluates data transfer effectiveness, addressing latency, congestion, and bandwidth usage. Model Inference assesses real-time decision-making speed and accuracy. Data Ingestion metrics gauge the efficiency of incorporating diverse data sources. User Engagement and Satisfaction KPIs provide insights into end-user experiences. Model Reusability ensures adaptability across contexts, maximizing the utility of ML models. These metrics collectively drive continuous improvement within the DataOps framework, creating an efficient and responsive data ecosystem.

Additionally, the capability to leverage diverse data sources is demonstrated in Section 3.4.2.1. This facilitates the development of personalized ML models tailored to individual preferences, contributing to enhanced user experiences, targeted services, and improved decision-making across diverse network domains. The scope of

ML applications is expanded, allowing its application to various industries without the necessity of sharing data with operators. This not only ensures data privacy but also unlocks the potential for new, data-driven insights and decision-making capabilities within specific verticals and among application owners. The principles of efficient and secure data management in DataOps environments are aligned with by these features mentioned in Section 3.4.2.1.

3.4.2.1 E2E 6G Network Slice Instance Employing Distrusted Intelligence Solutions

In distributed intelligent management solutions, data-producing components and systems within a management framework seamlessly deliver either cleansed data or trained local AI/ML models to a centralized management entity via standardized interfaces. This federated approach ensures efficient data sharing and model training while protecting sensitive information.

The centralized entity, as depicted in Figure 3-14, plays a critical role in aggregating individual local models or training cleansed data to construct a comprehensive global model for a specific optimization function. This global model leverages the collective learnings from local models and cleansed data, enhancing the overall performance of the management framework.

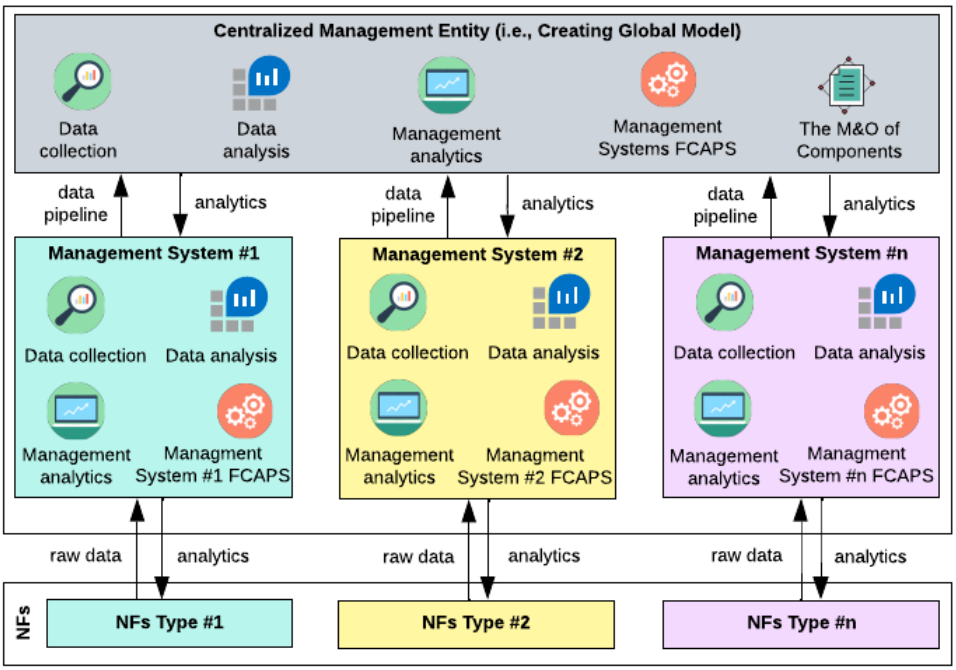


Figure 3-14 Enabling Efficient Data Sharing and Model Training Across Management Systems

Local models, also referred to as domain-level models in this article, are trained within specific management domains using the data or network data stored on those domains. This approach maintains data privacy and security by keeping confidential information within the respective management domains, allowing for secure data sharing only when necessary or at specified intervals.

The global model, trained on cleansed data or trained local models stored in the central repository of the centralized management entity, encapsulates the collective knowledge from all management systems. It functions as a coordinator, periodically aggregating updates from local models and incorporating them into its own parameters, ensuring continuous improvement and optimization across the management framework.

3.4.3 Evaluation

From a DataOps perspective, a comprehensive approach in Section 3.4.2.1 is highlighted, showcasing the leveraging of diverse data sources, the facilitation of personalized machine learning models, and the enabling of efficient data utilization. DataOps principles are adhered to, emphasizing the aggregation of insights from distributed devices to streamline data operations. The empowerment of edge devices and end-users in the

learning process resonates with the decentralized nature of DataOps, fostering distributed intelligence and facilitating real-time decision-making at the network's edge.

In the context of 6G, the approach in Section 3.4.2.1 stands out for its potential to handle vast amounts of data, resulting in the reduction of network traffic and bandwidth usage. The capability to apply the approach across a wide range of applications and industries without the need to share data with operators aligns with the diverse and dynamic requirements of 6G networks. The integration of standardized AI/ML approaches, such as the 3GP''s Management Data Analytics System (MDAS), underscores compatibility with 6G technologies, offering a pathway for intelligence incorporation into various management functions across different domains.

3.4.4 Summary

Effective AI relies on high-quality data, driving the need for a flexible data ingestion architecture like DataOps. DataOps ensures data quality through processes like cleansing and validation, crucial for AI model training. Automation and continuous monitoring accelerate the AI development lifecycle while emphasizing scalability, flexibility, and robust security. Key KPIs and KVIs within DataOps focus on model convergence, network efficiency, model inference, data ingestion, user engagement, satisfaction, and model reusability. Leveraging diverse data sources enables personalized ML models, enhancing user experiences and decision-making while ensuring data privacy. These principles align with efficient and secure data management in DataOps environments, fostering an efficient data ecosystem.

DataOps enhances diverse technological applications. For Seamless Immersive Reality, it efficiently manages data integration, ensuring quality and deploying machine learning models for immersive experiences. In autonomous robots, DataOps integrates sensor data, processes it in real-time, and manages machine learning models, improving efficiency and safety. Similarly, in vehicle networks, DataOps optimizes performance, fosters communication, and ensures security, enabling smart transportation. Moreover, for Digital Twin use cases, DataOps integrates and analyses data for improved decision-making and optimization. Lastly, in human-centric 6G services, DataOps ensures efficient data management, real-time analytics for personalized healthcare, and privacy protection, enhancing reliability and scalability.

Moreover, the mapping of this enabler to the 6G system blueprint is illustrated in Figure 3-1. Table 3-4 summarizes the main benefits and implications of the data-driven architectural means and protocols enabler.

Table 3-4: DataOps Enabler

Description	DataOps ensures data quality by cleansing and validating data, crucial for precise AI model training. It fosters collaboration and efficiency accelerating development through automation, integration, and monitoring. Additionally, it ensures scalability, flexibility, and robust security measures.	
Benefits	**KPI improvement**	From a DataOps perspective, the most critical KPIs are accuracy rate, automation rate, and data drift detection rate. These metrics ensure the reliability of data processing, efficiency of operations, and proactive identification of changes in data patterns, respectively.
	Design principles [HEX223-D21]	Principles #1, #2, #3, #4, #5, #6, #7 and #8
	Dependencies / Basis for another enabler	AIaaS, architectural means and protocols, and MLOps depend on DataOps for efficient data management and collaboration. Architectural means and protocols can integrate DataOps seamlessly, ensuring compatibility and interoperability. MLOps relies on DataOps for efficient data preparation, enabling smooth deployment and monitoring of ML models. AIaaS benefits from DataOps by leveraging its streamlined data workflows and efficient processes
Implications	**Requirements**	Key requirements for a data-driven architecture within the DataOps perspective include implementing robust data quality management, establishing automated end-to-end data pipelines, and implementing version control for data artifacts.

4 Network modularisation

A modular 6G design is introduced in [HEX223-D32] where the architectural aspects for 5G evolution as well as its implications have been discussed. In this deliverable, the four fundamental enabler blocks of the modular 6G design (i.e., presented in [HEX223-D32]) are aggregated into two main aspects of network modularization, namely *6G network modularization* and *E2E service design in modular 6G*. "6G network modularization" details a methodological approach on module design, where the module can incorporate different 5G NFs or micro-services that are allocated within the same network container. In other words, the modularization study can be considered as the design and analysis of new 6G NFs. Whereas "E2E service design in modular 6G" focuses on how the modules with various degrees of granularity and composition are utilized for various deployment options and scenarios to achieve a high degree of operational flexibility and efficiency, e.g., improved latency, throughput, better sustainability via scalability etc. Figure 4-1 maps these two enablers to the 6G E2E system blueprint that has been proposed in [HEX223-D22]. Network modularisation enablers present fundamental changes to the way the 6G network functions and their interfaces are designed and the E2E interactions of these network functions, i.e., represented with the solid red line on Figure 4-1. Moreover, the envisioned 6G pervasive functionalities can impact as well as be impacted by the enablers of this section, i.e., represented with the dotted red line on Figure 4-1. In the following subsections, the details of architectural changes on the network functions and pervasive functionalities layers and their preliminary evaluations are presented. At the end of this section, the summary sections detail the benefits and the implications of the architectural changes.

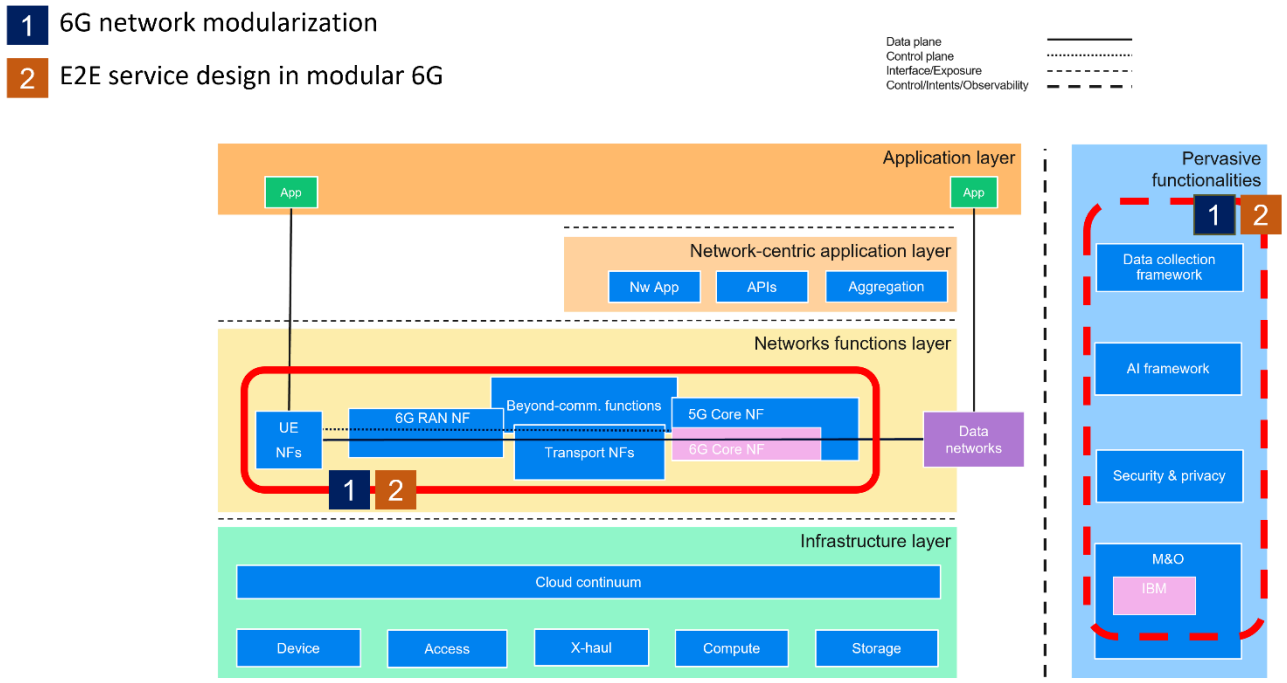


Figure 4-1 Mapping of the network modularisation enablers to the 6G E2E system blueprint of [HEX223-D22].

4.1 6G Network modularisation

4.1.1 Introduction

A multitude of services and use cases have been envisioned for 6G [HEX223-D12] that encompasses various requirements and deployment scenarios. However, to achieve the KPIs posed by these services, the network architecture (i.e., the NF design, the relevant interfaces and procedures) needs to be optimized per service needs, which requires a highly flexible and efficient network design. Flexibility and efficiency have also been the fundamental principles in the previous technologies. For example, with 4G networks the idea of separating the control signalling and the user data streams, i.e., Control and User Plane Separation (CUPS), was introduced. However, the low granularity of CP network functions (NFs) was a major limitation on the flexibility. To solve this limitation, the 5GC is built upon an SBA, where the CP NFs have a higher degree of

functional decomposition. Through the interactions of these NFs, it is possible to execute relatively complex 5GC procedures. This highly granular structure of the 5G core ensures an increased flexibility and scalability. However, 5G SBA also increased the inter-NF dependencies. For example, to perform a UE registration procedure more than six network function needs to interact with each other [23.502]. Consequently, the current design can lead to increased signalling cost due to the increased number of new modules that need to be interacted to execute a procedure which would also impact the latencies. The modularization as a candidate solution is revisiting the functional composition of the NFs and optimize them for specific deployment options or procedures. Although a module can be built using 5G NFs, it is also possible to decompose a 5G NF and create more than one module from this NF.

As the research community's focus shifts towards 6G, the modular network architecture and how to design such modular NFs become a major point. The design of a module (or modular NF) depends on not only the service based KPIs but also the network provider's KVIs as well as the deployment options. To this end, a better balance between NF granularity and the number of required interactions between the system elements is required, so that NFs can be added, updated, and replaced in a flexible and modular manner. The idea is to minimize the number of NFs involved in the different procedures by aggregating common ones into new network modules. Thus, it will be possible to reduce today's 5G functional dependencies that create long sequential procedures with several variants.

In this chapter, the 6G network modularization enabler and its implications are described. First the impacts of different module granularities are investigated. As summarized above, a high granularity of the control plane would bring both advantages and disadvantages. To understand the potential of network modularization, a clear understanding of the different granularity options and their implications on the E2E performance is needed. A module can contain different network functions or functionalities (e.g., microservices), and there are different ways to determine how to create a network module (cf., Figure 4-2). In this enabler, different decomposition options and their performances are presented. Finally, the high degree of flexibility is achieved via the interactions between different NFs in 5GC. In 6G, taking advantage of the findings of 5G, the inter-NF or inter-module integration needs to be streamlined, i.e., by optimizing where possible, removing unnecessary interactions, and preserving the already optimum ones. In the last part of this enabler, these streamlined interactions are presented.

4.1.2 Module design and composition

The current design of the 5GC enhances flexibility and agility in introducing new functions to the core network. In this architecture, individual NFs, such as the Access and Mobility Management Function (AMF) and the Session Management Function (SMF), are characterized by specific logic for execution, with each NF generating services i.e., consumable by others. The core network facilitates basic procedures like UE registration, UE deregistration, and PDU session establishment by defining interactions and information exchange among these NFs [23.502]. However, the collaborative execution of procedures by different NFs leads to an increased volume of signalling traffic as they exchange messages and information elements. Furthermore, the Procedure Completion Time (PCT), representing the time required for a procedure's full execution, is prolonged due to inter-NF communication among the involved NFs [GSH+22]. Exploring alternative designs for 6G involves considering new control plane core network functions to minimize inter-NF signalling and reduce PCT in the system.

The placement of 5GC NFs is typically static and depends on RAN network topology and mobile network connectivity to other networks. The core NFs placement optimization goal is to minimise network reaction time, the overall signalling traffic and network cost. Users' locations and user generated traffic depend on time-of-day, e.g., sport event can cause a serious local traffic growth. To manage signalling congestion and to increase 5GC programmability, the CP NFs have relatively high degree of functional decomposition in 5G SBA. An important feature of 5GC is the ability to add new CP NFs that can interact with other NFs to create more advanced CP services. 5GC CP NFs can be implemented in the cloud which gives various benefits. Firstly, the scaling mechanism of the cloud can allocate resources to a virtualised NF if needed, for example to handle more requests by an NF. Secondly, virtualised NFs can be migrated or cloned during network runtime. However, the use of stateful NFs can make such processes complex. An important performance implication of SBA in 5GC can be the delay in transferring and processing CP messages. The transfer delay comprises a component related to the physical distance between NFs, a delay related to a message bus and a

delay related to virtualisation (typically not negligible), whereas the processing delay can be controlled by allocating resources to a NF using scaling. In particular, NF can be scaled in response to signalling traffic growth to avoid NF overload.

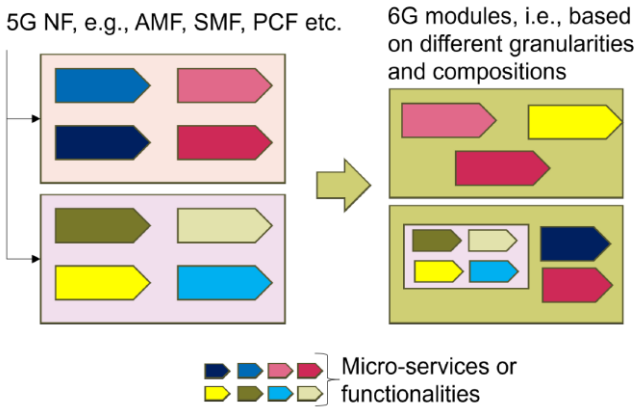


Figure 4-2 Decomposition of 5G NFs with changing granularity

In addition to the delay, there are multiple implications of the NF granularity and the module creation, i.e., outlined in Table 4-1. Therefore, to evaluate the different architecture options and procedures, a multitude of design criteria must be considered that affect the KPIs of the system [AKE+24], including but not limited to

- the affinity constraints, e.g., the nodes are meant to be placed in different physical locations;
- latency limitations, which concern how a procedure is constructed based on the current architecture;
- reliability requirements, which manifest themselves in affinity and anti-affinity rules, persistent memory, etc.;
- new technologies, e.g., cloud, virtualization, IP protocols, and quantum technologies;
- unique requirements posed by the use cases such as Joint Communication and Sensing;
- signalling traffic volume;
- self-sufficiency, i.e., indicates how many times a certain function depends on another function to complete a task;
- backward compatibility;
- the number of failure points that indicates how many times a functional entity requires a re-start of a procedure resulting from a failure to send/receive a message;
- service (including CP/Management Plane services) reaction time (delay) and other factors understood as service KPIs;
- service-CP/CP/MP overhead (control/management traffic volume minimization);
- data type and characteristics, i.e., dealt within procedure execution which would impact how many and what type of data processing mechanisms/libraries are needed to be deployed in a node;
- easiness of functions migration and cloning without disturbing other services.

Customised techniques can be used to optimise the virtualisation-related delay, e.g., by the selection of a suitable inter-NF message passing mechanism.

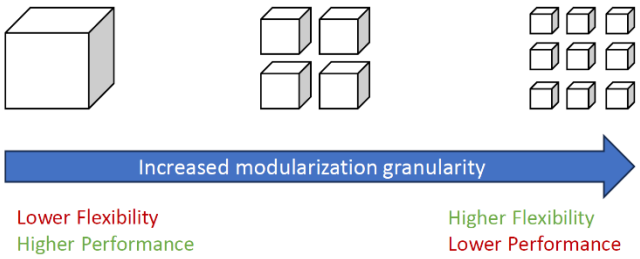


Figure 4-3 Impact of modularization granularity on performance

Generally, there is a trade-off between NF granularity and the overhead of inter-NF interactions, as illustrated in Figure 4-3. Low granularity modules reduce the number of inter-NF interactions but long paths to an NF increase response time, while highly granular NFs can be placed closer to the traffic source or sink to reduce response time, but the number of inter-NF interactions increases. When the approach defines more granular modules, the execution time of procedures will be shorter due to the reduction of messages round-trip time. High granularity of NFs should be combined with identification of services that a chain of such NFs can handle. Such a service can be linked with mobility management, security or a user.

Table 4-1 Advantages and drawbacks of fine-grained modularisation

Advantages	Disadvantages
Scalability management: efficient and targeted scale-up and scale-down of the resources assigned to the fine-grained modules.	Performance: the latency increases when traversing more modules to complete the NF execution.
Maintenance: easier to identify, isolate, and replace faulty fine-grained modules.	Interfaces definition: interfaces between modules should be defined for cross-vendor deployments.
Development: faster when developing different independent modules in parallel. Note that this may come at the cost of increased complexity in terms of integration and testing.	Management: managing a bigger number of modules and their interactions increases the management overhead.
Deployment: flexible in placing modules in distributed deployments.	Signalling: more signalling and data exchange is needed between fine-grained modules.
Reusability: efficient when reusing fine-grained modules in different implementations.	Context data management: more memory transactions are needed by the modules to read and store stateful data.

Different granularity level of a module raises different potentials and drawbacks that need to be considered based on the specific use cases as well as the overall KPIs. Following the granularity of a module, the second criteria is to choose how to create the module. In this deliverable, two major modularization methods are considered, namely *Procedure based decomposition* and *Dynamic decomposition and placement.*

Procedure based decomposition: The primary objective of this design is to disrupt the strong inter-dependencies between NFs by introducing Procedure-based NFs (PbNF). Unlike the current 5G architecture where the logic needed to execute a full procedure such as UE registration or UE deregistration is distributed among different NFs, the procedure-based decomposition methodology consolidates the logic for executing a complete procedure into a single PbNF that has a higher modular granularity compared to the current 5G Core NFs. For example, one PbNF is the UE registration NF which is made up of the processing logic that is distributed in the following 5G NFs: AMF, Authentication Service Function (AUSF), PCF, NRF, UDM, and User Date Repository (UDR). The design and implementation of this methodology is described in the following and is based on [GSH+22].

The implementation of the PbNFs is done by reusing the source code of the stateless Free5GC v3.0.6 simulator [Free5GC]. For instance, the source code of the NF handling the Registration procedure (Registration NF (RegNF)) includes logic that, in an SBA system, is distributed among AMF, AUSF, PCF, and UDM. Consequently, the processing logic as well as the services involved in this procedure execution is consolidated

into one RegNF as shown in Figure 4-4. Since NRF, UDR, and Unstructured Data Storage Function (UDSF) serve as service-discovery mechanisms or database abstraction layers whose implementation depends on the operator's backend database, the NFs are not integrated into the new PbNFs.

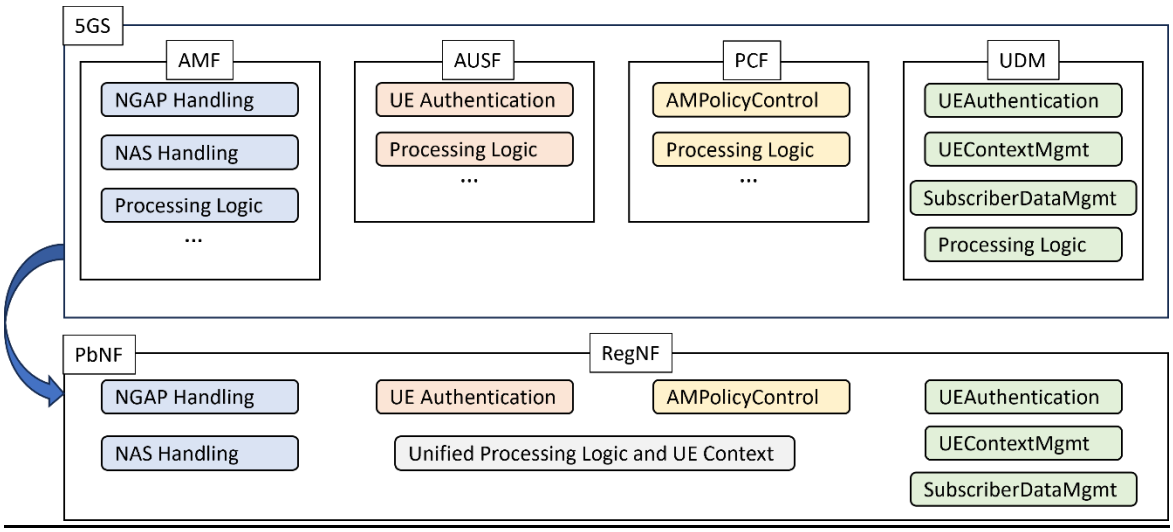


Figure 4-4 Integrating processing blocks from different NFs into UE Registration Procedure-based NF

During the implementation, the AMF is used as the basic NF whose source code is expanded by incorporating additional processing blocks code to realize the PbNFs. Additionally, a consolidated UE context is constructed to encompass all information that was previously distributed across various NFs. Special consideration is given to identifying any redundant data to prevent unnecessary duplication and overhead. Given the diverse procedures handled by different NFs in proposed system, it becomes imperative to implement at least procedural statelessness. Consequently, upon completion of a procedure, a PbNF serializes the UE context and state, transmitting it to the UDSF, which maintains the most up-to-date versions. The local context is then purged to avoid undue memory resource consumption. Subsequently, when a subsequent procedure is initiated, the responsible PbNF retrieves the information from UDSF before proceeding to fulfil the request.

Utilizing the previously outlined methodology, UE RegNF, PDU Session Establishment NF, PDU Session Release NF, and UE Deregistration NF PbNFs are implemented to perform the tasks associated with UE Registration, PDU Session Establishment, PDU Session Release, and UE Deregistration procedures, respectively. Figure 4-5 shows a 5G or 6G system based on the proposed PbNF. The PbNFs could be deployed in edge locations to mimic the distributed deployments of AMF. It still interacts with the other unmodified 5G NFs such as UDR, NRF, UDSF, and UPF. The RAN&UE Emulator Node depicted in Figure 4-5 is implemented as a gNB and UE Emulator application. The purpose of developing this application is to be able to generate scalable control plane traffic and evaluate the performance of the updated 5G system that incorporates PbNFs. Furthermore, the Emulator Node includes tools to collect and report different performance metrics such as PCT.

In future work, this design will be evaluated and compared against the current 5G system as a baseline while considering different quantitative and qualitative metrics. These include the PCT, the volume of signalling as well as the impact on deployment flexibility, duplicated services and interface terminations, and redundant implementation and testing.

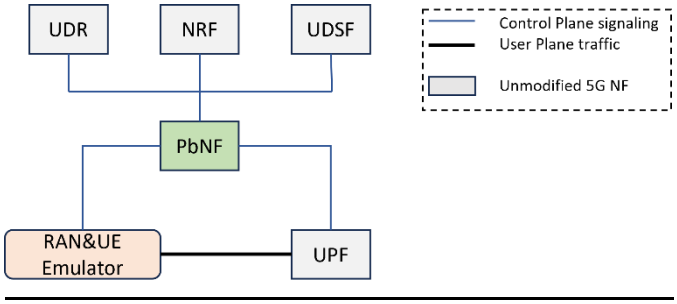


Figure 4-5 5GS architecture with the introduced Procedure-based NF (highlighted in green) interacting with the unchanged 5G NFs and the RAN & UE emulator (highlighted in orange)

Dynamic decomposition and placement: The Cloud Continuum allows flexible orchestration and placement that breaks the current split of a system into different orchestration domains. In this option a concept which exploits Cloud Continuum for dynamic optimization of deployment of NFs in order to optimize KPIs of deployed services is presented. Using Cloud Continuum, for example, (i) virtualised CN functions can be placed in UEs or RAN hosts, (ii) deploying some functions in the UEs facilitates the deployment of Distributed Mobility Management (DMM) and source routing (e.g., SRV6) – mechanisms that significantly reduce CP congestion probability.

The decomposition of CP into services provides the opportunity for a service-centric approach (i.e., the proactive management considering service context), in contrast to the function-centric approach (reactive management to optimize the function operation) i.e., used in previous generations of mobile networks. In the proposed approach, it is assumed that a certain service of CP uses several service-specific NFs (i.e., NF chain) and all created services are loosely coupled. The NFs, using Cloud Continuum, are initially placed in the data centres selected according to best practices. The initial configuration is evaluated in terms of service reaction time and interactions between NFs belonging to a chain. If within the Cloud Continuum a better placement will be found for an NF, one of the following actions can be taken:

- **migration of NF(s).** In this approach, a new placement of an NF reduces reaction time and service traffic volume, although assuring continuity of NF operation or minimizing the interruption time should be considered;
- **cloning of NF(s).** The service chain analysis may lead to identifying NF clusters with intensive signalling within a cluster. In the original configuration, the identified clusters are handled centrally (for example); however, the node that is central can be cloned according to cluster analysis results. After creating clones of some functions, more operations can be handled by cloned functions. Such handling contributes to quick reaction time and traffic volume reduction; however, the exchange of information between cloned NFs has to be considered;
- **on-the-fly creation of a 'macro-NF'** out of highly granular NFs to minimise delay and traffic of inter- NF interactions. Such delay, even in a single data centre, can be significant, and the SBA communication can also add non-negligible delay. Creating such a macro function, however, takes time, and the possibility of decomposing it back into highly granular NFs must be allowed. The operation allows for creating a service composed of NFs of different granularity.

The presented approach is assumed to use self-managed, highly granular NFs interacting via SBA. To implement it:

- a function that continuously evaluates the deployed service KPIs (e.g., reaction time, traffic volume) has to be added to each deployed service;
- a dedicated service orchestrator that, on the basis of the analysis of the service graph and its properties, takes decisions on NF migration, NF cloning or NF-macro creation. For that purpose, a dedicated set of cooperating AI algorithms is needed.

As the optimised services have to interact with each other, the placement of a function that acts as an inter-service gateway should also be optimised. During the service's or set of services' lifetime, the dynamicity of NFs placement change requests can be observed in the long run, and short-lived placement requests can be ignored according to operator policy. The process involved to define each configuration is costly and the cost should be evaluated for each service reconfiguration.

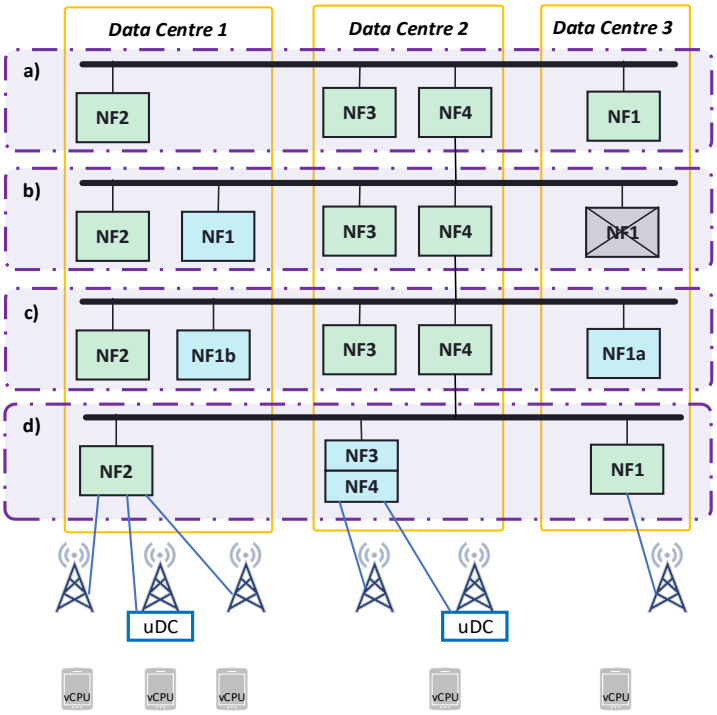


Figure 4-6 Example of possible independent optimisation options of a Control Plane Service.

Figure 4-6 illustrates possible independent optimisation options of a Control Service applied for RAN infrastructure deployed in three data centres: a) Initial placement of NFs; b) NF1 migration; c) NF1 cloning; d) Creating of macro-NF by compiling together NF3 and NF4. NF4 acts as a gateway between services. The orchestrator and service monitoring modules are not shown in the figure.

The main idea presented in the above subsection is a dynamic placement of highly granular functions on top a distributed, virtualised infrastructure. The approach benefits include a reduction of network traffic volume and data or signalling delay. The runtime optimisation may allow dynamic migration of functions in a distributed environment. The highly granular approach to 6GC problems includes delays related to interactions between multiple NFs. The sources of delay in virtualised networks with their typical values are presented in Table 4-2.

Table 4-2 Sources of delay of inter-NF interactions in a virtualized environment

Type of delay		Symbol	Typical value	Source of information	Remarks
Distance related delay		Td	3.3 - 6 µs/km	[DFM+21]	One way (for calculation 6 µs/km is used)
Virtualisation stack delay		Tv	50 µs -100 µs	[CBC+21]	The delay depends on virtualisation technology and data centre load
Message broker (SBA)	RabbitMQ	Tmb	0.6 - 1 ms	[SBA]	handshake
	Kafka	Tmb	3 -5 ms	[SBA]	handshake

The table shows that with the use of RabbitMQ as a message broker of CP, a typical handshake execution time between two NFs will take Te= 2 x Tv + Tmb + Td *2*D, where D is a distance in kilometres between the NFs. In the most optimistic case, it equals 0.7 ms + Td*2*D; in the average case, the delay is 1.2 ms + D*2*Td, and in the pessimistic case, 5.2 ms + D*2*Td. For a 10 km ‘duct’ distance between NFs, the delay will be 0.82 ms, 1.32 ms and 5.32 ms, respectively. For NFs ‘duct’ distance equal to 50 km, the Te value will be 1.8 – 5.8 ms. As can be seen, the distance between the NFs is not a critical factor in the overall delay.

The UP of 5GC traffic path typically can include at least one UPF-I and UPF-PSA [23.502]. If UPF is virtualised, the one-way delay in case of 50 km is 1-2 ms, which is a maximum value for low latency services. In such a case, the delay also includes the load-dependent queuing delay. Adding to a path, a virtual UPF adds a delay in the range of 0.3 ms. In the case of UP delay, the distance is an essential factor.

For evaluating the CP procedure, the execution time of the UE triggered Service Request (SR) procedure of 5G NAS signalling has been taken into account [23.502]. According to [MHH+22], such a procedure in 4G networks is the most intensive (about 45% of the CP traffic); moreover, it requires a short execution time. The Message Sequence Chart in this case shows, among others, intensive interactions between SMF and involved UPFs ($I\text{-}UPF_{old}$, $I\text{-}UPF_{new}$ and UPF_{PSA}) 12-18 messages are exchanged and depending on the scenario and at least 4 messages are exchanged between SMF and AMF [23.502]. For the distance of 10 km between UPFs, SMF and AMF (simple case), the mentioned message exchange (only part of the whole procedure) will take 10-14 ms on average, and in the case where the distance between the mentioned nodes is equal to 50 km, the delay messages exchange time will take 16 – 22 ms. Please note that the analysis concerns only part of the overall procedure, focusing on the nodes with the most frequent interactions and the processing time by nodes involved in the procedure is ignored.

The above-presented analysis shows that interactions between highly granular NFs, in the case of CP, can be a source of a significant delay, which is mainly linked with virtualisation and message broker – a low granularity provides, in such case, evident gain. On the other hand, a highly granular CP and UP comes with the possibility of quick local interactions and a reduction in the UP volume. However, it seems that without redesigning UP and the transport-related control plane (SMF), the potential of the analysed concept cannot be fully unleashed – the interactions-related delay is too high. The 5GC NFs cannot be decomposed into highly granular functions without changing the approach to data transport and using new signalling concepts. To that end, the replacement or modification of GTP by other UP concepts, for example, a combination of SRv6 [RFC8402] and P4 [P4], and a kind of speculative, proactive signalling, can significantly reduce CP message exchange related to UP connection setup and modifications.

4.1.3 E2E module interfaces and interaction

While cloud-native 5GSBA is mainly centralized in cloud datacentres with a single administrative domain, 6G is expected to become more of a fully distributed system across the entire compute continuum. This raises the need for flexible service routing capabilities that allow the best possible utilization of the distributed resources. In this section, we present the foundations for a data-centric interaction, modelled as a dataflow, of serverless highly granular functions as the redesign of current control-plane core network concepts and procedures. Currently, there are two new paradigms which can be used for this purpose: Data-centric networking (DCN) and Data Flow Programming (DFP).

Data-centric networking is a network design concept which argues that routing can be made more efficient if communication is based directly on application-specific data content instead of the traditional IP-style addressing. DCN appears as a more suitable networking model, instead of today's host centric model, to better fulfil the requirements of an Edge-Native 6G networks' SBA. By combining data-centric and dataflow concepts, 6G networks can evolve into dynamic chains of serverless, loosely coupled, and stateless highly granular NFs, dynamically orchestrated to achieve an optimal balance between consumed and available resources across the continuum.

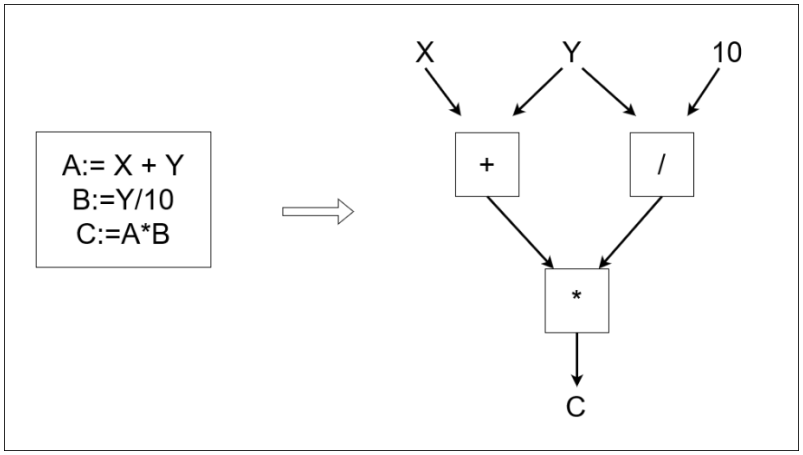


Figure 4-7: A simple logic and its data flow equivalent

DFP is a computational model that represents application as a directed graph, where each node of the application graph, called an *operator*, is a computational unit, and the directed arcs between them are unbounded First-In-First-Out (FIFO) streams, called *links* as shown in Figure 4-7. Operators are executed concurrently and triggered following different strategies, *e.g.*, all inputs (i.e., links that flow towards a node) need to be present, or only a subset of them. Naturally, this computational model has been widely adopted for network workloads. It is leveraged to provide serverless, fast, and cost-effective unified stream and batch data processing, paving the way for fully managed data processing services that are easily provisioned, managed, and scaled. Moreover, it eases the decomposition model of services into microservices, which provides several advantages in terms of not only resiliency, scalability, and composability but also flexibility, maintainability, performance, and security.

Although the evolution for a data-centric SBA and higher decomposition of core NFs are already underway, they are being tackled independently. This section explores dataflow-based service composability as an enabler for the seamless integration of these isolated approaches, thus promoting innovation and customization to meet the requirements of emerging applications and services while providing simpler and more convenient approaches for programmers, operators, and service providers. The principles behind the proposed Data-centric service-based architecture for Edge-Native 6G Network are presented in this section in the form of a system architecture, as depicted in Figure 4-8.

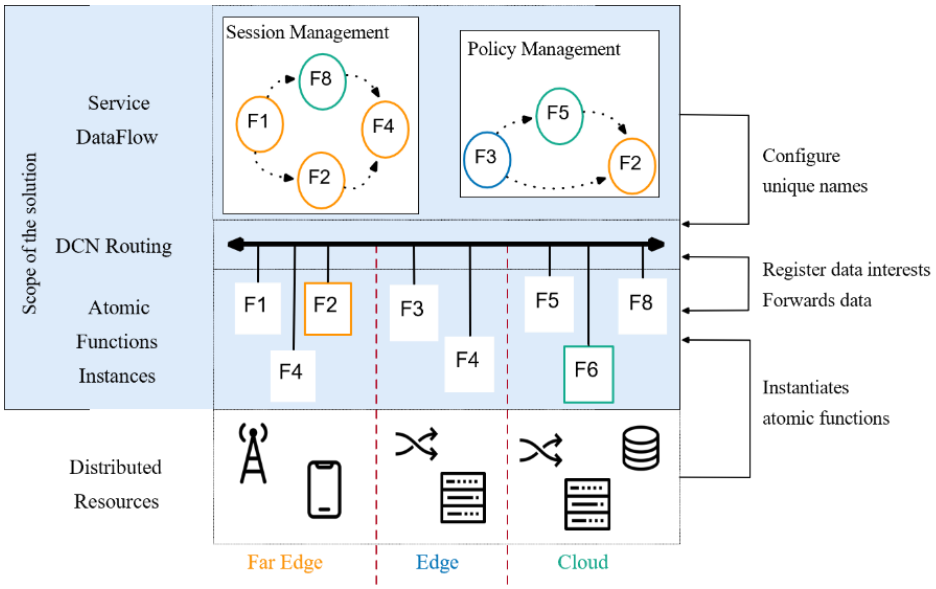


Figure 4-8 Data-centric Service-Based Architecture for Edge-Native 6G Network

The proposed solution in this section is based on a layered architecture that is hierarchical in nature, with each horizontal layer building upon each other. Each layer provides an abstraction that ease the development and management of the overall architecture, paving the way for a system that is flexible, scalable, and able to handle the complex demands of Edge-Native 6G Networks. It is worth mentioning that the solution proposed in this section is compliant with the design principles of the Hexa-X-II E2E blueprint focusing on the network's functions layer.

1. *Service Dataflow:* The service dataflow approach for service composability facilitates the definition of more complex services (hereinafter referred to as Service Functions) by combining and reusing highly granular components into service chains on demand. Moreover, since highly granular functions can be serverless and stateless, it allows parallel execution where function instances are executed concurrently as long as their data dependencies are satisfied. By relying on a dataflow programming model, services are represented as a directed graph, stitching and chaining together different highly granular components in a similar way to the Network Function Virtualization (NFV) framework. To do so, this layer creates and manages the interconnection between highly granular components across the continuum, which are dynamically established and isolated upon the instantiation of the Service Function. Such interconnections are configured in the underlying Data-Centric Service Routing in order to create a distributed location-transparent communication bus. The service dataflow automatically generates and configures unique names and mapping between such names and Service Function inputs and outputs. This layer is also responsible for the lifecycle management of not only the entire Service Functions but also each highly granular component. It includes tasks like bin-packing, orchestration, and scheduling, thus determining the most resource-efficient way to place highly granular components across the continuum, as well as instantiation, termination, and monitoring.

2. *Data-centric Service Routing:* Enabling the communication between dynamic, possible volatile, highly granular functions as well as the optimal selection of redundant functions pose a major challenge to the service routing layer. It is widely recognized that the Internet architecture is sub-optimal for addressing this sort of challenge, and novel paradigms such as ICN have been emerging and introduced at different levels. While the adoption of these technologies as a full replacement for the underlying Internet Architecture has proven to be harsh, its application in contained environments to address specific challenges has been increasingly studied. By relying on a data-centric, naturally distributed, approach, the realization of the communication bus for service routing offers location-transparent primitives and allows data to be directly addressed without specifying its origin. In other words, data is retrieved and routed solely based on its name, allowing highly granular functions to be developed without the complexity typically associated with endpoint configuration. Such capability becomes essential to support the reuse and/or sharing of highly granular functions across different *Service Dataflows*. Also, a Data-centric Service Routing is expected to provide increased performance and efficiency, better scalability, and more robustness as well as reduce network overhead and levels of indirection. This layer receives name-data tuples from the Service Dataflow layer and based on its configuration, forwards them to interested highly granular function instances. Furthermore, it automatically establishes links between datacentric entities, either routers or applications.

3. *Highly Granular Function Instances:* The dynamic and opportunistic realization of highly granular functions, on top of a set of resources distributed across the continuum, would be at the basis of the stringent performance indicators expected to be fulfilled by the next generations of mobile systems. By splitting NFs into fine-grained building blocks, highly granular functions can be designed and implemented as serverless and stateless instances, guaranteeing parallelization of operations, non-conflicting access, and control over system resources. Still, this layer raises challenges at different levels, including the decomposition of high-end services into serverless highly granular functions, their integration on top of heterogeneous multistakeholder distributed resources, and the optimal allocation and scheduling of resources to provide the required assurance at the service level. These issues remain a challenge, being consistently identified in 6G research [URB+21]. Interconnections between the instances, as already mentioned, are defined by the Service Dataflow layer, which abstracts the Datacentric service routing, by providing a DFP-based API to the highly granular functions. Such abstraction takes care of declaring the interests from an instance towards the data-centric service routing, thus enabling each highly granular function to receive and send data.

4. *Distributed Resources:* The resource layer comprises any available physical or virtualized resource spanning across the continuum. Resources are typically organized in a hierarchical and integrated computing infrastructure extending across multiple tiers comprising clouds and central data centres, edge data centres in

the middle, and far-edge computing devices that are available locally in the access area. Each tier of this distributed infrastructure differs in essential properties (e.g., network topology, computing power, volatility, mobility, and how they span across different administrative domains) which is perceived as an adoption barrier. Thus, the management of such heterogeneous distributed infrastructure requires a degree of abstraction and the necessary tools for streamlined and flexible management that conceals the underlying complexity. Cloud computing relies on huge, well-interconnected data centres, based on the latest computing and networking fabrics, providing virtually unlimited resources. The Edge is composed of small data centres based on Commercial Off-The-Shelf (COTS) hardware with metropolitan rings or enterprise network topologies. When reaching the Far-Edge, usually on-premises infrastructures (e.g., under a Public Network Integrated Non-Public Network architecture - PNI-NPN), the resources become sparse, less powerful, and, eventually, volatile.

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 data-centric primitives and intrinsic multicast capabilities of the proposed SBA, as well as the atomicity of functions and their composability as dataflow graphs, contribute to more efficient signalling, both from the perspective of faster procedures and lower message overhead. Additionally, it enables parallel signalling surpassing today's sequential signalling, while eliminating the service indirection of discovery processes. Moreover, it allows bundling more functionality together (e.g., for serving a given context), thus reducing the overhead that crosses the network.

2) Streamline Refactoring: the location-transparency and the data-oriented primitives of the SBA contribute to shipping software updates more efficiently and effectively, especially with the increased softwarization of distributed nature being considered for 6G networks. The capability to update and/or refactor existing functions, paves the way for rapid innovation cycles by identifying and quickly addressing ways to improve processes, yielding rapid results and, eventually, contribute to a reduction in the Total Cost of Ownership (TCO).

3) Serverless, Stateless, Reusable, and Shareable functions: by considering highly granular functions and their composition as a dataflow graph over data-centric primitives, the proposed solution leverages stateless functions that can be shared, and transparently replicated across the continuum in a serverless and dynamic fashion on top of likely volatile resources. Such simpler components are instantiated among different stakeholders (e.g., Internet Service Providers (ISP), cloud operators, and verticals), facilitating and creating a whole new set of opportunities for the realization of PNI-NPN concepts. Moreover, the intrinsic characteristics of the underlying datacentric communication bus (DCN Routing in Figure 4-8) enable seamless reusability of highly granular components by multiple service flow realizations (with the proper trade-off between isolation and reutilization).

4) E2E Orchestration over the Continuum: the Service Dataflow introduces a new approach for the orchestration of 6G networks, where the management of various heterogeneous software modules is performed in a unified manner while offering finely-grained services across the continuum. Despite this, the orchestration shall provide interfaces for interconnecting multiple administrative domains, promoting the sharing of components and providing a virtually interconnected continuum from private networks at the Far-Edge up to the Cloud.

Security and Fault-Tolerance: security and fault tolerance are taken in a multi-layer approach, each tackling different aspects of the architecture. The Service Dataflow layer provides security and fault-tolerance at the content level, thus providing both properties from an E2E perspective. The data-centric Service Routing layer focuses on securing the link, thus providing hop-to-hop security, leveraging both TLS and mutual TLS. At the highly granular function layer, application specific (e.g., authentication and access control) security is applied. Nevertheless, this approach allows each layer to focus on a different aspect of security, contributing to the onion security model, in which the system security is achieved by layering different layers of secure components.

A proof-of-concept (PoC) of the proposed architecture was implemented to validate its design and assess its impact. The PoC focuses on the data-centric principles of the proposed architecture, namely the *Data-centric Service Routing* and *Service Dataflow*.

Zenoh and Zenoh-Flow were selected to implement the proposed approach of a data-centric SBA for edge-native 6G networks. Both present a unique set of capabilities that make them a perfect fit for the *Data-centric Service Routing* layer and *Service Dataflow*, respectively. Zenoh /zeno/ is a pub/sub/query protocol unifying data in motion, data at rest, and computations. Moreover, it is fully decentralized and provides enhanced capabilities, like the dynamic discovery of nodes, data priority, and low network overhead. Such capabilities have already proven its suitability as communication protocol between edge applications [Sab21]. Zenoh-Flow is a dataflow programming framework that leverages data centricity and location transparency, thus allowing for applications to span across the continuum. It provides a declarative dataflow graph definition, seamless operation across the continuum (including automatic deployment), fosters code reuse by the means of composite operators, and achieves high performance with low latency and higher throughput than state-of-the-art solutions such as Istio, HTTP and gRPC. Zenoh-Flow functionalities are already being leveraged as a dataflow layer for network intelligence algorithms in the context of automated network management [GCF+22]. Open5GS an open-source implementation for 5G Core and EPC (Release 17 at the time of writing), was deployed, along with emulated gNBs and User Equipment (UE). This setup was used to obtain traces of the selected 5G workflows, which were later used to replay the behaviour of the NFs, their reference points, interactions, and exchanged data. Our experiments include not only the proposed solution, but also HTTP, gRPC, Istio, MQTT, and Kafka solutions due to their adoption in today's 5G systems. The validation has been carried over our internal testbed composed of 3 machines (AMD Ryzen 7 5800X @ 4.0 GHz, 32GB of DDR4 3200MHz RAM, running Ubuntu 20.04 LTS), directly interconnected via 100GbE fiber-based NICs over a ring topology.

The 5G workflow selected for evaluation purposes is the *PDU Session Establishment* [29.502]. This workflow follows a Request/Reply pattern. For this workflow, the proposed solution is compared against (i) HTTP, as the protocol defined in the 3GPP for the 5G SBA communication, (ii) gRPC, a widely used protocol for SBA-based interactions, and (iii) Istio, a widely used open-source service-mesh implementation. Results are shown in Figure 4-9 , depicting the total time required to complete the entire PDU session establishment exchange.

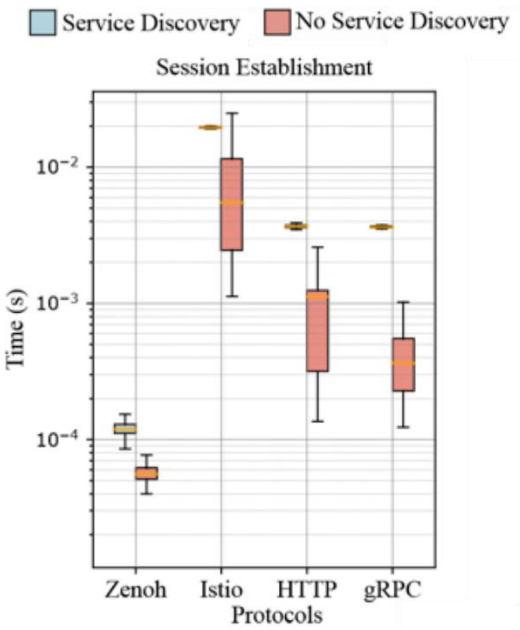


Figure 4-9 Workflow completion time for different protocols. (lower is better).

The proposed solution performs one or more orders of magnitude faster than HTTP, gRPC, and Istio, regardless of service discovery being performed. HTTP and gRPC show similar performance. The reasons lie on (i) a more efficient data exchange; (ii) faster serialization; and (iii) less wire overhead, thus less transmission time. On the other hand, for the service-mesh implementation, the reason lies in how the service mesh is implemented, consisting of a chain of proxies to provide location transparency, thus increasing overall latency.

4.1.4 Summary

The envisioned extreme KPIs targets as well as the efficiency and flexibility goals of the 6G system shift the research communities focus towards revisiting the 5G network functions and creating new modular structures, i.e., optimized towards particular aspects. This, however, raises the fundamental question of how the modularization should be performed, e.g., with which granularity or which methodology and how should, if any, the E2E interfaces and interactions be evolving. This enabler aims at answering these questions by stating the advantages and disadvantages of various options and presenting some preliminary results.

Through modularization, the network function composition can be designed to meet specific requirements i.e., set by the service or the operator. Consequently, it is possible to revisit the design to meet extreme requirements e.g., less than 1ms latencies i.e., set by collaborative robots and digital twins. Moreover, in addition to the measurable goals such as QoS metrics, it can also optimize the energy consumption and complexity. For the use cases that requires a high degree of flexibility, e.g., immersive experience and fully connected world, the modules designed in a highly granular way. The work in this section explains the different strategies to perform this modularization, the relevant evolution of the E2E module interface and interaction design and the possible implications to the E2E performance.

As a fundamental methodology, the network modularization can impact all the other enablers, i.e., including but not limited to Joint Communications and Sensing (JCAS) protocols and AIaaS. Consequently, it would have implications to all the 3GPP standards, including but not limited to [23.501], [23.502] and [38.413].

Table 4-3 Network modularisation enabler summary

Description	This enabler focuses on different granularities of a module, how to design a module and the evolution of the modules based on different deployment locations and use cases. This enabler also investigates the streamlining the interactions between different modules/domains	
Benefits	**KPI improvement**	Note that the major argument of this enabler is the capability to optimize the 6G NFs or modules according to certain KPIs. Therefore, it would decrease latency (E2E), procedure completion time, CP signalling, and it will reduce complexity. It will also increase efficiency via direct signalling and less interfaces while enhancing network scalability, flexibility (i.e., both deployment and execution), and reliability. It will provide uniform and reliable service for all users
	Design principles [HEX223-D21	As a fundamental enabler that has direct impact on the NF design and 6G architecture, it has impact on all the design principles provided by [HEX223-D21], with particular impact on Flexibility to different network scenarios (#3) by flexible composition of NFs/modules, Network scalability (#4), Resilience and availability(#5), Persistent security and privacy (#6), Internal interfaces are cloud optimized (#7), Separation of concerns of network functions (#8), Network simplification in comparison to previous generations (#9) through decreased number of interfaces and interaction to execute a procedure.
	Dependencies / Basis for another enabler	This enabler is a fundamental enabler that does not have any dependency to any other enablers. However, as it analyses and proposes new NF compositions and architecture, it can serve as a basis for all other enablers, including but not limited to E2E service design in modular 6G, JCAS protocols, and AIaaS.
Implications	**Requirements**	Firstly, this enabler requires the management functions between RAN and CN to be revisited and changing the core NF design and implementation. As the new NF compositions evolve, new interfaces might be needed for inter-node network-level coordination. Finally, redesign of the architecture considering new approaches is needed which might need a "*clean slate*" approach.

	Standard relations & regulations	The network composition changes as well as the relevant interfaces and interactions would impact all the 3GPP standards including but not limited to [23.501], [23.502], [38.413].
	Needed resources	All the solution is mostly software based

4.2 E2E service design in modular 6G

4.2.1 Introduction

Extending the service heterogeneity in 5G, 6G is envisioned to host a larger number of different services [HEX223-D12]. These services pose strict KPI and QoS requirements, which require the network resources to be optimized according to the needs of the hosted services. To optimize the shared network resources to meet the conflicting needs of these services, 5G utilized the network slicing concept, which is built upon the idea of creating virtual and dedicated resources (i.e., physical and computing) i.e., customized for specific use cases. However, the semi-static nature of the network slicing that is built upon strict SLAs (Service Level Agreements) limits the flexibility and places strains on the efficiency of the network. In order to provide a seamless and effective coexistence among different services, the 6G networks need to have a high level of autonomy and adaptiveness to the dynamic needs of the network. This, however, requires an E2E vision in the network modularization that would cover both UP and CP, namely the deployment location of the network modules as well as the specific use case(s) and their KPIs must be considered in the module design phase.

Moreover, the network autonomy and adaptiveness need to be ensured in this modular 6G architecture. The modules should be designed considering the reusability and autonomy requirements in multi-cloud environments, which would also allow the network providers to scale the available modules on demand. The respective interfaces to support such dynamic control need to be provided in the cloud continuum. The capability to have this high degree of dynamism in control operations would also bring a more efficient multi-tenancy in 6G. In addition to the regulatory implications, this enhanced multi-tenancy requires a revisit of the modular structure to determine the optimal degree of control over the modules that should be provided to a tenant, to ensure scalability, reliability and efficiency in this shared network. Finally, the maturity of AI/ML technology raises the opportunity to integrate certain AI functionalities inside the module design and further enhance the efficiency by proactively customizing the modules to the transient network states. Indeed, this integration requires a clear understanding of the extent of AI functionalities to be placed in the modules and how they should be exposed to the other modules/domains.

In this enabler, the analysis starts with the E2E implications of modularization and its architectural affects to both CP and UP. In particular the modularization in RAN, the RAN-CN interactions and modular UPF are discussed. The analysis is followed by a presentation of the network autonomy and adaptiveness via modularization where the potential of intent-based management/orchestration to manage the modular placement is discussed. Finally, quantum-based synchronisation to meet the extremely accurate inter-module synchronisation is detailed.

4.2.2 Extended/E2E network modularity in UP and CP

4.2.2.1 RAN modularity

The ever-increasing complexity and heterogeneity of wireless networks requires solutions that make RAN *open.* The flexible deployments brought by disaggregation and virtualization of RAN components allow for increased reconfigurability, interoperability and resiliency of the networks [PBD+23].

The move from the monolithic systems of the past towards open and flexible RANs is likely to continue in 6G, improving upon the principles of openness of the current and past generations. Disaggregation of RAN components and its related functional splits defined by 3GPP for 5G NR [38.401] is contemplated as a possible architectural option for RAN in the shift towards 6G.

The RAN functional splits proposed by 3GPP [38.801] divide the functionalities into CUs, DUs and RUs.

Inter-cell interference is an intrinsic limiting factor of network-centric cellular networks [AZD+16]. Users located near the cell edges experience the worst network performance. By shifting the view from network- to user-centric networks, this hindering limitation can be mitigated, providing uniform and reliable performance for all users. Cell-free MIMO is a technology in which each user is served by a group of RUs located around that specific user, effectively eliminating cell boundaries by providing a cell-free experience to users. That is, from the users' perspective, there are no cell boundaries during data transmission, and no perceived reconnection as users move [NAY+17].

RAN disaggregation is a crucial aspect for the practical implementation of cell-free networks, as there is a need for a particular level centralization between RUs, to enable joint processing. Certain functional split solutions allow for flexible and scalable cell-free solutions. Most of the focus is put in the functional split between the RU and the DU. There are some aspects to consider when selecting suitable splits, such as module complexity and fronthaul throughput and latency requirements. As more functions are placed in the RU, coordination between different RUs by the DU becomes more difficult or even impossible, as the channel estimates must be in the DU. The corresponding splits consider the separation of physical layer (PHY) functionalities into the RU and DU. The interface between the RU and the DU, the fronthaul, must support UP and CP traffic between these two nodes. There is a trade-off between the RU complexity and cost, and the amount of data that goes through the fronthaul, as well as the latency requirements. As more functions are placed at the RU, it becomes more complex and costly, whereas the total fronthaul traffic decreases. Additionally, the requirements on fronthaul latency, i.e., between the RU and the DU, become less stringent [38.801].

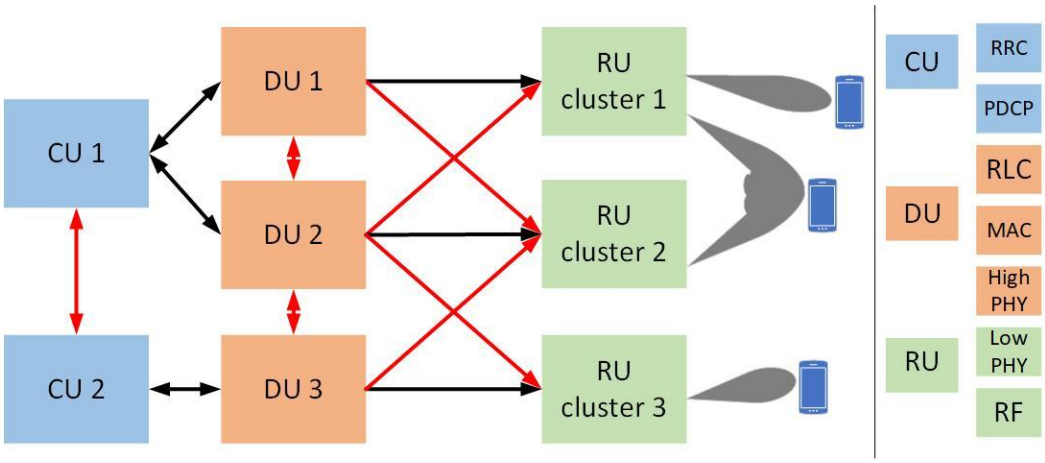


Figure 4-10 Logical RAN architecture enabling distributed cell-free operation. Users may be served by different overlapping clusters of RRHs. For this, RU resources can be divided to different DUs.

A possible implementation of cell-free operation on disaggregated RANs is with the introduction of the concept of RU multi-clustering, as depicted in Figure 4-10. As shown, in this case the RU performs low PHY functionalities such as cyclic prefix insertion or removal and (Inverse) Fast Fourier Transform ((I)FFT). The DU performs the rest of the PHY functionalities as well as providing functionality associated with the MAC and Radio Link Control (RLC). Finally, the upper layers (Packet Data Convergence Protocol (PDCP) and Radio Resource Control (RRC)) are implemented at the CU. The red lines represent the additional interface connections that make possible this implementation, compared to a conventional cellular Distributed MIMO (D-MIMO) network. The RU-DU split corresponds to split 7-2, whereas the DU-CU split corresponds to option 2, as defined by 3GPP [38.801]. The RUs are initially clustered depending on their physical locations, with one DU managing each of the resulting RU clusters, forming the so-called basic clustering. By allowing RUs to be connected to multiple DUs, multi-clustering is formed by the means of shifting, in which the RUs at the borders of the basic clustering are also connected to the neighbouring DUs, giving to each DU a fraction of their radio resources. Each DU therefore manages multiple clustering of RUs, each of them (aside from the basic clustering itself) resulting from a shift of the basic clustering in a certain direction, where new RUs become part of that cluster, and some are removed. The total bandwidth is orthogonally divided in the same number of partitions as clustering's there are in the network (the basic clustering and the new ones obtained

from the new RU-DU connections). Each partition is then associated to a clustering, effectively defining multiple orthogonal (in frequency) D-MIMO instances. By allowing negotiation between DUs on which resources are allocated to the RUs they share, it is possible to formulate a user scheduling and resource allocation problem that aims to improve the downlink spectral efficiency of users, especially of those located at the cluster edges. Ideally, users will be served by an RU cluster that places them at the centre, making the network more user-centric. Comparing the results of this approach to other benchmarks, such as single clustering D-MIMO and collocated Massive MIMO, it can be seen that it provides better service for users, especially for those who suffer from the cell-edge problem. This is presented in Figure 4-11, which shows the empirical Cumulative Distribution Function (CDF) of the average downlink spectral efficiency for the different approaches studied. The gain is higher in users which suffer from high co-channel interference in single clustering D-MIMO, as can be seen by comparing the spectral efficiency of the 5th percentile. Additionally, since the functional split defines the front-haul as carrying frequency domain samples, and the partition is also done in the frequency domain, the proposed solution does not increase the fronthaul traffic, compared to a single-clustering solution (D-MIMO).

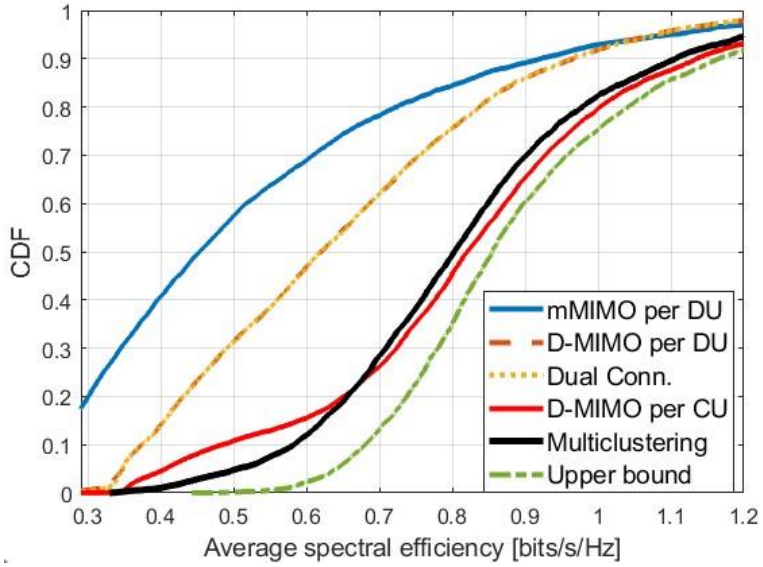


Figure 4-11 Empirical CDF of the user average spectral efficiency, comparing RRH multi-clustering with different benchmarks.

As mentioned in section 2.2.4, the LLS may work well with Cell-free MIMO instead of the aforementioned higher layer split of CU and DU. The LLS is a separation of baseband (RAN stack) and RU by a fronthaul interface. This interface is typically inside the physical layer (PHY, L1). The use of LLS instead of HLS and a split between CU and DU may reduce the number of signalling messages required to acquire and utilize all relevant information for each UE.

4.2.2.2 RAN – CN interface in modular 6G

So far, the discussion has focussed on the modularity and interfaces of the RAN architecture. This section provides a more detailed look at the interfaces between the RAN and the CN and how to improve the E2E modularity. The control plane signalling between RAN and CN communicates via the N2 interface and always communicates via the AMF (regardless of what function in CN being the "final destination"). One reason for using the AMF as proxy is security, including isolation between RAN and CN and a single NAS security termination point for the UE. Any CN function accessing RAN needs to go via the AMF. 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. So, can the N2 interface be made more modular for 6G? One option is to reuse a Service-Based Interface (SBI) as in SBA. One advantage with a SBI interface could be to reuse the SBA framework (e.g., registration, discovery, authorization) also for RAN – CN signalling. However, a RAN-CN SBI has challenges that are described in the following. A possible consequence that would need to be addressed is handling a high number of processes for service registration and discovery from many RAN functions. Furthermore, SBA is considered to be more "chatty" than previous

approaches and this chattiness could impact RAN-CN signalling performance. Security can also be an issue and some alternative to the AMF proxy needs to be found. Alternative here means some functions that provides a similar level of security.

The N2 interface between the current RAN and AMF supports the so-called NG application protocol (NGAP). The underlaying transport protocol for the NGAP is the SCTP. Currently, the NGAP relies on ASN.1 binary format with a specific order of the information (the so-called information elements, IEs) where each IE is associated with specific ID field allowing quicker parsing. NG Setup can be triggered after the Transport Network Layer (TNL) Association, handled by SCTP, between RAN-AMF has become active, see Figure 4-12.

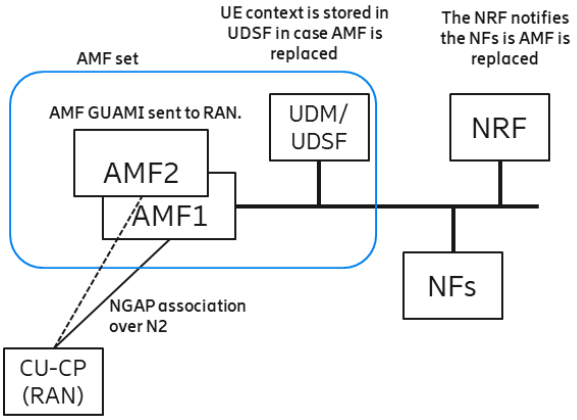


Figure 4-12 NGAP association and the AMF connection to the RAN

The binding between an TNL Association and an NG-AP instance are handled as part of the so-called NG interface management procedures. The NG interface mandates the usage of SCTP as transport protocol and the application logic of the NG interface is defined leveraging on SCTP features such as semi-permanent connections and real-time redundancy information.

The architectures of the RAN and CN CP are expected to undergo changes in 6G for the support of 6G radio features, new applications and services envision for 6G, energy efficient design, and more. Here the objective is to investigate how to design efficient network functions and how these functions interact in the architecture, in terms of combined KPIs such as latency, failure points, dependencies and number of messages. [HEX223-D32] discusses KPIs to use for evaluating efficient signalling and a "KPI map", the NF design, SBA and SBI on general terms. Here, we consider three evolution directions for 6G RAN-CN CP interfaces. The first is considered a stepwise evolution to 5G NGAP while the other two are more revolutionary.

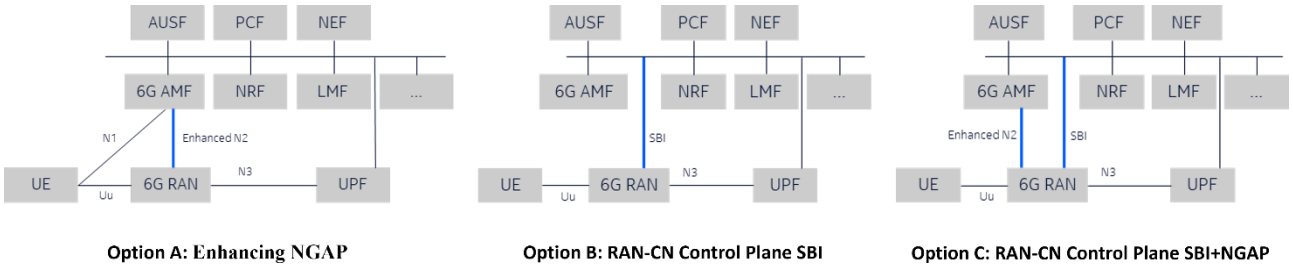


Figure 4-13 Considered RAN-CN CP options for 6G (Option A-C)

In the first option (i.e., Option A), existing architecture and interfaces are used where 6G RAN communicates via NGAP with the AMF, or a 6G equivalent of it. The main features of this approach include RAN maintaining ASN.1 encoded interfaces, 6G AMF maintaining the gateway functionality between RAN and CN NFs, thereby being an evolution of 5G. The nature of 6G AMF and the required modifications to this NF are to be studied. It can either follow an enhancement to AMF functionalities or a completely new NF design.

In option B, the RAN-CN interface is changed by introducing a Service-Based Interface (SBI) between 6G CN and RAN. In this case, this interface would enable the 6G RAN to directly expose and consume services towards and from the CN, respectively. Similarly, it would enable all CN NFs to directly expose and consume

services towards and from the 6G RAN. Communication between CN NF and RAN is performed over defined service APIs and operations. This approach supports the usage of cloud-friendly mechanisms and protocols and enables harmonization of services across the RAN and CN domains, such as security, exposure and discovery, etc. It is important to ensure that RAN and CN domains remain treated separately at standardization and implementation to maintain flexibility (such as RAN sharing), inter-vendor operability, etc. In any case, this option requires significant effort in standardization, implementation, optimization, and deployment.

As a trade-off between options A and B, a hybrid approach is a possible evolution of the 6G RAN-CN CP interface where service-based interaction is introduced with minimal impact on the existing point-to-point interactions and without creating duplication among the two interfaces. This can allow cloud-friendliness, deployment flexibility, and room for performance optimization for certain functionalities and services while decreasing the standardization and implementation impact when compared to option B. Yet, the consequences of implementing multiple protocol stacks in 6G RAN have to be analysed carefully as there might be complications in management, interoperability, testing, etc.

Table 4-4 summaries the NGAP procedures related to NG interface management and a possible equivalent SBA functionality, with focus on differences. The table includes the main management functions supported by NGAP, such as setup, error handling and overload functionalities.

Table 4-4 Summary of NGAP functionality and corresponding SBA functionality

NG management procedure:	**Possible equivalent SBA functionality:**
NG/N2 Setup	Can be replaced by SBA Service discovery/registration. Note that in 5G it is the RAN that triggers NG Setup towards AMF. When using service registration/discovery, it has to be understood if RAN should be discovered or not (and eventually if RAN should register its services). NG Setup also has the functional role to make RAN operational.
RAN/AMF configuration such as PLMN info, Slice support, Default Paging DRX, Globally Unique AMF ID (GUAMI).	The configuration parameters are few and should be possible to get from e.g., the NRF via Service discovery/registration and status notification.
AMF load management and AMF load balancing	The AMF load management still needs to be part of the SBA solution, new or modified procedures need to be standardized and implemented. NOTE: to be understood whether indications about load management should be handled via 3GPP procedures or if it is possible to handle this in the cloud platform with limited or no implications to 3GPP procedures.
Adding new AMF, replace AMF	In NGAP, the new AMF sends out the GUAMI to the gNBs; can this functionality be moved to the NRF and discovery/registration of SBA. The UE context can still be stored in the UDSF if an AMF is replaced.
Multiple TNL Association management	TNL Association management is not needed in SBA; this is handled inherently by the SBA.
NG Reset (e.g., to clean out context in RAN at failure)	This can be handled through NRF which keeps track of function's status (e.g., service/function de-registration when decommissioned). NOTE: to be understood how to handle the reset of signalling relevant to only one specific UE and not to the whole function.
AMF status, Overload start/stop	Possibly this can be handled by NRF changing some service information (or by direct signalling between RAN/AMF). NOTE: NG overload start has also functional value of e.g., informing RAN about which signalling should be prioritized (or rejected) in case of AMF load. To be understood how this could be realized.

Error indication between RAN and AMF	The AMF or the RAN can understand the cause of an error and take action based on this. This cannot be replaced with SBA framework functionality

As a summary of Table 4-4, SBA can replace the NGAP functionality for most cases. However, we do not see any actual benefits by adopting SBA over NGAP. Moreover, the error handling and possibly the AMF load balancing that exist for the 5G NG management is not directly supported by SBA, so these functionalities probably need to be added on top of the normal SBA functionality.

One principle should be that the signalling application is responsible for handling the features required by its logic (following an E2E principle), e.g., determine when a signalling transaction is completed. Transport layer re-transmissions of messages can still be used, but knowledge of application signalling losses should be at application level since this gives more freedom on how an underlying transport is deployed or used. For example, it will then be possible to use multiple transport associations without explicitly managing those associations on application layer or possible to deploy application and transport layers in different container without relying on the assumption that receptions of messages at transport layer implies also safe reception at application. The main benefit in moving the E2E responsibility to the application is that the coupling with the signalling transport goes away, e.g., removing the binding of application signalling to long lived SCTP associations, removing the reliance on SCTP association monitoring. However, this does not necessarily require the adoption of a SBI between RAN and CN, and it should be possible to include this e2e functionalities also to application logic of point-to-point interfaces between RAN and CN (e.g., relying on application layer acks transaction status handling as well as application-level connection monitoring if needed).

4.2.2.3 CN UPF modularity

Finally, while the control plane of 5G Core networks is designed to follow the SBA, the user plane and in particular UPF is still dealt with as one big monolithic block with many functions and features to support (approximately 20 based on 3GPP TS 23.501 Release 18). Big monolithic design of functions hinders the ability to scale in/out resources assigned for subfunctions at a fine-granular level, slows the development cycle in terms of deployment and debugging, reduces the flexibility in using different technologies for developing different components of the design, etc. [AAE16] As a concrete example, the UPF may handle an asymmetric volume of uplink/downlink traffic to be processed at the Core Network. In this case, a modular design of UPF would enable scaling in/out the processing resources for each traffic direction independently based on the demand. Another example is when the UPF is expected to run some features, such as Lawful Intercept, over a specific period of time and over different locations. In this case, the module that implements the Lawful Intercept functionality could be provisioned with processing resources independent from UPF to cope with the volume of the demand, and thus the processing resources would be utilized more efficiently and sustainably. To this end, we propose a modular design for the UPF of the next generation Core Networks. The following entities are proposed [HAR22]:

Ingress Steering Module (ISM): This module handles the incoming packets when they arrive at the UPF. First, it checks the validity of the received packets based on defined admission control rules to drop the invalid packets. Then, it steers the packets to uplink or downlink modules. This module can implement load balancer mechanisms to balance the processing workload when multiple replicas of the following modules (i.e., uplink function or downlink function) exist.

Downlink Module (DLM): This module handles the packets coming from the DN to the gNB and later to the UE. It implements all the packet processing required to fulfil the basic tasks assigned to the UPF when handling downlink data traffic.

Uplink Module (ULM): This module handles the packets coming from the UE through the gNB to the DN. It implements all the packet processing required to fulfil the basic tasks assigned to the UPF when handling uplink data traffic.

On-Demand Modules (ODM)s: This list of modules includes all optional functionalities assigned to the UPF, which can be activated on-demand, and which are considered discrete compared to the basic packet processing. Example of On-Demand functions could be:

1. Lawful Intercept Module (LIM): This module handles the lawful interception of user data for some devices and over a certain period of time on a per-demand basis.
2. Enterprise specific Processing Module (EPM): This module handles Time Sensitive Networking/Communications Translator functionality, or similar functionality defined for supporting integration of 5GS to DetNet, or other similar future enterprise specific handling.
3. ATSSS processing Module (ATSSSM): This module supports ATSSS with all relevant steering functions like MP-TCP, potential future steering functions like QUIC, MP-QUIC, etc., and all existing and future steering modes. Flavors of this module might be steering function specific like, e.g., ATSSS microservice for MP-TCP, ATSSS microservice for MP-QUIC.
4. Redundancy Module (RedM): This module handles user plane traffic duplication and elimination for high reliability like, e.g., support of redundant N3/N9 tunnels.
5. Cellular IoT Module (CIoTM): This module handles IoT related user plane traffic supporting extended buffering, Reliable Data Service etc.
6. Multicast Broadcast Services Module (MBSM): This module supports MBS, MB-UPF functionalities.
7. N6 Tunnelling Module (N6TM): This module serves the various N6 tunnelling options like for example MPLS.
8. DPI Module (DPIM): A typical user plane module applicable in many cases.

The optional UPF functionalities provided by the ODMs can be flexibly activated/deactivated on demand and can also be scaled in/out over time as needed. This modular design facilitates discrete specialization, and the best integrated result Figure 4-14 shows the modularized UPF in the Core Network. The packet processing in the user plane will be defined in the spirit of Service Function Chains (SFC), where these SFCs are created as chains made up of the modules defined above.

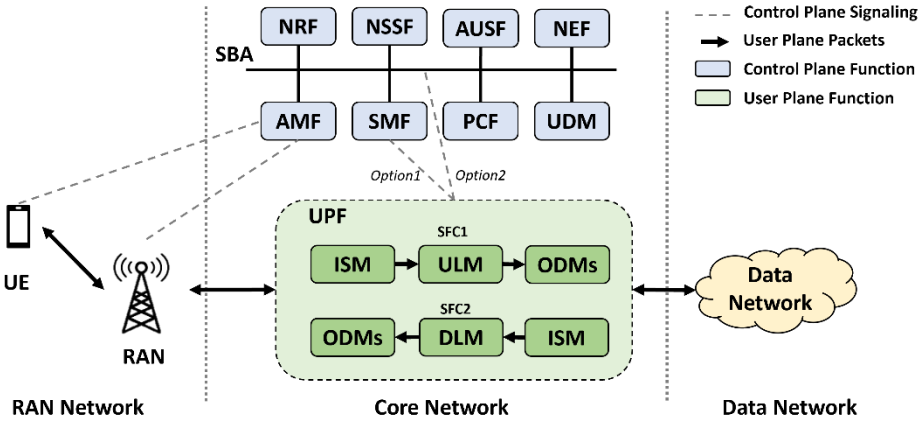


Figure 4-14 Modular UPF design integrated in the E2E mobile network system

4.2.3 Network autonomy and adaptiveness via modularization

The Cloud Continuum is a seamless integration of various types of clouds that extends from the centralized cloud to the on-premises equipment, passing through the far-edge and the near-edge. This extended cloud is hence distributed in nature and possibly constituted by several heterogeneous cloud technologies. Potentially 6G network functionalities should be deployed coherently on all the Cloud Continuum, while being effectively deployed only where the specific function, micro-function or module is needed and only for the time it is strictly needed. Also, different decomposition and interactions between network modules may be implemented in the 6G system differently from the current 5G SBA model. As a matter of fact, as depicted in the previous sections, 6G system may be further disaggregated to encompass submodules of network functions that are composed and reused by several functions to provide the resources needed for fulfilling a services request.

Consequently, the modularization model can evolve during 5G and subsequently 6G and be highly dynamic. Therefore, an orchestration model, which does not bind to the specific architecture of current network modules, is needed; it must be natively extensible and manage orchestration in a generic way, moving away from static 5G workflow-based orchestration models. In this context, a, flexible orchestrator, should deliver carrier-grade, simple, open and cloud native intent-based automation. It must be natively expandable and not related to specific network architecture or specific network modules. It should be based on common automation

templates that materially simplify the deployment and management of multi-vendor cloud infrastructure and network functions across large scale edge deployments. It must enable faster onboarding of network functions to production including provisioning of underlying cloud infrastructure with a true cloud native approach and should reduce the costs of adoption of cloud and network infrastructure. Extension components must be a key feature supported by such an orchestrator to enable the use of custom resources allowing the management of the upcoming modules and their components. The orchestrator behaviour should be extensible without modifying the code by linking controllers to one or more custom resources. Consequently, the orchestrator must be able to manage 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, especially for enhancing networking capabilities and applying configuration customizations to tune performance parameters.

The development of a network orchestrator hinges on its intent-driven nature, a crucial element for ensuring its effectiveness. While the existing automations have many limitations (e.g., complex templates, difficult to read and test, limited re-use due to huge lists of values that need setting), the ability to continuously reconcile systems with an intent-based approach stands out as a significant advantage, particularly in comparison to imperative tools, as it enhances robustness at scale. In the context of large-scale edge deployments, distributed actuation of intent emerges as a requisite feature. The traditional approach of triggering all actions from a centralized location is unreliable and impractical in such scenarios, where scalability is paramount. The intent-based nature of the orchestrator allows to put emphasis on uniformity in systems management enabling the management of deployment, repairs, and configuration through common components and standardized workflows streamlines operations. Focusing on the initial intent, allows to reduce cognitive load for the operations team, while the distributed actuation takes care of the downstream complexities for the edge locations. This approach not only facilitates more rapid responses but also minimizes the likelihood of human errors in the orchestration process.

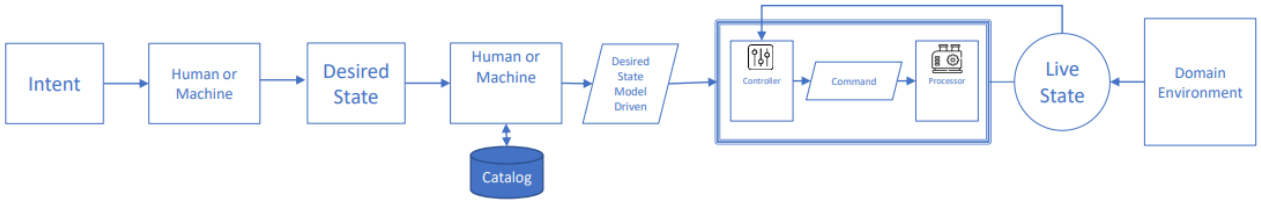


Figure 4-15 Intent-based orchestration and how the intent can impact the live environment

Different network operators may have different requirements and challenges for their network design, performance, and integration which may lead to the adoption of different architectures to suit their specific needs and scenarios. The architectural heterogeneity of different network operators poses significant challenges for the integration and interoperability of heterogeneous networks. On top of this, the cloud continuum represents an additional layer of complexity that can differ from one operator to the other.

An orchestrator that leverages an extendable intent-based approach can be used to comply with regulatory and SLA implications by providing a high-level and abstract way of expressing the desired outcomes and goals for the network and the services, without exposing the low-level details and configurations of the network devices and functions. By using intent-based orchestration, network operators can specify KPIs and service level objectives that they want to achieve, such as availability, reliability, security, latency, bandwidth, etc. The intent-based orchestrator can then automatically translate these intents into network policies and configurations and enforce them across the network. The policies are dynamically applied only on those network function that are affected based on the requested KPIs. Moreover, network modules can be serverless functions in order to provide a flexible and scalable way to implement and execute network functions and services. Serverless modules can enable a more efficient, reliable, and adaptable network service delivery and management since they are intrinsically more suitable for a highly dynamic environment.

This dynamic orchestration necessitates an enhanced synchronization of different modules and distributed cloud functions. This synchronization is crucial for seamless communication and efficient resource management across various network components [UKK+23]. With the increasing demand on high data rates,

even will-established classical synchronisation methods, such as Precision Time Protocol (PTP), are limited by the need for precise topological information and their being susceptible to jitters and inaccuracies [NPS+23]. The integration of quantum technologies / techniques into 6G networks is expected to represent a significant advancement in network synchronisation and security. By employing entanglement of qubits across the network, allowing for a shared state of time that is consistent across all nodes (quantum modules), a quantum master clock distributed across different network nodes rather than centralised at a single location can be realized. This strategy aligns with the concept of the Telecom Grandmaster (T-GM) in classical networks but with the added benefit of entanglement, providing ultra-precise timing information that is crucial for the functionality of real-time tasks such as those demanded by 6G, the Tactile Internet, and Time-Sensitive Networking [KKB+14]. This quantum-enhanced time references can then be utilized within classical network synchronization protocols. This also introduces a level of security inherently protected by the principles of quantum mechanics since any attempt to observe or interfere with the entangled particles would result in detectable disturbances in the system.

In this context, quantum synchronization represents a significant step forward for 6G networks, providing not only a solution to the limitations of current synchronization standards but also paving the way for a new era of communication technologies. However, the availability and scalability of the quantum hardware including reliable quantum repeaters, efficient quantum memory systems, and the integration of quantum error correction techniques is a prerequisite to maintain entanglement fidelity over long distances. Further, integrating quantum modules into standard network designs demands the creation of sophisticated interfaces capable of translating the intrinsic quantum mechanical qualities of entanglement and coherence into signals that can be processed by conventional networking equipment. These interfaces function as transducers, transforming quantum bits (qubits) to classical bits while keeping the temporal information inherent in the quantum state. This procedure involves not only advanced physical devices to detect and analyse quantum states, but also complex algorithms capable of interpreting these observations as useful synchronization signals.

In a hybrid quantum-classical network scenario, a network architecture that synergizes quantum and conventional (classical) timing methods, enhancing both accuracy and security is shown in Figure 4-16. Here, the potential of Time-correlated Entangled Photons (TCEP) to substantially refine time standards, achieving picosecond precision is investigated. The network consists of several components essential for integrating quantum capabilities into classical networks:

- Synchronised nodes: Located at various TCEP nodes, these devices generate the entangled photons necessary for quantum synchronisation across the network.
- Quantum-synchronisation Layer: This layer lets quantum signals pass through it. After detection and post-processing of correlated signals, this layer can make the local oscillators present at TCEP nodes to be synchronised up to the picosecond level.
- Classical Network Architecture Layers: Represented by the Boundary Clocks (BC) and primary reference time clocks (PRTC), these are the standard components of the classical network infrastructure that maintain time synchronisation.
- Dynamic Adjustment Mechanisms: The network's ability to adapt to changing conditions is indicated by the feedback loops from the classical architecture back to the quantum-classical interface modules.

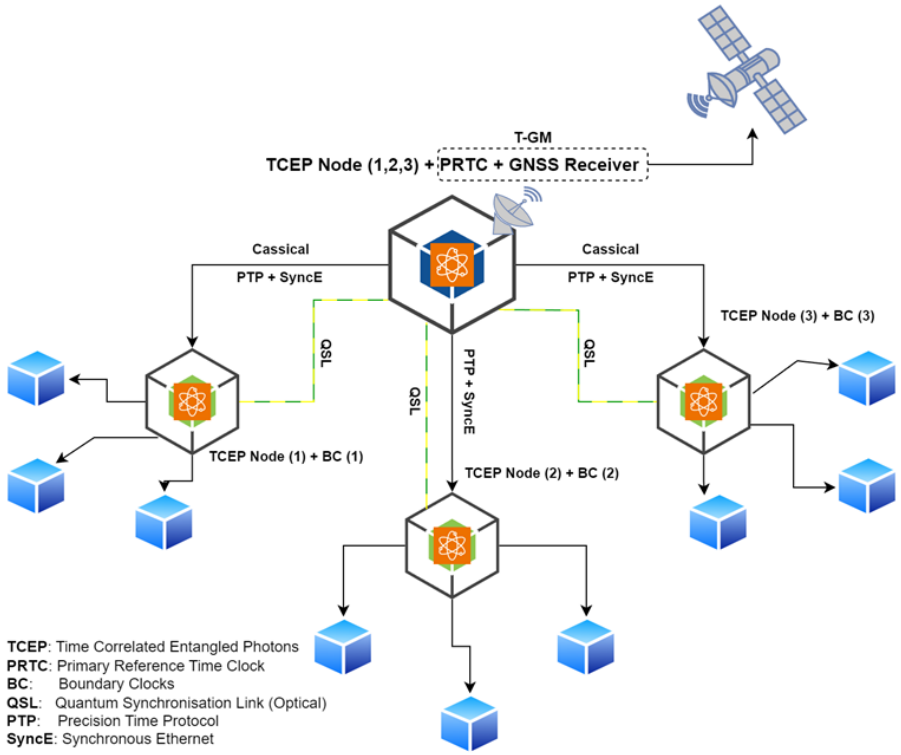


Figure 4-16 Integration of quantum modules within classical network architecture.

The ability to generate entangled pairs on demand and regulate their distribution across the network is critical for realizing quantum technology's full potential in improving network synchronization and communication. For this purpose, a novel scheduling mechanism is proposed that uses time-segmented intervals to alternate between classical and quantum channels. This scheduler is inspired by the Time-aware Shaper of Time-sensitive Networking (TSN) defined in [IEEE 802.1Qbv], a standard already implemented in many classical networking devices and software stacks. This similarity underscores the scheduler's potential for integration and compatibility with existing network infrastructures, providing a streamlined approach to managing hybrid quantum-classical data transmissions while ensuring minimal delay and high efficiency. The scheduler incorporates quantum capabilities into the classical link layer in line with the OIC model. The scheduler plays a central role in determining channel selection based on two different protocols: time-based and event-based scheduling. Figure 4-17 shows a switch that selects the specific scheduler and routes either quantum or classical signals through the scheduler and then through the hybrid channel.

The time-based scheduler, which sits atop the data link layer of the hybrid quantum-classical network stack, uses a unique mechanism to manage data transmission. This scheduler uses two separate queues: one for quantum packets and one for classical packets. Each queue is associated with its own gate - a quantum gate (QGate) for quantum packets and a classical gate (CGate) for classical packets. Packets are enqueued and dequeued only when their respective gates are open.

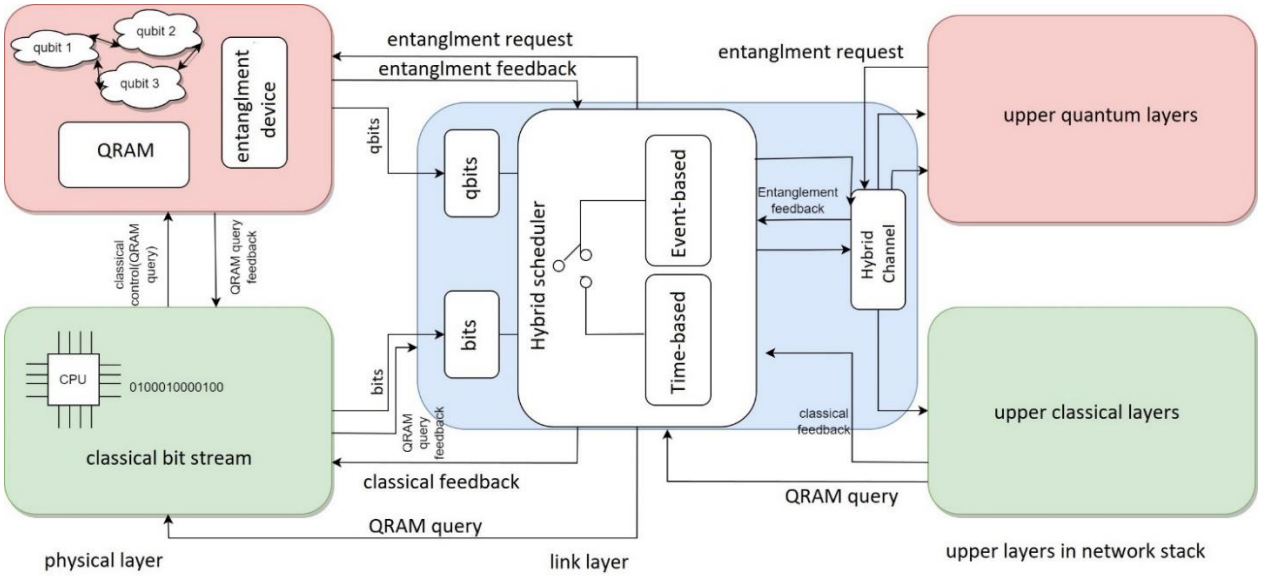


Figure 4-17 Hybrid Classical-Quantum Network structure and flow of information through the stack.

The left part of Figure 4-17 visualises this setup, showing the dynamics of the queues and gate operations. In addition, the scheduler can integrate second-level scheduling protocols, such as FIFO or Weighted Fair Queuing (WFQ), between each queue and gate to improve management efficiency. The transmission (Tx) selection component plays a critical role, allowing only those packets that pass through an open gate to enter the network. The operation of the scheduler is further explained on the right-hand side of the figure, which illustrates the allocation of time windows and slots. A time window is segmented into a number of slots dedicated to either quantum or classical transmissions. When a particular slot, such as a quantum communication slot, is active, the corresponding QGate is open, facilitating the transmission of queued quantum packets. The same principle applies to the classical communication slot and its associated gate. A special feature of this scheduler is the inclusion of an entanglement slot, which is crucial for the generation of entanglement pairs between nodes. These pairs are essential for quantum data communication within quantum communication slots. Depending on the physical capabilities of the network, if simultaneous transmissions on different wavelengths are possible the entanglement protocol can run concurrently with a classical communication slot. Otherwise, the entanglement slot is treated like any other slot, typically placed before a quantum communication slot to prepare for quantum communications.

Dependencies for the execution of hybrid schedular are as follows:

Synchronization: Accurate synchronization is crucial for the proposed schedulers to operate effectively. Employing advanced synchronization techniques, including quantum techniques, has the potential to facilitate seamless integration into hybrid networks.
Noise in the Quantum devices: The functionality of hybrid schedular will crucially depend on the quantum error correction. This is due to the poor noise properties of the state-of-the-art quantum devices.

4.2.4 Summary

As previous section detail, network modularisation has the potential to optimize the network function blocks to meet specific requirements. This section summarizes how the modularization can be utilized at different parts of the network, e.g., RAN, CN, UP, see Table 4-5. Moreover, this new modular structure will require enhanced control aspects, to ensure network autonomy and adaptiveness. In this section, the implications of network modularity on the controller design are presented. In addition to the previously discussed advantages of modularity, e.g., decreased latencies, CP signalling, etc., the extension of the modularity towards RAN would also enable the implementation of cell-free MIMO, which allows for improvements of user-spectral efficiency.

Being a fundamental change in the network architecture, it can enable flexibility in network topologies and resource allocation while providing functional support to meet extreme requirements and dynamically support QoS/QoE, e.g., latency, reliability, high bandwidth utilization or dependability. Consequently, this enabler has

the potential to support all the use case families presented in [HX223-D12]. Similar to *6G network modularization*, this enabler affects all parts of 6GS, as such it can impact all the other enablers and functionalities within RAN, CN, and M&O.

A successful integration of this enabler would require an analysis of the legacy implementations providing some fixes where possible. It would be expected to impact a multitude of 3GPP standards, i.e., including but not limited to [23.501], [23.502], [38.410], [29.510], [24.501], RAN modules defined in [38.401] as well as ORAN.

Table 4-5 Benefits and implications of "E2E service design in modular 6G" enabler.

Description	This enabler focusses on the E2E modular 6GS design for services, that includes, the modular UP/CP design, network autonomy and adaptiveness via modularization and flexible orchestration.	
Benefits	**KPI improvement**	Decreased latency, CP signalling and the number of hops. Increased user spectral efficiency, same front-haul user-plane traffic. Increased efficiency by lower number of interfaces, scalability, flexibility in terms of deployment and execution, and resiliency
	Design principles [HEX223-D21	Being a fundamental change in the 6G architecture, the E2E service design in modular 6G affects all the design principles provided by [HEX223-D21], with particular implications on support and exposure of 6G services and capabilities (#1), Full automation and optimization (#2), Flexibility to different network scenarios (#3), Network scalability (#4), Resilience and availability (#5), Internal interfaces are cloud optimized (#7), Separation of concerns of network functions (#8), Network simplification in comparison to previous generations (#9)
	Dependencies / Basis for another enabler	Depends on the findings on 6G network modularization. As this enabler is a fundamental enabler that has implications to all parts of the 6GS, it has impact on all the other enablers and functionalities within RAN, CN and M&O.
Implications	**Requirements**	The legacy implementations need to be analysed and providing fixes for the legacy implementations. All the proposed changes need to be considered within the gains and loses with respect to the legacy implementations and all non-backward compatible changes need to be justified. The proposed changes need to support new use cases and services and an analysis of different solution options needs to be considered.
	Standard relations & regulations	This enabler will impact the 6G NF/module design and the respective interactions between RAN and CN. Moreover, the autonomous design will have implications also on the M&O. The major standards that are envisioned to be impacted by this enabler includes but not limited to [23.501], [23.502], [38.410], [29.510], [24.501], RAN modules defined in [38.401], Open RAN (O-RAN) considers an open-interfaced disaggregated RAN.

5 Architectural enablers for new access and flexible topologies

Three architectural enablers for new access and flexible topologies were identified in [HEX223-D32], namely "network of networks", "multi-connectivity" and "E2E context-awareness management", along with problem statements related to each of these enablers. This chapter presents architectural proposals, which are required for the operation of these enablers, as well as their evaluations. Each architectural enabler is mapped to the 6G E2E system blueprint Figure 5-1 [HEX223-D22], as explained in the introduction of each enabler's section. Note that the mapping corresponds to the studies of each enabler in this deliverable.

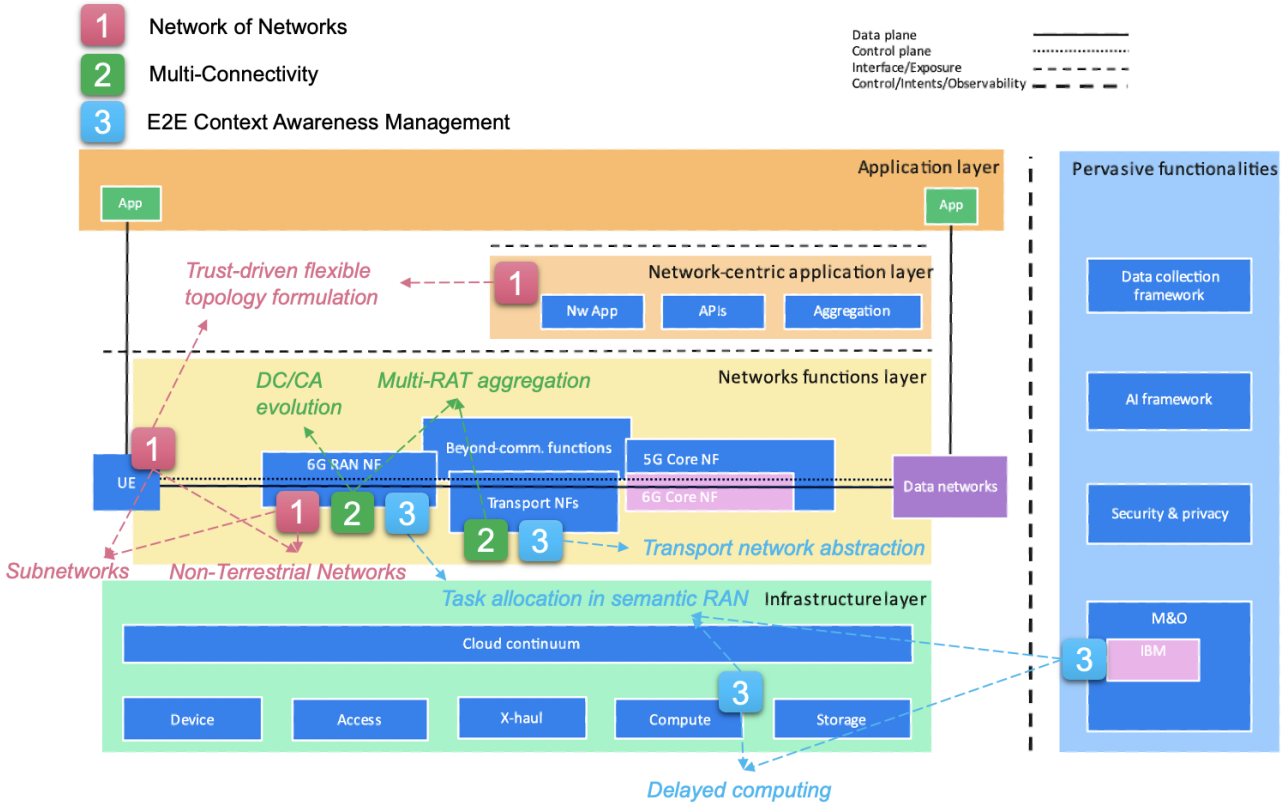


Figure 5-1 Mapping of the network of networks, multi-connectivity and E2E context awareness management enablers to the 6G E2E system blueprint of [HEX223-D22].

5.1 Network of networks

5.1.1 Introduction

The previous deliverable [HEX223-D32] stated that the network of networks is the enabler of a seamless and ubiquitous communication system. The use of NTN and terrestrial subnetworks can aid in achieving a network of networks. With the use of NTN, cellular coverage can be extended to the underserved areas and innovative methods for accessing the Terrestrial Networks (TN) and NTN cells during the mobility of both the device and the base station (e.g., satellite) would be required. Due to the new services introduced in 6G, a revision of the notion of coverage may be needed, where coverage would be defined in relation to other KPIs such as latency, throughput, or robustness at a certain location. This revised notion of coverage would then be used for the evaluation of the various architectural enablers. Forming inherently trustworthy subnetworks will also both extend coverage and create a seamless communication system. In order to achieve the latter, a device may smoothly transition from being served directly by a Base Station (BS) to being served by a Management Node (MgtN) and vice versa. Using AI/ML-driven assessment to determine the optimal flexible network formation in regular intervals will also contribute to these goals.

5.1.2 Architectural implications

5.1.2.1 TN-NTN integration

There are several architecture options for NTN for 6G. The most common architecture so far has been the transparent option, which means that the satellite relays (using the "amplify and forward" scheme) the signal from the UE to another UE (or to a ground station). The BS is located on the ground together with a ground station. The ground station is a wide antenna dish relaying the constructed signal from the BS to the first satellite (or vice versa, receiving the signal from a satellite). For the regenerative architecture [HEX23-D53], the BS is located onboard the satellite, which means that there is no need for a BS to be placed together with the ground station.

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].

An alternative option to a regenerative or transparent architecture is to deploy only some of the BS's functionalities on the satellite. One option is for example to include the RU on the satellite. This means that 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, see Figure 5-2.

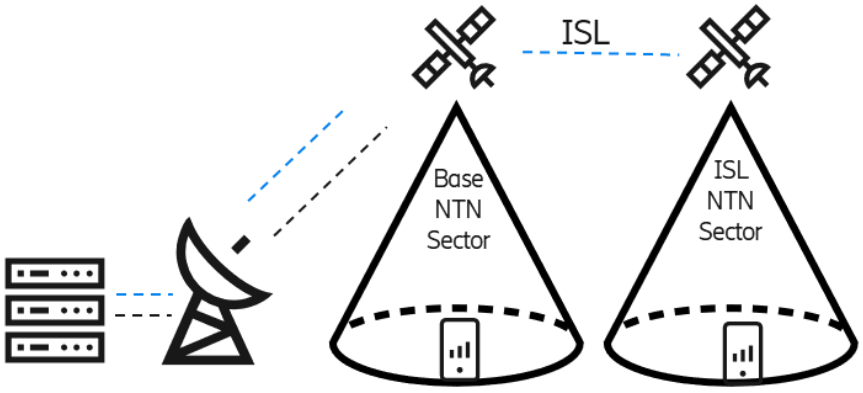


Figure 5-2 An NTN architecture where the BS is on the ground, together with the ground station antenna, and a multi-hop ISL.

To get full coverage, inter-satellite links (ISLs) are needed to support remote areas such as oceans, where there are no visible ground stations. In that case, the total delay may differ between different connections, and the gNB on the ground would need to know the delay when it constructs and schedules the data to the UE. In 3GPP there is no L1-based multi-hop, but it may be possible to use an extension of current LLS (cf., Section 2.2.4).

Another NTN architectural option for 6G is to have the DU onboard (i.e., PHY, MAC and RLC layer of the RAN stack). With the DU onboard, the Integrated Access & Backhaul (IAB) scheme developed for 5G is possible to be utilized for multi-hop ISL communication. However, with the current trends towards using a lower layer split instead of a higher layer split, it is not certain if 3GPP will adopt the CU/DU split for 6G. For this reason, the option to have the full BS onboard the satellite can be a more future proof track for the 6G NTN architecture.

Traditionally, dual connectivity has been used to enhance user throughput, whereby a UE connects to both Master Node (MN) and Secondary Node (SN). Due to long round trip time for NTN radio access [38.821], achieving a greater user throughput during NTN-TN dual connectivity may not be feasible. However, the dual connectivity mechanism can be used to provide coverage extension allowing a quicker switch from TN to NTN, where the TN BS does not have good coverage.

Regarding the role of the NTN and TN BSs as either MN or SN, there are two options. The first option is that the UE is connected to the TN as MN and to the NTN BS as SN. In this case, on the UE side, the NTN SN could be in deactivated state, or the NTN RF chain could be switched off for power saving purposes when the UE has a stable connection via the TN BS. If the NTN RF chain is switched off in the UE, then one benefit is that the UE will not perform any NTN measurements but will keep the RRC configuration for NTN. This RRC configuration will be ready for use once the UE moves away from the coverage of TN cell (and reconnects to the NTN). To detect the TN coverage loss and to switch on the NTN RF chain well in time, the TN network may configure UE measurements such that the UE triggers a measurement report when the UE is about to lose TN coverage. This can be achieved by either configuring triggering of measurement events earlier or by defining a new measurement event when UE is about to lose TN coverage. Once the UE triggers this measurement report, it may switch on the NTN receiver at the same time. After receiving this measurement report, a TN BS provides a notification to the NTN BS to take the control of the UE and perform uplink/downlink transmissions. While the UE is out of coverage of the TN BS, the NTN cell could either continue to assume the role of SN or switch to the role of MN. 5G already supports procedures for connection via SN when MN connection has failed, therefore the same procedures and signalling can be reused. Therefore, it is not necessary that the NTN BS always takes the role of MN.

The second option is that the UE connects to the NTN cell as MN and to the TN cell as SN. In this case, any mobility between the TN cells can be handled as SN change or Primary Secondary Cell (PSCell) change, which are well defined procedures in 5G. However, one disadvantage of this approach will be that UE handover will take place very frequently due to the movement of satellites and not necessarily due to UE mobility. In this case, the UE will suffer user plane interruption due to handover every few seconds. This approach may have detrimental impact on UE power consumption and signalling load. Therefore, the first architectural TN-NTN option may be more beneficial than the second approach, at least in terms of reduced UE power consumption and lower service interruption time. However, selection amongst these two options depends on factors like agreement between NTN & TN MNO and the core network connectivity for each cell and UE.

Finally, for the TN or NTN cell and mobile edge resource allocation, It may happen that a service is terminated at the TN edge and edge resources are located in a closer proximity to a TN cell/network infrastructure for service delivery. These resources may be deployed in different geographical locations compared to the NTN edge resources. For NTN-TN dual connectivity and when the switch happens from the TN to the NTN BS, the compute resources should also be moved accordingly in order to meet service requirements. So, when a UE is about to fall into a TN coverage hole then the compute resources at the edge of the TN BS are moved from a node closer to the TN BS to a node closer to the NTN base/earth station.

5.1.2.2 Subnetworks

A Subnetwork (SubNW) is formed voluntarily by UEs based on mutual trust. A UE may assume the new role of a MgtN in a SubNW and become the SubNW's primary node, which can communicate with the BS and other UEs. The MgtN would take over the coordination within a SubNW while still being in coordination with the global 6G Network (NW). Even though the SubNW is formed among UEs with limited NW configuration and awareness, the SubNW itself may be associated to the global 6G NW, along with all local UEs. In addition, the SubNW shall be well integrated in the 6G NW, so that local UEs are able to seamlessly move between the SubNW and the global NW.

One architectural option is that a SubNW is unknown (i.e., transparent) to the global NW, where the NW is unaware of the MgtN acting as man-in-the-middle (e.g., impersonating multiple UE IDs). The BS communicates virtually with individual UEs, but the physical connection is always with the MgtN. The challenges of this architecture are two-fold. On the device side, the MgtN needs to be capable of instantiating multiple UP/CP entities for all devices, as well as of sending and receiving on behalf of every device in parallel. At the same time, the MgtN needs to be ensured that its own device capabilities will not be exceeded. On the NW side, the BS shall unnecessarily maintain individual procedures per UE (e.g., L3, L1, CSI-measurements/reporting, link adaptation, timing advance), which will have the same results, since all links will in practice be the same BS-MgtN link. Finally, since the NW is not aware that a SubNW exists, licensed spectrum cannot be used in the SubNW, and the NW cannot provide any new SubNW-specific functionality.

Another architectural option is that a SubNW is known (i.e., non-transparent) to the global NW, where the NW is aware of the MgtN and its role, which is to manage local devices and aid other UEs in the SubNW with CP procedures. The UP for all UEs should be handled through the physical MgtN-BS connection, where the MgtN transfers data between a UE and the BS. Even though the challenges of the previous architectural option are resolved here, the aforementioned responsibilities of the MgtN have to be defined.

The CP entities of local devices can be flexibly deployed on the MgtN. The global NW does not have to be aware of this CP offloading. Figure 5-3 depicts an example of such an architecture, where the whole or part of UE3's CP is deployed at the MgtN. The SubNW may then use a new lightweight SubNW CP (snCP) between the MgtN and the UE. Note that the snCP is transparent to the NW, since it includes the configuration and the procedures that take place within the SubNW, as well as the offloaded UE CP information. An alternative architectural option could be that each UE's CP is terminated at that UE. In that case, the MgtN forwards the RAN-BS's CP data to each UE. Another option would be for a UE to offload its CP to another UE and not to the MgtN. This is similar to Figure 5-3, with the difference that the MgtN forwards the RAN-BS's CP data to UE3 and then UE3 forwards that to UE2 in order to decode it and send a simplified version back to UE3. Still referring to Figure 5-3, the MgtN-BS interface would be the 6G Uu-equivalent interface, while the MgtN-UE3 interface may be any access network (e.g., WiFi, Sidelink, Uu). The higher layers of the UP terminate at the local device (e.g., PDU session, IP addresses). The lower layers may terminate at the MgtN, making the MgtN-UE interface SubNW-specific.

Regarding the relevant use case families, the immersive experience and cobots use case families [HEXD223-D12] can be enabled by subnetworks, due to the locality, the density and the diversity (e.g., high-capability, low-capability) of the involved devices. Moreover, the subnetwork topology and the MgtN may also be used in the context of aiding energy-neutral devices with wireless power transfer by receiving those devices' data from the base station and buffering it until they wake up to receive it [HEX224-D53].

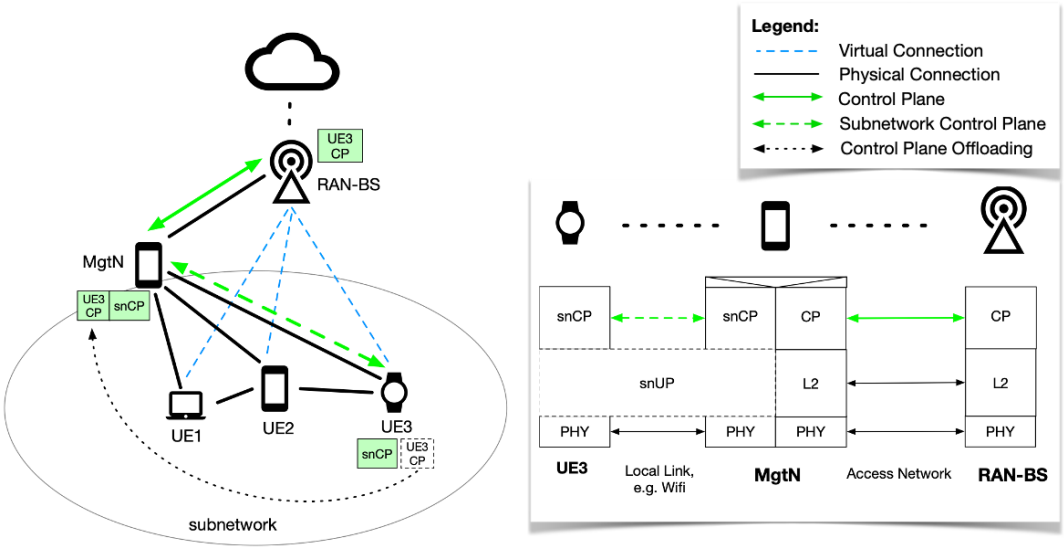


Figure 5-3 UE CP deployed at the MgtN and use of snCP within the subnetwork.

5.1.2.3 Trust-related network functions

Traditional mobile network operator (MNO) infrastructures, built on pre-deployed static frameworks, often face challenges in dynamically adapting to traffic bursts and varying user requirements. Rather than relying solely on central nodes, the system should lean towards a mesh-like structure, allowing for greater flexibility [EGG20]. The potential incorporation of terrestrial and aerial nodes, such as UAVs, into these flexible network formations further broadens the architectural horizon. Such integrations emphasize the need for a comprehensive trust framework that spans both terrestrial and aerial nodes, ensuring that the Connectivity-Everywhere paradigm is not just achieved, but is also trustworthy, which, within this context, refers to a multi-dimensional concept encompassing not just the security of the network, but also its capacity, energy efficiency, throughput, cost-effectiveness, and related aspects.

The proposed solution follows this decentralized paradigm, with functions comprising node discovery and trust evaluation at its core. The node discovery function operates by continuously scanning the network environment to identify and register new nodes. This process involves collecting information such as node identification, location coordinates, operational status, and network capacity. By maintaining an up-to-date map of the network, the system can efficiently manage and optimize connectivity paths. Functions such as continuous trust evaluation introduce another layer of complexity. Architecturally, this means that the network must be capable of real-time assessment of node trustworthiness. The trust evaluation function, on the other hand, is built on a sophisticated algorithm that assesses various aspects of each node's performance and behaviour. Nodes are assigned trust scores, which are dynamically updated based on ongoing interactions and performance metrics. This evaluation is based on various parameters, including historical performance data, response times, data integrity, and security certifications. Trust between two nodes is established through a mutual evaluation process, where each node assesses the other based on the trust scores and recent interaction history. This ensures that connections are made only with nodes that meet the required trust threshold, fostering a secure and reliable network environment.

The trust related functions draw data from the node discovery component and the AI/ML-driven resource optimization component, leading to a tight coupling between trust assessment and resource optimization in the 6G network ecosystem. As such, the communication between these components becomes crucial. Networks must be architected in a way that allows for seamless data flow and instant decision-making, ensuring that only the most trustworthy nodes are selected for data transfer, even in dynamically changing scenarios.

Furthermore, the AI/ML resource optimization component, while essential for trust management, also holds broader architectural implications. Its integration means that 6G networks will inherently be smarter and more autonomous, capable of making real-time decisions based on a combination of node trustworthiness and resource availability. This reduces the need for manual oversight, but at the same time, necessitates robust AI/ML algorithms embedded at the core of the network design.

5.1.3 Evaluations

5.1.3.1 User-centric coverage models and prediction

The concept of cellular coverage has been largely unchanged since the early days of cellular systems: A certain location is considered 'covered' when a connection with a device can be maintained with a sufficiently high probability. Thus, cellular coverage maps that are maintained by operators typically show coverage as a binary reality (either a geographical location is covered or not). Sometimes, operators introduce some nuances by showing a few distinct maps for certain phone conditions (indoor, outdoor, for instance) or data rates, but in essence, a location is considered covered when a connection can be maintained sustainably, regardless of other indicators of service quality. In the new 6G ecosystem, such a rudimentary assessment of coverage may not be adequate to describe the full nuances of 6G. In fact, in other parts of this document the notion of coverage has been related to other KPIs such as the latency, throughput, or robustness at a certain location.

While network of networks and the integration of NTN will improve coverage, a clear understanding and evaluation must be adopted for what this implies. The notion of coverage needs to be tied to one or more certain service quality, e.g., throughput, latency. For instance, a geographical location may be covered under one latency requirement and not covered under another. In fact, because of the broad variety of 6G use cases, the actual support of a certain use case determines the coverage of that use case in a certain location. For instance, NTNs are likely to provide lower throughputs and higher latencies in certain locations compared to terrestrial networks. Certain use cases will therefore not be supported regions with only NTN support.

Furthermore, a large-scale assessment of a region's coverage is currently often carried out by (1) assessing the percentage of the area that is covered and (2) by graphical coverage maps of a region. While the first notion (i.e., the coverage percentage) provides a simple, easy-to-compare, scalar value, the second (i.e., coverage maps) has limited use for comparing regions quantitatively [CB24]. Especially, there is a need to compare networks in terms of the coverage of rural regions versus urban regions. Currently, good quantitative measures are largely missing. When 6G introduces NTNs and multi-connectivity, which are expected to improve coverage in rural regions, measures (beyond mere areal percentage) are needed to assess and compare networks.

A new notion of "*coverage fairness*" across a larger geographical region is needed that take into account other KPIs (such as local latency or throughput) and can be formulated based on the broadband inequality index proposed in [Fou16]. This notion then captures the extent *and fairness* by which a large geographical region is covered with 6G access. It provides quantitative understanding to what extent a new network-of-networks architecture improves coverage into the rural and remote parts of a large region. Hence it can be a useful tool for operators. The fairness indicator combines classical radio propagation models with analysis tools from economy and uses notions as the Lorentz Curve and the Gini Index and applies these to the detailed coverage maps of a network, see [Fou16]. The Gini index and related indices in the literature then provide a scalar value between 0 and 1, where 0 reflects complete uniform distribution across the region, while a value of a reflects a completely inequal scenario where distribution is concentrated in a single point.

Figure 5-4 illustrates the development of the areal coverage across a large-scale network deployment region (typically a country) and shows the percentage of areal coverage (x-axis) along with a fairness indicator (y-axis), which represents the extent to which the coverage is equally distributed over urban and rural parts of the area. With such a graph, a network's deployment in a large-scale region can be tracked over time: after a standard has been established the deployment of networks takes years and develops over time. Typically, upon initial deployment, the coverage percentage will be low, and since initial deployments typically appear in urban environments, the fairness indicator will reflect this through a high value (i.e., connectivity is concentrated to small regions in the region). In later stages of network development, typically not only the percentage of coverage increase, but also the coverage fairness with a more even distribution of coverage in urban and rural regions.

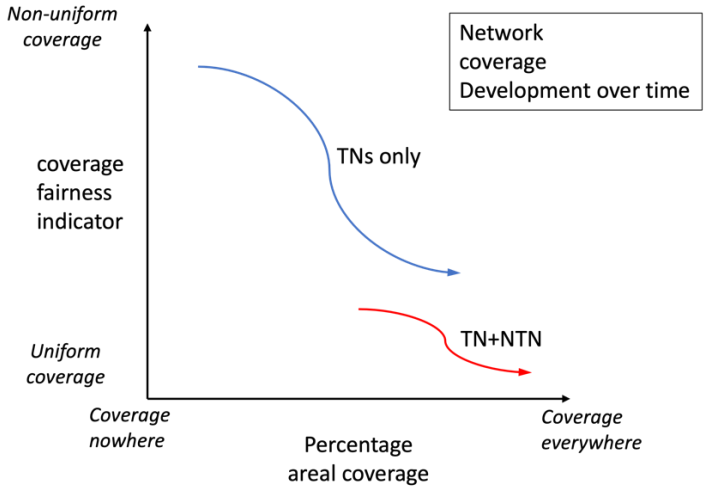


Figure 5-4 Anticipated development of a network's coverage over time in terms of 'percentage areal coverage' (x-axis) and 'coverage fairness' (y-axis).

This notion is proposed specifically for application to the introduction of NTNs, such that in a later stage the value of NTNs in terms of large-area coverage improvement can be related to existing coverage by terrestrial networks. Due to the distinctly different nature of NTNs' deployment compared to terrestrial networks, the development of these networks over time will likely not follow the above-described trajectory of terrestrial networks in this diagram.

5.1.3.2 Flexible topology instantiation and evaluation

The foundational principle of this study lies in the development of a robust framework that facilitates the on-demand realization of scalable, resilient, and flexible network topologies. This framework, which relies on procedures such as node discovery, trust assessment, and resource optimization, employs AI/ML techniques to ensure seamless connectivity associations based on parameters such as cost, trust, resource availability, and specific traffic source demands. A significant step in this research is the introduction of a methodology for dynamic drone placement. This strategy, which is shown in Figure 5-5, unlike traditional brute-force or grid-based approaches, leverages an innovative isometric grid to approximate drone positions, significantly enhancing the search process. Coupled with the utilization of Minimum Spanning Trees (MST) that minimize edge weights, and thus propose drone positions, the process of on-demand topology creation is streamlined. A

pruning algorithm further optimizes the network by iteratively eliminating non-essential leaf nodes, while redundancy elimination ensures that only those drones that enhance connectivity remain part of the network.

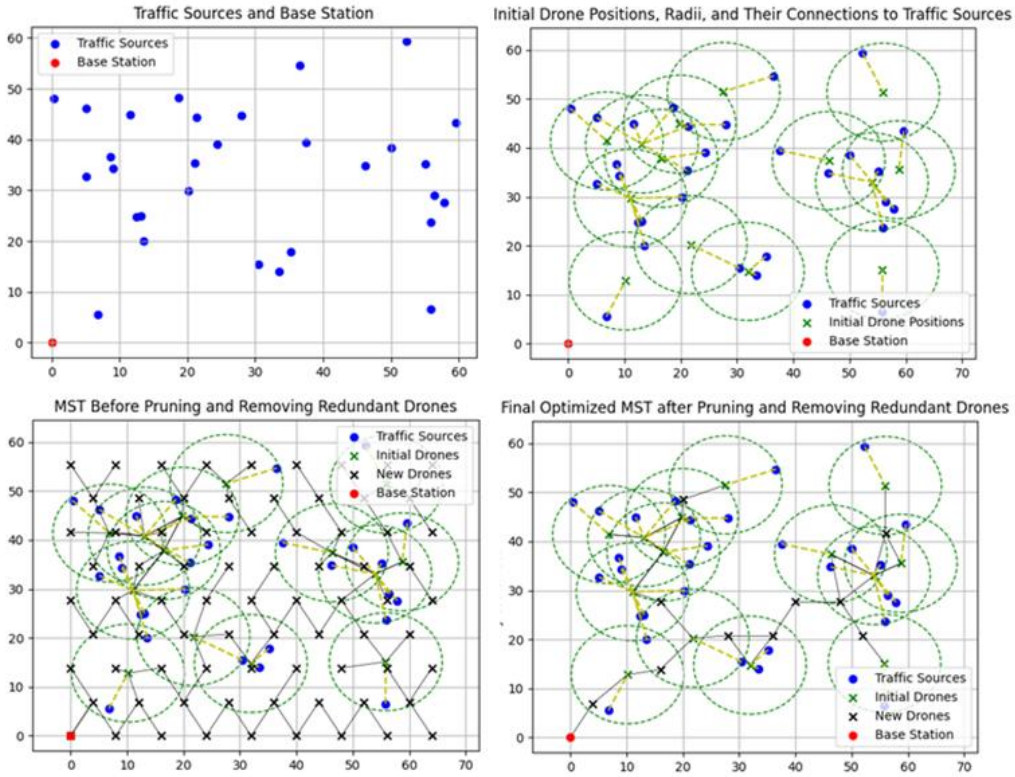


Figure 5-5 Flexible Network Topology Using Systematic Drone Positioning

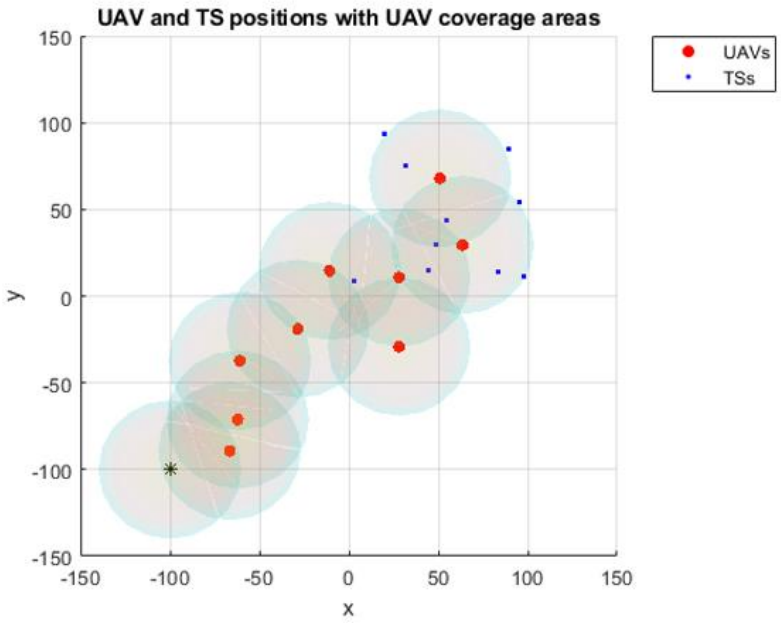


Figure 5-6 AI-Enhanced Flexible Network Topology Using Dynamic Drone Positioning.

The described procedural algorithmic approach outlines the steps to ensure optimal drone placement, in terms of coverage, for seamless network connectivity. Through the efficient use of isometric grid points, distance matrices, MSTs, novel pruning and redundancy elimination techniques, the solution ensures that all the traffic

sources will be served from at least one access point. Moving into the heuristic aspect of this research, the study expands its scope by using Genetic Algorithms (GA) for the serving of multiple traffic sources (TS), as shown in Figure 5-6. This approach, more than just a test but a thought-out exploration, complements the systematic methods mentioned before. The GA, designed to handle various challenges like trust levels, energy needs, bandwidth limits, and related costs, provides a flexible solution, since it can smoothly adjust to changing environments and capture the core ideas of scalability and adaptability.

In essence, flexible topologies not only address the need for dynamic connectivity in remote areas, but also ensures that considerations such as infrastructure costs, energy consumption, trustworthiness, and sustainability are at the forefront of the solution. This paradigm-shift in network design and implementation, with its systematic and heuristic approaches, stands as a testament to the transformative potential of 6G telecommunications in addressing the plethora of use cases of the next generation of networks.

5.1.4 Summary

Apart from enabling a seamless and ubiquitous communication system, this enabler can aid the reduction of complexity in certain devices and network nodes, by offloading some of their responsibilities to other more capable nodes, and also improve the connection reliability (e.g., via using NTN as anchor cell in certain scenarios). As shown in Figure 5-1, this has implications on the RAN signalling and procedures and hence new or enhanced methods for the communication between non-terrestrial nodes, as well as between a device, a terrestrial cell and a non-terrestrial cell. Similarly, the roles and responsibilities of each node in a subnetwork will also affect the RAN procedures on both the devices and the BSs. Moreover, a methodology for a trust-driven intelligent selection of nodes for formulating a flexible topology, which could either be network-centric or device-driven, should also be proposed. Table 5-1 summarizes the main benefits and implications of the network of networks enabler.

The use case families “fully connected world”, “immersive experience” and “collaborative robots” [HEX223-D12] may be enabled by the network of networks. The fully connected world use case family aims to provide network access everywhere, including remote areas with difficult access, airspace, and oceans [HEX223-D12]. The service coverage extension offered mainly with the use of NTN and the NTN-TN dual connectivity, as well as with subnetworks, may enable certain services of this use case. Immersive experience may involve multiple devices of various capabilities and/or owners, which may be co-located. Coordination between the devices and with the overlay 6G network can be achieved via the use of subnetworks. Collaborative robots will require the ad-hoc formation of trustworthy flexible topologies, which is part of the network of networks enabler.

Table 5-1 Benefits and implications of "Network of networks" enabler

Description	Design of terrestrial subnetworks and NTN to create a seamless and ubiquitous communication system.	
Benefits	**KPI improvement**	Increased coverage, reduced interruption time and time in outage, increased availability and reliability, reduced complexity, enabling service continuity.
	Design principles [HEX223-D21]	The flexible topologies of the terrestrial subnetworks and NTNs contribute to the flexibility to different network scenarios (#3 design principle in [HEX223-D21]). Moreover, NTN and terrestrial subnetworks will improve coverage, which fits in higher availability and resilience (#5 design principle in [HEX223-D21])
	Dependencies / Basis for another enabler	NTN-related mobility and subnetwork-related mobility and flexible radio protocol depend on this enabler. Both low-cost and high-capability devices could use the framework of subnetworks, where certain functionalities are offloaded to the MgtN.
Implications	**Requirements**	The ISL solution for 6G NTN, e.g., modified IAB or any other multi-hop solutions depends on which NTN RAN architecture is selected for 6G NTN (e.g., gNB onboard, RU onboard).

		A TN-NTN dual connectivity procedure should be introduced in order to switch faster to NTN in the cases where there is no good TN coverage. The new UE roles and their responsibilities in a subnetwork, as well as the coordination with the overlay network, will have an impact on both the UE and the RAN functions. Two architectural options are foreseen for subnetwork: transparent to the NW and non-transparent to the NW. The latter resolves the challenges of the former but requires the definition of the MgtN-NW procedures. A new lightweight subnetwork CP between the MgtN and the UEs in a subnetwork can be introduced. The trust of diverse network nodes may be device-centric or network-centric.
	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. The architecture of the subnetworks, the new UE roles and their responsibilities in a subnetwork could be defined in 3GPP RAN2.
	Required resources	For NTN-TN DC for coverage enhancement, edge compute resources will need a migration from a node closer to a TN base station to an NTN base station/earth station in order to reduce latency.

5.2 Multi-connectivity

5.2.1 Introduction

The previous deliverable [HEX223-D32] stated that Multi-Connectivity (MC) enables the aggregation of different carriers and Radio Access Technologies (RAT). CA is used for increasing both the user and the system throughput. DC usually employs either different Frequency Ranges (FR) or different RATs to increase either the user and system throughput or the robustness and reliability of the system, by offering multiple Over-The-Air (OTA) paths for transmitting data. The evolution of CA/DC from 5G to 6G aims to combine the best features of DC and CA to avoid the complexity of having two similar solutions, and instead focus on one 6G MCV solution. This enabler also targets to decrease the complexity of activating carriers of different FRs, which have an inherently high-power penalty because of monitoring the different FRs. In parallel, the enabler also investigates the aggregation of different access networks, such as Wireless Local Area Network (WLAN), to enhance coverage while still providing the increased reliability of having multiple OTA connections. The benefit of a device having simultaneous connections with multiple network nodes is evaluated with a study on how to manage communication and computation resources more efficiently in such scenarios.

5.2.2 Architectural implications

5.2.2.1 CA/DC evolution

In [HEX23-D53], a new 6G multi-connectivity solution was proposed, cf., Figure 5-6 for a high-level view. The aim with this 6G MC proposal was to combine the best features from CA and DC to provide both extreme reliability and excellent flexibility, as well as simplify the solution by reducing the number of architecture options. The proposed solution combines the best features from CA and DC. The new solution aims to decouple

Downlink (DL) and Uplink (UL) (e.g., two DL connections and one UL connection, see Figure 5-6) and inherent use of in-active connections. For the in-active connections, the UE only need to sparsely monitor the control signalling from the network. In addition, the in-active connections should be able to be activated on a short notice. To increase the robustness of the system, there is a need for a more flexible use of the UL so that the SCell may take over the role of control signalling in the UL. The CA/DC evolution should also take into consideration the introduction of flexible topologies (e.g., NTN and subnetworks) of the network of networks enabler.

As part of the CA/DC evolution, faster addition of cells compared to 5G would be beneficial. Early Measurements Reports (EMR) was introduced in 5G by 3GPP to allow for proactive UE measurements on NW preconfigured LTE/FR1/FR2 frequency layers, which would be ready for potential reporting once it transitions from Idle to Connected mode. The NW can use the measurement results provided by the UE to configure it with an SCG (i.e., active PSCell) and/or one or more activated/deactivated SCells, resulting in reducing the CA/DC setup delay significantly (i.e., from >500 ms to ~50-70 ms). However, this would require the UE to keep measuring the configured frequency layers (i.e., at least for 10s-300s) in Idle mode, deeming the EMR feature too power hungry (i.e., especially for FR2 layers if FR2 PCell is not supported), hence not gaining much traction.

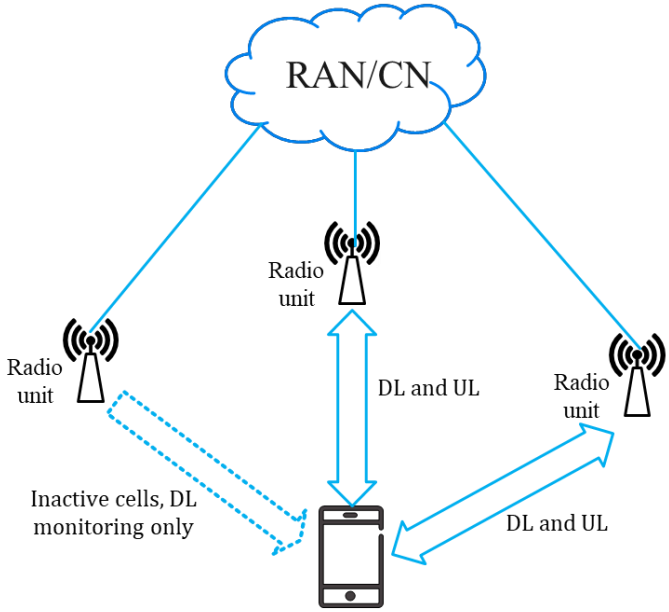


Figure 5-6 Proposed 6G multi-connectivity solutions overview [HEX223-D32].

An enhanced mechanism for PSCell/SCell addition when transitioning from Idle mode to Connected mode could be introduced. With this mechanism, the UE may perform measurements during Idle mode of specific, pre-configured PSCells and not on all frequency layers. Note that both the decision to perform measurements and to report them are based on the UE's internal logic. During the random access procedure, the NW could then directly setup an SCG based on reported measurements by the UE, allowing for faster DC setup. Section 11.3.1.1 details an example of a fast addition of a PSCell or SCell when transitioning from Idle mode to Connected mode. This mechanism could be combined with the feature where the NW configures the UE with a set of PSCells during RRC Setup, where only one of them is activated right away (e.g., the first in the list, or indicated via a MAC CE) and the rest are deactivated.

The concept of multi-server offloading (cf., Section 11.3.1.2 for details) can be particularly benefited from the advancements in next-generation multiple access techniques and the support of radio resource sharing among multiple concurrent user transmissions [LZM+22]. To effectively treat interference under a multi-user multi-server communication scenario, different users can utilize different frequency bands to accommodate their computation task offloading. However, a single mobile user's concurrent transmissions to multiple Multi-access Edge Computing (MEC) servers can be performed over the same time and frequency resources by employing a NOMA technique, such as the power-domain NOMA and the emerging Rate-Splitting Multiple Access (RSMA) techniques [LZM+22]. At the same time, however, this implies that MEC's performance is majorly interwoven with the radio resource allocation, and thus should be studied jointly. Typical optimization

objectives constitute the minimization of the total multi-user multi-server network's time and energy overheads due to both communications (i.e., offloading) and processing locally at the user devices and remotely at the offloading MEC servers. Also, typical optimization variables and controllable parameters comprise the users' task offloading decision, subchannel assignment, uplink transmission power, and allocated computing resource by the respective MEC server.

5.2.2.2 Multi-connectivity with different access networks

Different access networks could be integrated with 6G cellular networks in the RAN by using adaptation protocol layers or in the transport layer by using network virtualization and programmability. In this section, architectural options for each option are described.

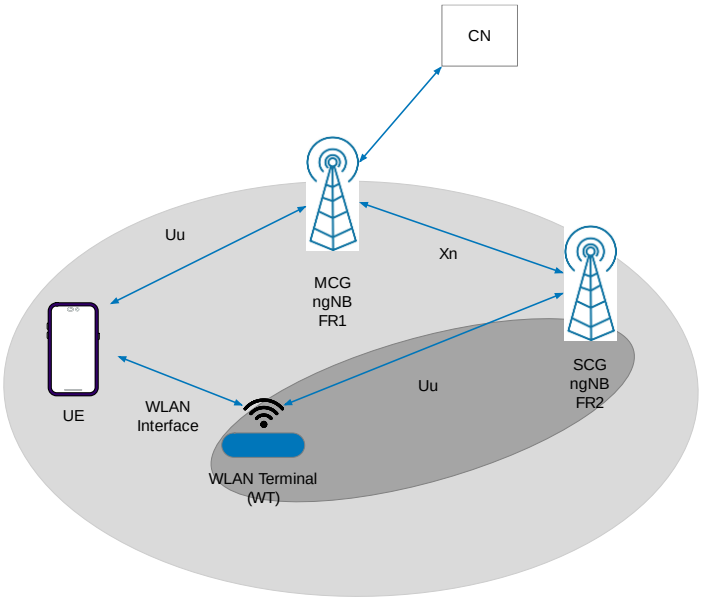


Figure 5-7 WLAN-Cellular Aggregation (WCA) where the UE and the WLAN terminal are connected to different BSs with different frequency ranges.

Focusing on integration in the RAN and using WLAN as an example of a different access network, one of the main motivations for an integrated WLAN and 6G solution is the indoor use case (e.g., in-home), where a trusted WLAN Terminal (WT) is within cellular coverage, which could be leveraged to support the UE. Both the UE and the WT could have a wireless connection with the BS, while the UE and the WT also have a wireless connection with each other. The WT-BS link may use a different FR than the UE-BS link, therefore different resources may be assigned to the UE and to the WT. An example of such a system is depicted in Figure 5-7. The proposed WLAN-Cellular Aggregation (WCA) framework can aggregate the direct UE-BS path as well as the UE-WT-BS paths on RAN level at the BS where the radio bearer originated from. Another use case of interest is when a UE has no cellular coverage (i.e., remote UE), but is within WiFi range of a WT, which in turn is within cellular coverage and may act as a relay for the remote UE. The current and predicted usage of various QoS flows and Radio Bearers (RB) necessitate the exploration of how this framework would be integrated in a WT. The BS may configure the UE to transmit the same packet (i.e., packet duplication) or different packets (i.e., packet split) over the UE-BS and the UE-WT-BS paths. Packet duplication would increase the reliability of the connection, while packet split increases the throughput. Aggregation takes place below the IP layer, which means that the WT does not have an IP connection with the cellular network for the sake of relaying the UE's packets from/to the cellular network, hence ensuring privacy. In addition, the UE would only utilize the trusted WT and involve the cellular network, only when the UE decides to enable this feature.

There are benefits for the cellular NW to share radio resources between the WT and the UE instead of allocating all radio resources to the UE. For example, the WT may have more powerful cellular capabilities than the UE, it may be connected to power, and it may be more stationary, allowing for a better connection in higher frequency ranges, where beam management is required. At the same time, the UE could save more power by offloading data to the WT instead of directly transmitting it to the BS. Note that data transmitted via a public

(i.e., not trusted) WT shall be protected via cellular security. Nevertheless, it should be investigated if and how the UE may save power when transmitting data only via a private (i.e., trusted) WT. At the same time, the UE-BS link would still be required for achieving higher reliability via packet duplication and for controlling and configuring control plane procedures such as mobility and WCA-related configurations. The LTE-WLAN Aggregation (LWA) [36.300] and the LTE-WLAN Radio Level Integration using IPSec Tunnel (LWIP) [36.300] features aggregated and integrated, respectively, LTE with WLAN. In the next deliverable, more details on WCA will be provided, as well as a comparison with LWA and LWIP.

Shifting the focus to integrating different access networks in the transport layer, the concept of a dynamic federation of loosely coupled domains (technological or administrative ones) in the context of 6G is introduced. The proposed approach uses network virtualization and programmability to dynamically connect self-managed and self-operated domains with embedded intelligence that supports this operation. The main idea lies in using inter-domain gateways to provide uniform domain interactions. The approach is an evolution of the 6G-LEGO concept [KTK+21]. The main idea is to use abstracted protocols (intent-like) between different networking domains. To use this approach, each domain has CP and UP gateways responsible for translating the abstracted messages into domain-specific ones. Each domain service and respective interacting protocols must be defined. In addition, a set of functions responsible for inter-domain level operations, i.e., discovering, adding, or removing a domain, has to be specified to implement the approach.

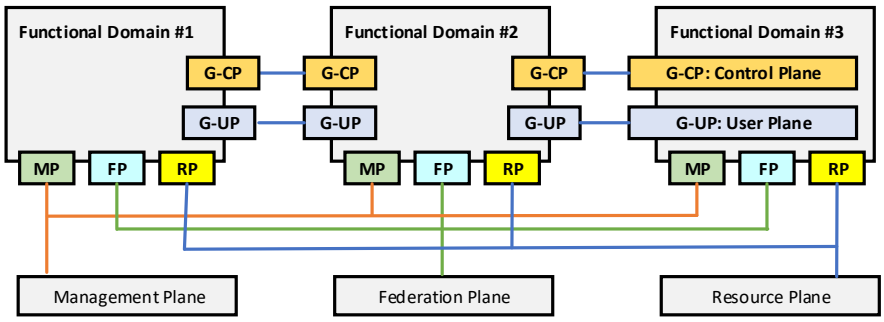


Figure 5-8 Simplified view of the dynamic federation of loosely coupled domains concept.

The main features of the proposed framework, outlined in Figure 5-8, are described below. More details can be found in Section 11.3.1.3.

- The solution is composed of autonomous modules called Functional Domain (FD). An FD can be a complete networking solution, a network segment or a set of networking functions that can be used for building a complete networking solution, e.g., WiFi or any other type of network.
- FDs can be dynamically federated, creating a Federation of Functional Domains (FDF). Depending on the goal, FDs can be aggregated (federation of FDs of the same technology), integrated (federation of FDs of different technologies) and chained.
- The interactions between FDs use high-level abstractions (intents and KPIs) to enable flexible FDF reconfiguration and to avoid complex end-to-end protocols.
- The proposed framework uses the UP, CP, Management Plane (MP), Resource Plane (RP) and Federation Plane (FP). A set of high-level services should be defined for each plane, as the Federation of FDs is made on each plane and its services level. FD does not have to have all planes, similarly to the concept presented in [KTO+18]. The FP is a concept-specific plane used for federation-related operations only.
- The framework includes framework-specific functions (FD internal functions, Federation Dedicated Functions and System Common Functions), plane-dedicated functions, and message buses used for interactions between FDs. Such an approach simplifies intra-FD and inter-FD interactions and enables the easy addition of necessary functions to FD or FDF.

5.2.3 Evaluations

5.2.3.1 Initial evaluations for selected CA/DC scenarios

The aim with this study is to investigate in which scenarios it is beneficial to aggregate carriers from different cells. These scenarios should be realistic, using commonly used frequencies and deployments. Table 5-2 shows the most important parameters for the selected realistic scenario (Scenario 2) and a reference scenario, i.e., Scenario 1. Both scenarios employ a low-band and a mid-band, the only difference is the bandwidth of the mid-band frequency.

Table 5-2 MC simulation parameters for the scenario 1 and 2

	Scenario 1 – reference scenario		Scenario 2 – realistic scenario	
NB	Low-band	Mid-band	Low-band	Mid-band
Carrier frequency	800 MHz	3500 MHz	800 MHz	3500 MHz
Bandwidth	10 MHz	10 MHz	10 MHz	100 MHz
Tx/Rx antennas in gNB	2	16	2	16
FDD or TDD	FDD	TDD	FDD	TDD

For the realistic deployment, the mid-band is 100 MHz compared to 10 MHz for the low-band. The main reason for this is that the low-band (below 1 GHz) is a very scarce resource, and an operator seldom has access to more than 10 MHz. The mid-band frequencies are less scarce, so in this case it is common that an operator gets around 100 MHz. For both the reference and the realistic scenarios, the mid-band has 16 antennas compared to 2 in the low-band, which to some extent mitigates the worse propagation aspects of the mid-band.

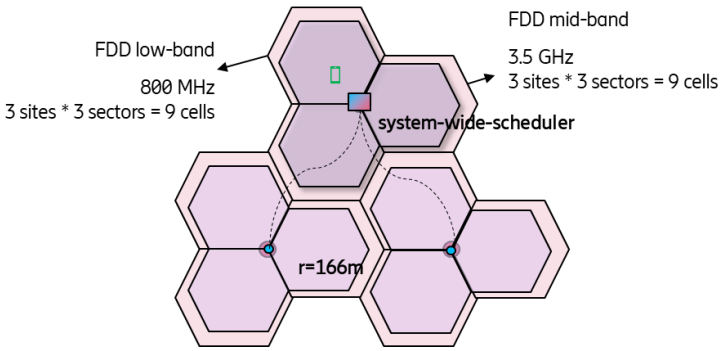


Figure 5-9 Deployment of the simulation: the low-band and the mid-band are using the same system-wide scheduler (gNodeB (gNB)), and a user can be connected to both the low and mid-band using aggregated bandwidths.

The simulations employ a system-wide scheduler, see Figure 5-9. This means that any cell that fulfils the requirements can be aggregated for a UE. This represents a centralized-RAN (C-RAN) deployment where a centralized baseband serves many cells (i.e., similar to DC but without any backhaul delay between the cells). However, maximum of 2 cells (carriers) are aggregated here. The condition to being added is that the RSRP must exceed -110 dBm. In practise, only cells that are co-located fulfil this requirement. The simulation deployment consists of 3 sites, each with 2 carriers and 3 cells. The cell radius is 166 m and the traffic is 10 Mbyte FTP download. The users select the mid-band carrier at cell selection if the RSRP is larger than -110 dBm.

Figure 5-10 shows the user's object bit rate (i.e., the user bitrate during actual downloading of the packet) for carrier aggregation on and off, for the reference and realistic scenarios, respectively. The results show that there is a clear gain for the reference case, almost 100% user bit-rate increase.

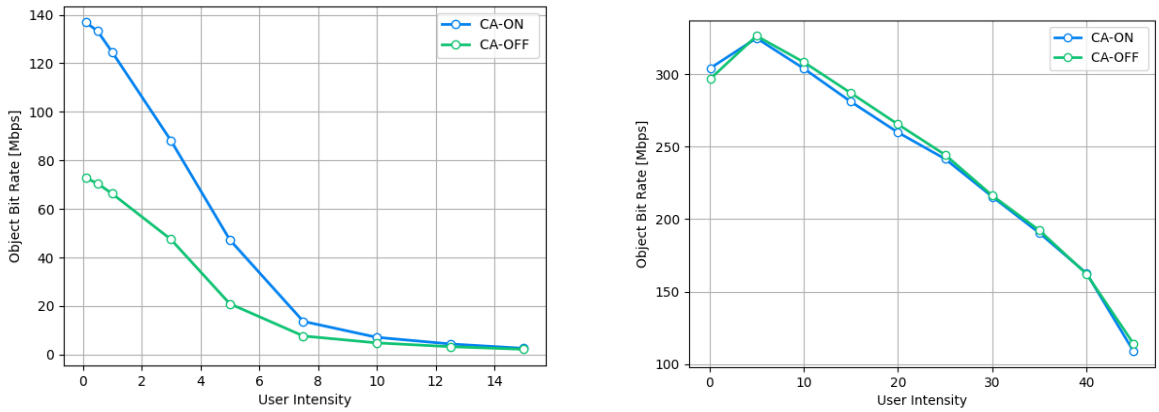


Figure 5-10 Simulation results for reference and realistic scenario for carrier aggregation on and off. The figures show the user bit-rate vs. the user intensity (mean number of users entering the system per second).

However, for the realistic scenario, there is no gain at all. There are two reasons for this. The first reason is that the users start the FTP downloading session in the mid-band, with 100 MHz. After some initial measurements, the user also notices that there is another (low-band) carrier. If the new low-band carrier has an RSRP higher than -110 dBm, the network will then also add the low-band carrier after some delay. However, by then the 10 Mbyte packet has already been downloaded to a large extent. The second reason is that adding 10 MHz to the 100 MHz will not increase the download throughput more than 10%.

In conclusion, aggregated carriers are only beneficial if their bandwidths are relatively equal i.e., when the throughput is relatively equal from both carriers. For small packets, the gain of aggregation can be low even with equal bandwidth of the different carriers since the master cell probably has already finished the session before the secondary cell is added.

5.2.3.2 Multi-server offloading scenario evaluation

Capitalizing on the ability to simultaneously access multiple servers in a multi-server MEC system, the application of the novel RSMA technique to facilitate the concurrent user transmissions, i.e., offloading, to multiple MEC servers is studied. The users' computation tasks are partitioned into subtasks that are fully offloaded to a combination of the different available MEC servers for remote processing. The transmission/offloading delay of each user to each offloading MEC server is determined by the ratio of offloaded bits to the achieved data rate. Additionally, the processing delay for a user's offloaded task at an MEC server is calculated as the ratio of the CPU cycles to be processed to the allocated computing power, measured in CPU cycles/s. Consequently, the total delay of a user at an MEC server is a function of the user's computation task assignment ratio to the respective MEC server, uplink transmission power and data rate, and the allocated computing power by the MEC server. The objective is to minimize the sum of users' maximum experienced delay resulting from task offloading and processing across the different MEC servers by jointly optimizing the latter variables and parameters, i.e., the users' computation task offloading assignment ratios to the different MEC servers and their allocated rates, uplink transmission powers, and computing resources related to the corresponding MEC server. The formulated min-max-sum optimization problem is initially equivalently transformed, and the Karush-Kuhn-Tucker conditions [BV04] are applied to decompose it into two independent subproblems. In this way, suboptimal and optimal solutions for the radio and the computing resource allocation derive, respectively.

In the following, indicative numerical results are provided from the preliminary evaluation of the proposed multi-server offloading scenario. For the scope of the evaluation, consider N users randomly and uniformly distributed in a square area of size 150×150 m. The users concurrently offload part of their computation tasks to M MEC servers spatially uniformly located within this area. The channel gain between a user and a MEC server is calculated based on the distance-based path loss model $\mathrm{PL} = 128.1 + 37.6 \log_{10}(d)$, with d measured in km, while the standard deviation of the shadow fading is 4 dB [TP19], [YCS+21]. The total system bandwidth considered is equal to 20 MHz, the users' maximum transmission power level in the uplink is 24 dBm, and the minimum data rate requirement of each wireless link is as 1 Mbps. The computation task

size of the users is characterized by a maximum amount of CPU cycles to executed equal to 80 Mega-CPU cycles, while the computing power of the MEC servers is defined equal to 5 GHz for all of them, i.e., the MEC servers are of equal computing capabilities. The presented simulation results have been averaged over 3000 different channel realizations based on the log-distance path loss model.

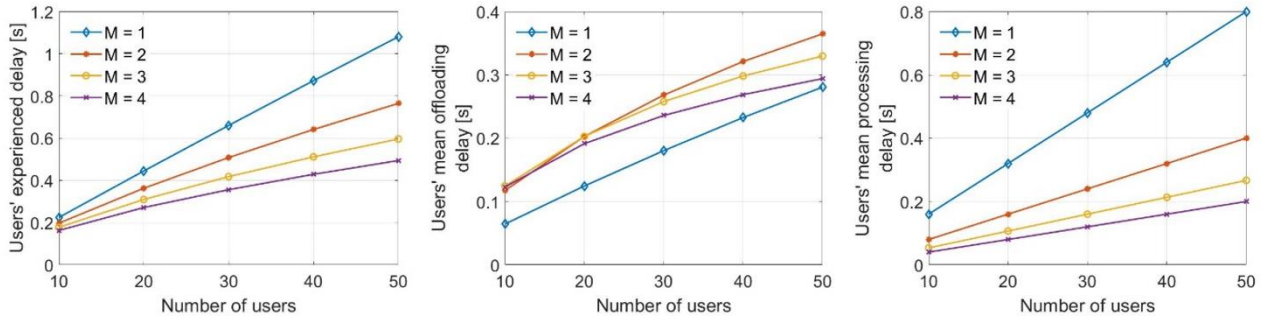


Figure 5-11 Evaluation of multi-server offloading in terms of the users' (a) overall experienced delay, (b) mean offloading delay, and (c) mean processing delay.

Figure 5-11 depicts the mean values of the users' (a) overall experienced delays, (b) mean offloading delays to the different MEC servers, and (c) mean processing delays at the different MEC servers. Different numbers of users and MEC servers are scrutinized in the network, indicated in the horizontal axis and by using different graph colours, respectively. Apparently, considering a fixed number of MEC servers, the higher the number of users existing in the system, then the higher their overall experienced delay is. The inverse behaviour is observed as the number of MEC servers increases, given a fixed number of users. The rationale behind this operation is as follows. Considering that the MEC servers are spatially uniformly distributed within the square area, the denser this area becomes with the number of MEC servers increasing from 2 to 4, the users are getting closer to the MEC servers and achieve higher data rates, while the overall system's computing capacity increases at the same time. For this reason, both the users' mean offloading and processing delay at the different MEC servers decrease. Concerning the special case, when a single MEC server exists in the system, a lower offloading delay is achieved compared to the multi-server MEC case since the users utilize the whole available bandwidth to them without splitting it to accommodate multiple concurrent transmissions.

Nevertheless, given the high computing demands of the users, a significantly higher processing delay is caused in the single-server case than the multi-server case, which dominates and results in low overall experienced delays for the users. This highlights the superiority and benefits of multi-server to single-server MEC offloading in dense and high-computing demands systems.

5.2.4 Summary

Multi-connectivity could simultaneously increase the system throughput, as well as the system robustness and reliability. Aggregating different access networks, such as WLAN, would enhance coverage while still providing the increased reliability of having multiple OTA connections. The implications of making this enabler operational lie on the RAN layer in the case of the CA/DC evolution, where novel implementations will be required for both devices and base stations. As far as achieving multi-connectivity with different access networks is concerned, different layers are affected, depending on where the adopted solution will aggregate the data packets that are transferred via these access networks. This mapping is depicted in Figure 5-1. Table 5-3 summarizes the main benefits and implications of the multi-connectivity enabler.

The use case families "fully connected world" and "immersive experience" [HEX223-D12] may be enabled by multi-connectivity. Fully connected world will require high service availability (i.e., percentage of time the service can be delivered) and reliability (i.e., success of transmission), while immersive experience will also require higher data rate. High service availability and reliability can be achieved via multi-connectivity with different access networks and the evolution of CA/DC with packet duplication, respectively. Aggregating carriers via CA/DC are only beneficial if the throughput is relatively equal from both carriers (i.e., similar bandwidths and coverage). However, higher reliability may still be achieved even with unequal carriers.

Higher throughput may be achieved with the use of CA/DC as well as via multi-connectivity with different access networks (e.g., Wi-Fi and 6G), by allocating the data packets to the carriers and/or paths with the highest throughput.

Table 5-3 Benefits and implications of "Multi-connectivity" enabler.

Description	Multi-connectivity enables simultaneous connection to different carriers, which may belong to physically separated nodes, different radio access technologies or access networks	
Benefits	**KPI improvement**	Increased user and system data rate. The CA and DC throughput gains depend on the differences between the carriers to be aggregated. For example, if the pathloss difference (due to e.g., frequency) is high, the gain is low since the UE will seldom use the cell (carrier) with bad pathloss. Similarly, if the bandwidth difference is large between the cells, there is small gain to aggregate the cells. Aim to increase robustness and reliability, by allowing control procedures to be performed via different paths, such as secondary carriers or different access networks. Packet duplication over the different OTA connections would also increase the reliability. More efficient management of network resources, since the system would be able to determine which of the connected computed nodes should serve the UE and which of the OTA connections should be used at a point in time based on the service's QoS requirements.
	Design principles [HEX223-D21]	Increased coverage by using different radio access and the increased reliability having different paths to the NW both improve the resilience and availability (#5 design principle in [HEX223-D21]).
	Dependencies / Basis for another enabler	Mobility procedures should take the multi-connectivity proposals into consideration
Implications	**Requirements**	Depending on the CA/DC solution, new interfaces and protocols between nodes may be needed, which may lead to an increased complexity in coordinating different NW nodes. Faster addition of cells compared to 5G is needed. To increase the robustness, there is a need for a more flexible use of the UL so that the secondary cell may take over the role of control signaling in the UL. New mechanisms and procedures for aggregation of different radio technologies, such as 6G cellular and WLAN, should be defined, which would impact the layers where the aggregation takes place, e.g., 6G RAN, transport.
	Standard relations & regulations	RRC protocol modifications and procedures for CA/DC evolution could be defined in 3GPP RAN2 [37.340][38.331]. RAN protocols for aggregation with other RANs could also be defined in 3GPP RAN2.
	Required resources	For offloading computation to different network nodes, both computation and communications resources are required.

5.3 E2E context awareness management

5.3.1 Introduction

The previous deliverable [HEX223-D32] stated that E2E context awareness management enables each network and compute component to dynamically adapt to the context to ensure the expected E2E QoS for the services and the expected QoE for the users. To reach such target it is relevant to leverage on effective automation and orchestration mechanisms to facilitate the interaction among such components. Both the underlying networks

and the edge computing layer's infrastructure may be dynamically adapted by using the mechanisms included in this enabler. Mission-critical operations can be enhanced to reduce the network overhead and allocate edge and device resources flexibly, ultimately improving system performance. This allows for multiple edge allocations and RAN slices.

5.3.2 Architectural implications

5.3.2.1 Transport network abstraction

The context-aware paradigm pertains to the ability of a transport network to dynamically adapt its connectivity to support a variety of services, each with specific E2E QoS characteristics outlined in their respective SLAs. This paradigm is particularly applicable in scenarios with strict requirements, such as availability. The primary aim of context-awareness is to optimize infrastructure resource usage by minimizing over-provisioning.

In this context, efficiently designing a UP to avoid traffic concentration points and simultaneously handle mobility and QoS of user sessions is crucial. The IP communication relies solely on locators (host interface addresses) that are also used as node/service identifiers at the network layer. The approach makes IP mobility management troublesome, and to solve the problem, traffic anchors and tunnels have been introduced to handle mobility while preserving the identifier exposed to the transport layer. The existing solutions use the GTP protocol, which involves certain overhead, causes traffic aggregation, and leads to ineffective traffic steering because the payload data needs to leave the tunnel before it can be redirected. Software Defined Network (SDN) can replace IP/GTP transport, offering easy traffic flow redirection by using IP headers with source/destination addresses as labels and native mobility support without anchors, encapsulation overhead, and need for tunnelling.

As a result, to implement context-aware transport, a resource orchestrator creates an abstracted view of transport resources and employs an SDN transport controller for resource management, ensuring that the QoS associated with a slice is met. It also carries out E2E admission control to maintain the expected QoS for both active and incoming services. The E2E service orchestrator places all network functions on the abstract view to guarantee the QoS of the targeted slice.

Figure 5-12 depicts an abstraction scenario where internal connectivity of the transport network is hidden through the exposition of logical links between the edge nodes. Each logical link represents a portion of the network's capabilities, offering an aggregated view of parameters, such as bandwidth, latency, resiliency level, etc. In this way, network resources are exposed through relative service parameters while keeping many resource details, as the number of channels, physical properties, and actual topology, hidden. Different abstraction depths present different degrees of technical information to higher levels. The lowest abstraction depth can contain raw, detailed data, and specific information about complete network topology, hardware, protocols, and configurations. A highest depth may represent network elements more generally, with most intermediate nodes hidden (present only "border nodes"), focusing mainly on aggregated network capabilities.

The SDN Domain Controller (SDNDC) provides an abstracted SDN domain view and executes Guarded Command Language (GCL) commands concerning path establishment in its domain. It is also responsible for discovering domain topology and its changes, monitoring local links/paths and programming flow handling, i.e., configuring flow forwarding rules. The SDN controller interacts with switches using the OpenFlow protocol for that purpose. Each SDN domain exposes information to GCL about its local paths, transit paths, and flows. The Border Nodes act as inter-domain gateways. The path between the source host and the destination host is chosen based on the inter-domain paths metric. An example showing the inter-domain and intra-domain connections in the form of a graph is presented in Figure 5-13.

Overall, abstraction offers scalability, allowing the transport layer to grow and adapt to changing demand while maintaining stability with RAN/CN. It also enables easy reconfiguration and optimization of transport resources without affecting upper layers, while improved reliability comes through isolating and independently managing network components. Lastly, abstraction simplifies network management, reducing operational costs by providing a streamlined view of transport resources to centralized systems. However, the challenge lies in maintaining the appropriate level of information for higher levels, i.e., the orchestrator. Providing too much technical detail can overload higher levels and hinder network operations, while excessive abstraction can lead to the loss of crucial details needed for troubleshooting [LC15].

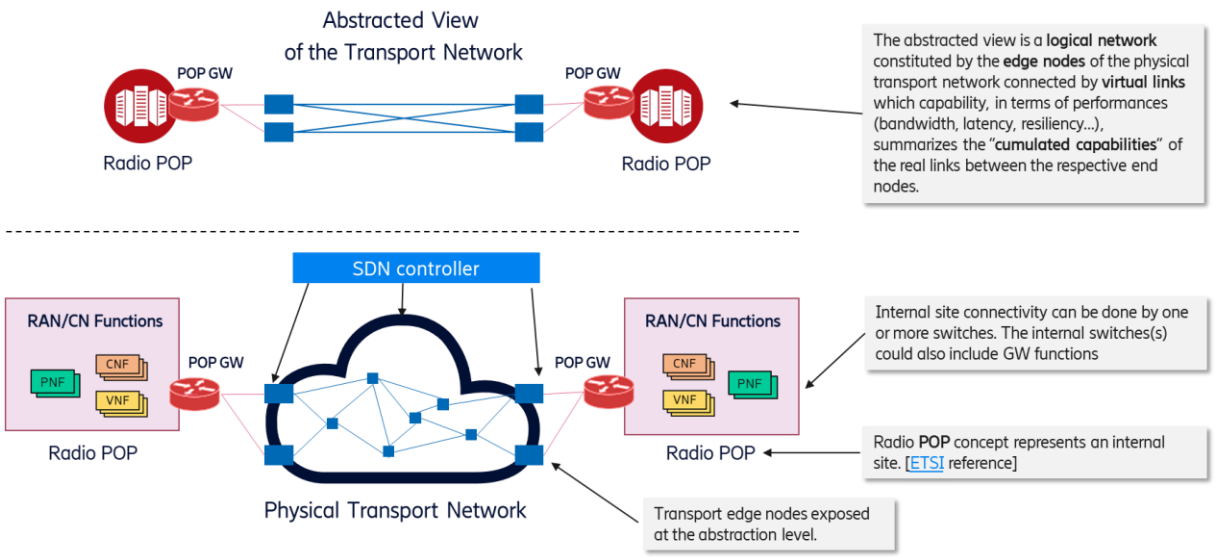


Figure 5-12 Example of physical transport network connecting two sites of RAN/CN with related abstraction.

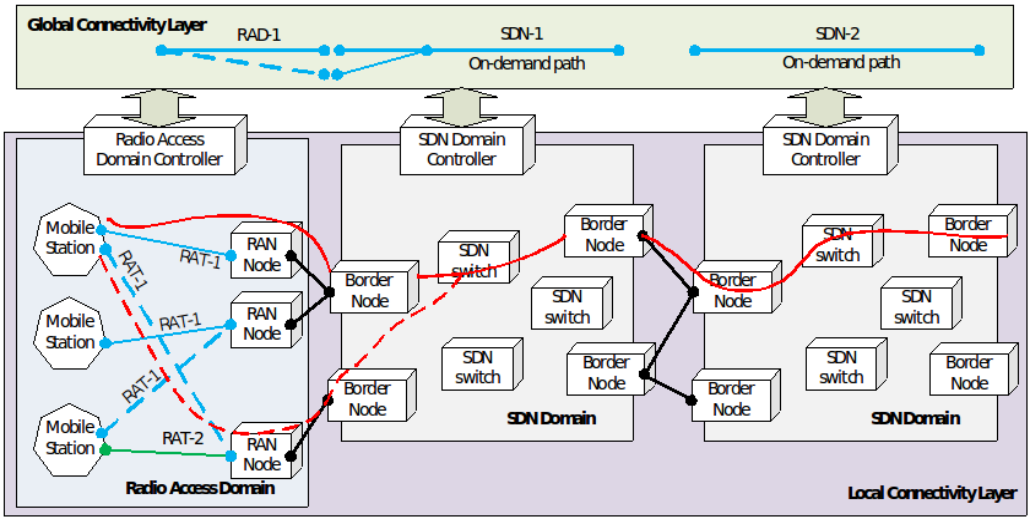


Figure 5-13 Proposed multi-domain SDN architecture.

The following macro-steps can be considered for the abstraction procedure:

- **Step 1- The global initial abstraction creation** is based on available information and serves as the initial step in the network configuration phase. This process relies on the traffic data available at step 0. The deeper the understanding of traffic forecasting, the more accurately this initial abstraction can match traffic behaviours over time.
- **Step 2 - Local "ad-hoc" abstraction updates** are triggered by actual request or traffic flows while traffic is in progress, and the corresponding network resources are being configured. This allows the network to dynamically adapt and optimize based on real-time traffic patterns and resource utilization.
- **Step 3 - Global periodic abstraction updates** are triggered by specific policies or criteria (e.g., periodically, when a basket of abstracted links reaches a certain threshold of available paths, etc.). The baskets are modified by removing certain paths that are not being utilized, while other baskets may be created or updated with new paths. This operation occurs asynchronously with respect to local updates and, in most cases, does not trigger local updates. This allows the network to maintain flexibility and adaptability in response to changing traffic conditions and resource requirements.

Further details of these steps are provided in Section 11.3.2.1, along with an illustrative example.

5.3.2.2 Semantic end-to-end system optimization

Flexibility and semantic context of the applications are crucial to minimize network congestion and optimize end-to-end performance. The concept of Semantic RAN introduces the idea of flexible application allocation, allowing applications to be processed either locally on the user equipment, offloaded to the edge server, or a combination of both. This leads to the development of an orchestration algorithm that optimizes offloading policies, radio, and compute resources jointly to meet performance requirements, considering factors such as data compression, accuracy, network latency, battery life, and time-sensitive resource allocation to meet end-to-end latency bounds. An example of such applications is the collaborative mobile robots use case, where the semantic context of the robotic task can be used in order to meet the strength QoS/QoE requirements while optimizing the available resources in the end-to-end system.

The architectural modifications for having such a semantic solution includes integrating two main components in the ORAN architecture: the Semantic Deep Learning Analyzer (SDLA) and the Semantic Edge Slicing Module (SESM). The SDLA enhances network resource allocation by learning and interpreting semantic descriptions of application tasks. Meanwhile, the SESM solves the slicing algorithm, which allows for managing the edge and radio slicing at the edge and choosing the best fitting offloading policy. To ensure the proper functioning of these components, the radio, edge, and device status is monitored. The radio status involves measurements used to calculate the latency, e.g., signal strength or interference levels, the edge status regards context about the available resources to the SESM, e.g., CPU and memory usage, or computational load, and the device status checks the battery levels, current workload, and sensor information to select the optimal offloading policy.

In the following, the O-RAN standard architecture is covered, exploring, and showcasing how these two components should be integrated to ensure efficient performance of mobile robot applications. A similar study regarding the ETSI-MEC architecture can be found in Section 11.3.2.2, along with a special case focused on the harmonization of ETSI MEC with the 3GPP-5G architecture.

The integration of the SDLA and SESM components into the ORAN architecture is presented in Figure 5-14, while the specific steps that take place are as follows:

1. **Petition for an instantiation of a robot task.** The corresponding petition comprises a task descriptor of the robot task, which includes the deep learning model to be executed and its execution requirements, such as latency and accuracy, using an O-RAN Slice Request (OSR). The OSR is sent via a Human Machine interface from a Virtual Network Operator (VNO).
2. **Semantic Analysis of the task.** The SDLA uses the Non-Real-Time RAN Intelligent Controller (RIC) infrastructure to compute the corresponding latency and accuracy functions with the information gathered from the Radio Status (via O1 interface) and the task descriptor. These functions are then shared with the SESM (through A1 interface) at the Near-Real-Time RIC.
3. **Semantic Edge Slicing of the task.** The SESM determines how to accomplish the robot task by choosing between all available policies and identifying the necessary slicing requirements for radio and compute resources. For this purpose, the SESM uses as inputs the latency, accuracy, and battery function, the requirements of the tasks, and the metrics gathered from the Radio, Edge, and Robot status services (using interface E2). The computation and radio slicing are shared with RAN Edge and the CU using E2 interface. Additionally, if the task is offloaded, the Policy and a Compression Factor for data stream are shared with the VNO, which then communicates them to the robots.

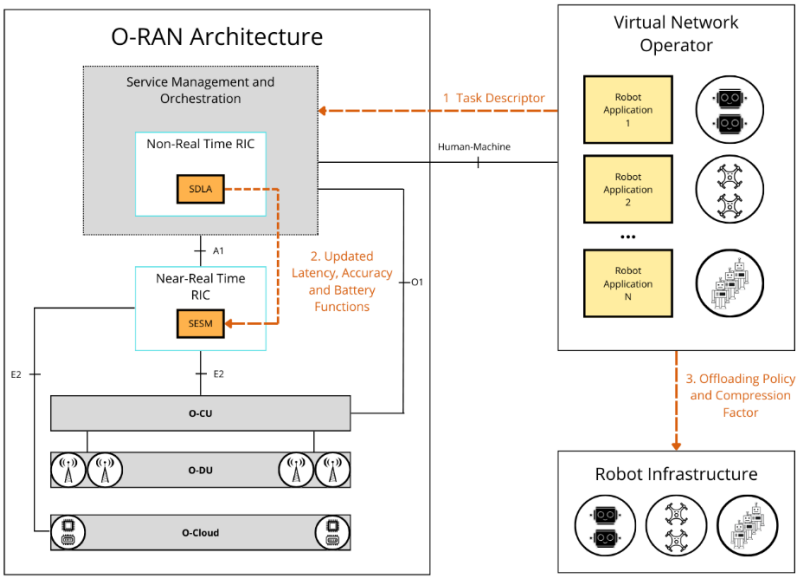


Figure 5-14 Integration of the SDLA and the SESM components in the O-RAN Architecture

5.3.3 Evaluations

5.3.3.1 Multi-domain SDN scalability evaluation

The split of the SDN transport into multiple domains improves SDN scalability and makes the problem of the SDN controller placement less critical. However, such an approach raises inter-domain issues. In the multi-SDN approach, path calculation (algorithmic delay) is shorter due to a smaller number of domain switches compared to the number of all switches in the network and path execution (deployment) – can be done in parallel in each SDN domain. The path computation time can be further improved using efficient path computation algorithms (see complexity details in Section 11.3.2.3) at domain and network levels.

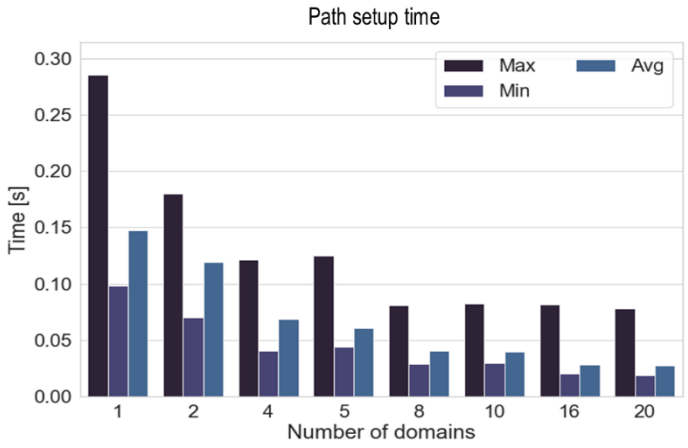


Figure 5-15 Path setup time dependency on the number of SDN domains

To evaluate the improvement caused by the multi-domain SDN, simulations have been carried out concerning path setup time (i.e., path calculation and path deployment) for the same number of nodes but different numbers of SDN domains. In the experiment, 80 OpenFlow switches were divided into 1 to 20 equally sized domains. The network had 118 links and 20 hosts. The basic Dijkstra algorithm was used for route computation, and 100 x 0.5 Mbps data flows were generated using *iperf3*.

As Figure 5-15 shows, the multi-domain approach provides a shorter path setup than a single-domain approach - in the case of 4 domains (20 switches per domain), the gain is about 60%. In the case of 16 domains (5 switches per domain), the average path setup time is about five times shorter than in the case of a single domain.

Further reduction of domain size due to the small number of switches provides no significant improvement. The results show that the multi-SDN approach shortens the path setup time.

5.3.3.2 Autonomous robots

As described in [HEX223-D32], the Semantic RAN concept is built upon the following three assumptions: (a) different Compression Factors can be used for efficient data transmission without sacrificing the performance of the task, (b) different offloading policies can meet the accuracy and latency bounds while considering the robot status, and (c) different Radio Resource Block configurations can meet the latency bounds.

Figure 5-16 (a) Compression Factor of 100 using YOLOX-Tiny

Figure 5-16 (b) Compression Factor of 1 using YOLOX-Tiny

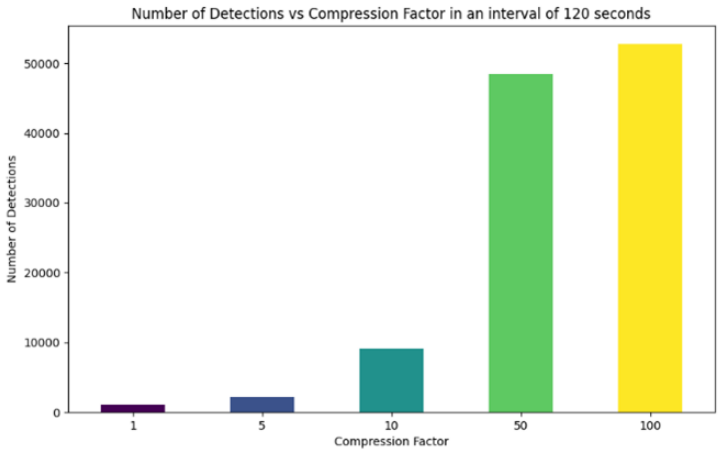


Figure 5-17 Number of detections per compression factor using YOLOX-S

To validate the latter assumptions, an object detection experiment is performed using two YOLOX [YOL+21] models: YOLOX-S and YOLOX-Tiny. The YOLOX-S is a larger and more detailed model, designated to operate in a server equipped with a GPU, representing a high-capacity computational scenario. In contrast, the YOLOX-Tiny is a more compact model that runs on a laptop without a dedicated GPU, simulating a more resource-constrained environment as the case of a robot. Both models were connected to a camera. This input video flow is modulated using a Compression Factor. Adjusting this value from 100 down to 1, allows to gradually decrease the camera resolution. The same evaluation is performed in two different contexts: in a laboratory full of different classes to be inferred and in an empty corridor, where typically there are no classes to be inferred, rather than people sometimes crossing by. A more detailed explanation of the experimental analysis can be found in Section 11.3.2.4. In the following, the first two assumptions are covered, while the Radio Resource assumption will be considered in future work.

In Figure 5-16 (a) and Figure 5-16 (b), different values for the Compression Factor are examined. As can be seen, the YOLOX-Tiny model with a large Compression Factor, effectively distinguished little objects like a cup and bigger ones such as a person. In contrast, when the Compression Factor was lowered to 1, the model failed to detect the cup but still identified the person. It is noteworthy that the different classes detected depend not only on the Compression Factor but also on the model itself, the training dataset, and the distance from the object. In this experiment, the model was trained with the COCO Dataset, which contains many person samples, and the picture was taken no more than a meter away from the inferred objects. It is therefore validated that different Compression Factors can lead to similar accuracy bounds without sacrificing the performance of the task.

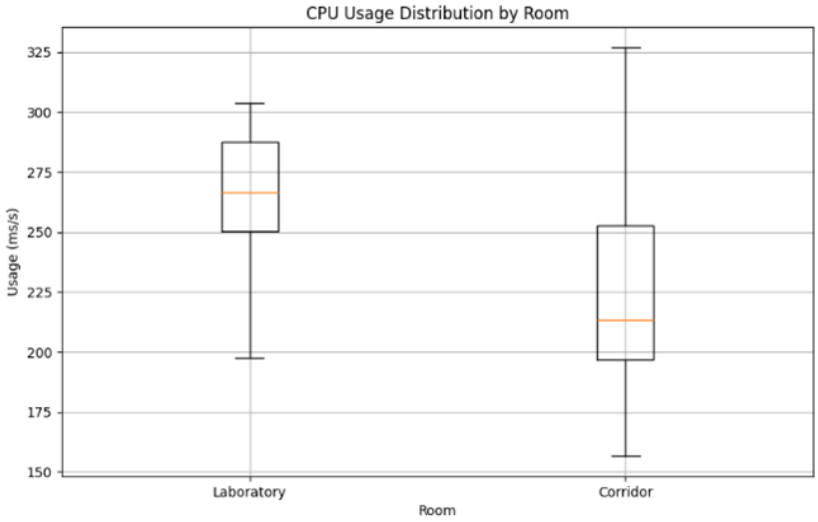


Figure 5-18 CPU Usage under the two different contexts

Figure 5-17 demonstrates that for different Compression Factor values, the model can detect a varied number of objects, which decreases drastically from 50 down to 1. In this experiment, the YOLOX-S model was used, but the case is the same happens if the YOLOX-Tiny model is selected. Figure 5-18 shows the CPU Usage at the robot laptop for the two different contexts, as previously described. The box plots lead to the conclusion that in a room full of objects, there will be higher CPU Usage, which will result in more battery consumption for the robot. Thus, it is confirmed that different offloading strategies must be considered to achieve the accuracy and latency bounds while being energy efficient. In contexts where a detailed inference is needed, the semantic RAN should choose the most appropriate model for the use case.

5.3.3.3 Delayed computing

Consider a network architecture where multiple computing layers—namely, edge, fog, and cloud—collaborate seamlessly to provide transparent computing services to subscribed users. The computing layers differ in computational power, resulting in different response times for the subscribed users. In this context, the variety of tasks and user application requirements create an opportunity to employ different computing options. Motivated by the latter and aiming to optimize resource utilization, the delayed computing paradigm suggests users leverage their delay tolerance and price sensitivity. This flexibility enables intelligent task orchestration across the computing continuum, reducing service costs or subscription fees through incentive mechanisms.

In the studied delayed computing scenario, incompleteness of information arises from the a priori unknown characteristics of the offloaded task to the service provider, specifically in terms of computing intensity and service delay requirements. To address this issue, contract-theoretic modelling can be adopted for effective incentive mechanism design. Specifically, Contract Theory [BD04], belonging to the domain of Labor Economics, provides the theoretical foundations to construct mutually agreeable contracts or arrangements among economic players in the presence of incomplete information between them. Following the principles of contract-theoretic modelling, the service provider utilizes existing datasets about the potential characteristics of the offloaded tasks and distinguishes the users into different types. Then, based on this statistical knowledge, the service provider designs a menu of contracts tailored to each prospective type. The contract comprises a subscription fee to the service and a corresponding response time approximation. Subsequently, each user

autonomously selects the one contract out of the menu that best fits its type. By intelligently managing the trade-off between payment and service response time, users can be incentivized to choose a less restrictive contract in terms of response time, allowing for the network's flexibility to orchestrate across the continuum.

The preferences and motivations of both the service provider and users during the design and selection of contracts, respectively, are captured by a utility function. A common representation of the utility function is the difference between gain and cost. The service provider's gain can quantify resource utilization efficiency across the network, while the cost can be associated with the subscription discount offered to users. Conversely, the monetary reward can represent users' gain, with the cost corresponding to the experienced service delay. To determine the menu of contracts, the service provider solves an optimization problem to maximize its utility while securing the users' participation in the contract, i.e., by guaranteeing their non-negative utility.

5.3.3.4 Context-aware connectivity for maritime ports

This section encapsulates key findings from a study on enhancing connectivity for smart maritime ports. Data can be harmonised/standardised in various aspects of port operations, such as traffic flow, cargo handling, and supply chain management. The integration of 6G and 5G as well as Wi-Fi technologies can ensure the medium for the exchange of information, and the use of IoT can integrate other technologies that will allow an answer to other challenges for port management. Protocols like TSN can help combine different types of traffic in the same network, making best-effort networks more predictable.

The combination of TSN and AI is studied to perform predictive maintenance of the network, anticipating possible faults and degradations, taking into consideration the historical context and status of the network. This can be an important tool for enhancing the performance and competitiveness of the port networks, allowing them to better meet the needs of shippers, carriers, and other stakeholders in the global logistics industry. For the TSN network simulation, available simulators such as NS-3 will be used.

The demanding connectivity requirements from some applications in maritime ports, like real-time monitoring of vessel movements, automated cargo handling, and supply chain optimization, put additional stress on local networks covering the port location. Key performance indicators have been identified, such as AI/ML-related capabilities, E2E latency, and reliability to gauge the success of connectivity enhancements in maritime ports.

5.3.4 Summary

The implications of the mechanisms involved in this enabler are that different system components e.g., RAN, transport, management and orchestration modules should interact with each other and become aware of the context. This means that new signalling and synchronization is required. Moreover, the novel resource allocation and orchestration mechanisms should also be operational and effective even when incomplete or partial context awareness is available. The mapping of this enabler to the 6G system blueprint is illustrated in Figure 5-1. Table 5-4 summarizes the main benefits and implications of the E2E context awareness management enabler.

The use case families "cooperative robots" and "physical awareness" [HEX223-D12] may be enabled by E2E context awareness management. With the aim to improve both the applications and the communication's performance, cobots are expected to take advantage of contextual information (e.g., RAN measurements, delay requirements, computation resources). Based on the aforementioned context, appropriate task allocation, as well as communication and compute resource management may enable this use case family. Physical awareness would require context-aware communication in the form of path selection and would benefit from the transport network abstraction.

Table 5-4 Benefits and implications of "E2E context awareness management" enabler

Description	Mechanisms to allow network components to dynamically adapt to the context to ensure the expected E2E QoS.	
Benefits	**KPI improvement**	Reduction of the network overhead and flexible allocation of edge resources, ultimately improving the system performance by allowing multiple edge allocations and RAN slices. Optimal energy consumption of the end devices

	Design principles [HEX223-D21	Effective and optimized use of the network infrastructure resources, as well as personalized and dynamic resource allocation improves the flexibility to different network scenarios (#3 design principle in [HEX223-D21])
	Dependencies / Basis for another enabler	In the existence of sensitive datasets, privacy preserving techniques and split neural networks should be used to perform cross-network function distributed ML model training to fuse the context model with other network models without necessitating raw data transfer.
Implications	**Requirements**	Different network components e.g., RAN, CN, transport, should become aware of the context and need to interact, implying the need for signalling and synchronisation. A resource orchestrator would have to create an abstracted view of transport resources and to employ a software defined transport controller for resource management, ensuring that the required QoS is met. Effective resource allocation and orchestration mechanisms that operate even when incomplete or partial context awareness is available should be designed. In a mobile robotic context, diverse system elements must be exposed for the effective resource allocation and system orchestration. Thus, a correct resource exposure of the robot status should be ensured. Automatic translation of requirements from one component to another, according to the peculiar characteristics of each component would be required. In addition, harmonization of optimization mechanisms in each component is needed, to guarantee an efficient E2E interworking. Finally, a suitable abstraction and exposition of the capability of each component to facilitate the E2E managing of the network components according to the context is a prerequisite.
	Standard relations & regulations	In the context of task allocation in semantic RAN, the ETSI MEC architecture [MEC035], the harmonization with 3GPP [ETSI36] and the O-RAN architecture [OAD24][OSA24] may be impacted.
	Required resources	Cloud computing, edge computing and extreme edge devices

6 Architectural enablers for network beyond communications

As stated in [HEX223-D32], the evolution of the network is pushing the boundaries, beyond conventional connectivity, into accommodating and supporting novel services, expanding the network's scope by processing data, generating insights, and delivering added value from societal, innovation, and business perspectives. Examples of new services comprise sensing, enhanced localization and tracking, compute-as-a-service, etc [HEX223-D12]. In the initial analysis provided by [HEX223-D32], four enablers have been identified towards introducing Beyond Communications Services (BCS), namely Enabler #1 - Exposure and data management aspects; Enabler #2 - protocols, signalling and procedural aspects; Enabler #3 - Application- and device-driven optimisations, and Enabler #4 - Enhancements of BCS capabilities, such as JCAS. In the context of this chapter, those enablers are further analysed; the sub-chapters have followed the above structure with the Enabler #2 split into two sub-chapters to further elaborate on the two primary directions of this enablers, i.e., JCAS and Compute offloading services (into sub-sections 6.2 and 6.3 respectively, while Enablers #3 and #4 are presented under the common sub-section 6.4 in order to better illustrate the architectural implications).

This chapter examines the requirements, architectural implications, and foreseen benefits in relation to novel data exposure and management, protocols, signalling procedures, as well as network and application function placement towards service and QoE enhancement for users.

In order to provide further insights, each architectural enabler is mapped to the 6G E2E system blueprint of Figure 6-1 [HEX223-D22]. This mapping aims to suggest the implications of the proposed enablers being part of the E2E system, in terms of layer functionality and respective interfaces.

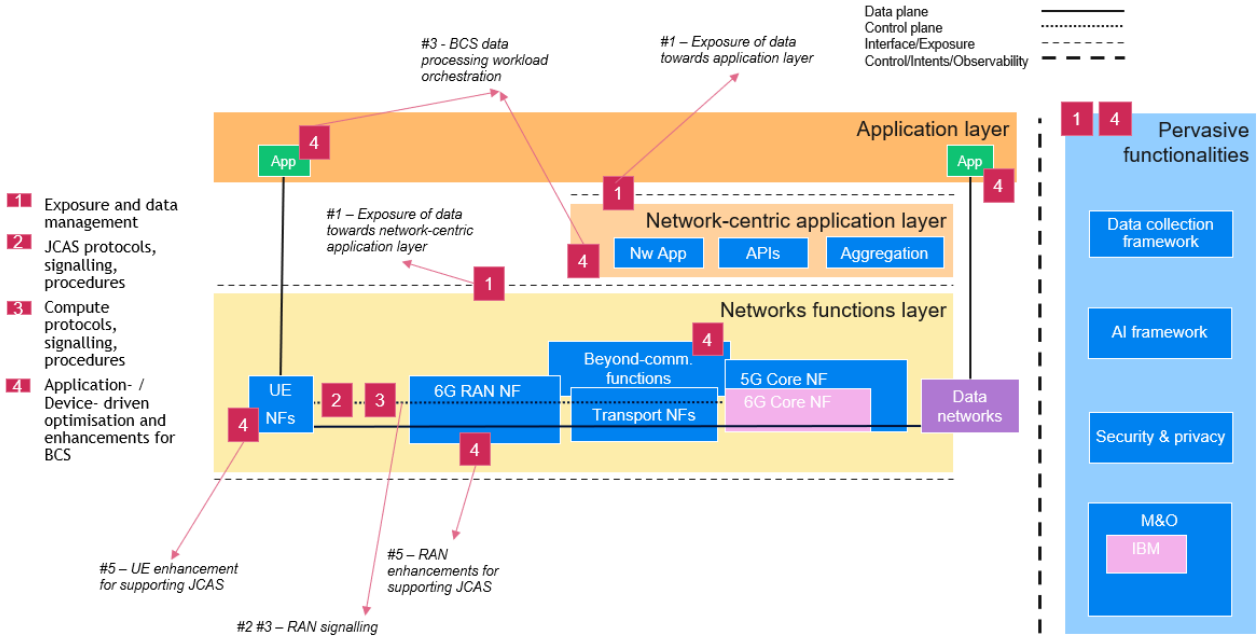


Figure 6-1 Beyond communication enablers potential mapping to the E2E System Blueprint

6.1 Exposure and data management

6.1.1 Introduction

As we move towards the establishment of the next generation networks, efficient data management and strategic exposure are fundamental for leveraging the network's capabilities and for optimizing network functionality and enhancing capabilities, with particular attention to the secure and efficient exposure of data. This is essential for handling the extensive information that may be generated by the network's diverse devices, comprising RAN infrastructure, user equipment, radio sensing nodes, or other IoT devices. Importantly, the integration of JCAS represents a critical evolution, differentiating by its dual-purpose technology that enhances both data transmission and environmental sensing, thus optimizing the network's operational efficiency and situational awareness. The focus of this enabler is on data generated both by the RAN infrastructure (radio

sensing-related data), as well as on application layer data generated by various devices such as wireless sensors, cameras, robotic platform modules, etc., along with the interfaces towards network functions or service components that may be residing in other segments of the network or compute continuum. The placement of such functions and components is thus critical, as numerous aspects related to data privacy, communication and processing latency, reliability and other critical KPIs are affected. Placement of the various application components, e.g., micro-services in (multi-domain) cloud-native environments, and/or PNFs/VNFs, comprising end-to-end (vertical) application functionality involves strategically positioning those components to optimize performance, balance network load, and ensure data privacy and security. This placement is connected with KPIs such as data throughput, reliability, and cost, which are vital for assessing the efficacy of the network's data management strategies.

6.1.2 Architectural implications

The architectural landscape of 6G will most probably be influenced by the exponential rise of IoT devices and the emergence of novel applications which leverage BCS capabilities, such as sensing, precision positioning, compute offload, etc. As the network evolves to accommodate a wider range of devices, from sensors to broadband devices, the challenges in such beyond communications data management and beyond communication services/data exposure become more pronounced. Such challenges mainly include (a) interfaces that support data collection, (b) data processing, (c) data distribution & scaling of interfaces, (d) trust differentiation when exposing to 3rd party applications, (e) network overload on the exposed Application Programming Interfaces (APIs), (f) privacy risks, and (g) latency/performance challenges. Devices with BCS capabilities introduce a unique set of data streams, distinct from the conventional user plane (UP) and CP data. These streams necessitate an architectural paradigm shift, creating an entirely new stream of data which, while resembling UP data, is not tied necessarily to a specific user.

From an architectural standpoint, the core-RAN continuum's role in aggregating, processing, and exposing data becomes pivotal. A careful balance must be struck between ensuring data integrity and meeting QoS requirements for both small data packets from power-limited sensors and high-volume data streams for UEs and broadband services. This balance is further complicated by the trustworthiness of data exposure. In the 6G context, data is not only collected but also cleaned and labelled at various network locations. This process necessitates robust architectural enablers that prioritize security, privacy and trust, ensuring data protection throughout its lifecycle.

Furthermore, the exposure and data management enablers are critical to the effective implementation of the services such as JCAS. These enablers facilitate the interaction between network functions, services, and third-party applications. Architecturally, this means a more intricate data flow, with data being exposed to both in-network and external entities. Such exposure poses challenges, including trust differentiation when exposing to different network sub-systems (e.g., RAN), when exposing to third-party applications, potential network overload on exposed APIs, and the strategic placement of applications leveraging sensing functionalities to meet stringent QoS targets.

Sensing solutions are essential for an authentic digital representation of the physical world (i.e., digital twin DT) enabling new human experiences through immersive mixed reality digital worlds. DT covers a complete view and the full lifecycle of the physical system, unlike simulators, which only focus on parts or several parts of it. The utilization of real-time data from diverse sources is a fundamental aspect of DT [HEX223-D12]. In the context of 6G, the integration of JCAS mechanisms serves to enhance the digital twinning capability. In this regard, data collection, data fusion, and data analysis are essential tasks for the generation of DTs. More specifically, identifying the sets of 6G nodes that are the most pertinent and provide the highest level of contribution towards the establishment of a precise DT would help avoiding unnecessary data collection and redundant sensing, saving network resources. 6G will play an important role in acquisition and fusion of data as well as rendering the digital twin. The network should assure that the latency of the data flow between the real network and its DT is kept in check to ensure timely ML model training and inference. However, due to the expected large amount of data that needs to be communicated between the 6G network DT and the physical 6G system (in real-time in some cases), how to efficiently integrate/interconnect these two entities is a big challenge [NMS+22].

Platform requirements for data collection operations shall be provided and dynamic adaptation of platform resources (e.g., storage) and capabilities (e.g., data formats compatible to different compute platforms) to changing requirements shall be supported. Data discovery/ exposure capabilities, enabling data advertisement to network entities that may be interested in consuming the said data together with the determining the said entities shall be supported.

Any network entity with proper access rights should be able to access data or model(s), stored from another entity. A form of authorization and/or authentication should be performed when a network entity is trying to access, update or share data, analytics and model from another entity. Security should be enabled E2E for any operation of the data collection services, including access, exposure, storage, cleaning, processing and encoding. The network should be able to identify energy-aware data collection services and facilitate their operations. Data collection and exposure can be based on a local configuration or a configuration received from the requester. Different data consumers exist in the network (defined as general network entity), such as UEs, RAN nodes, CN NFs, AFs, 3rd party applications, OAM, etc. Discovery, configuration and in some cases evaluations of such data sources are among the functionalities to cover in 6G.

The network's storage capabilities must ensure that data regarding its operations and services are stored according to quality standards set between data producers and consumers. Storage should be efficiently distributed for various network functions, considering factors like data longevity, sharing limitations, ownership, and technical constraints. Different data types may require separate databases but are generally categorized under a common term "database."

Separating control and data planes seems reasonable since the control requests and actual data flows have different characteristics (e.g., small vs. massive data flows). Also, the measurement data may have characteristics that resemble actual user data, however, there are important differences. One possible solution to deal with such large data collection is to take advantages from a data plan, i.e., a data plane to carry sensing measurement data and other large data sets within the network has a number of benefits, e.g., dedicated hardware can be used to improve performance of the data plane.

6.1.3 Preliminary workflows and evaluation

As long as the position of gNBs and UEs are known to the operator, this information can be used for sensing and enable some kind of *QoS-based* sensing. However, if this is not the case, it is likely that sensing services will have to be provided by the network on a *best-effort* basis. One way to improve sensing quality is obviously to carry out more radio measurements prior to exposure of the measurement report to the requesting application, preferably measurements that are geographically distributed. Involving for instance more UEs in the sensing and measurement process is likely to improve sensing quality. Involving more network nodes (UEs or base stations) naturally leads to architectural challenges of centralized vs distributed inference and processing of the measurements.

Sensing in the context of Hexa-X-II WP3 refers to the network *inferring information about the physical environment* (for instance detect an object in a specific area of interest) from *radio measurements*. This inference is likely to be carried out in a *staged process* within the architecture. The outermost measurement node will acquire raw radio measurements which, on one hand, it could transfer directly to a more centralized network node or, alternatively, it could pre-process and compress and already carry out initial stages of the inference.

As stated in the previous section, the core-RAN continuum's role in aggregating, processing, and exposing data becomes pivotal. Not only from an architectural standpoint (as addressed above) but also from a signal flow and data volumes, and sensing performance standpoint, as addressed here. Figure 6-2 shows two canonical architectures that clearly illustrate the architectural challenges (Here, we address challenges related to the signal flow and volumes, and sensing performance, but also challenges related to privacy and robustness against attacks are relevant in these canonical architectures). In a first architecture, the measurement node provides the final (exposed) measurement report directly from the radio measurements it carries out. Inference of the requested geographical information fully carried out locally in the measurement device: signalling compression to the deeper network layers is maximal, signalling overhead is minimal, and the sensing performance is likely to be modest at best (in some measure). In a second architecture, the measurement node

merely transfers its measurements to a centralized network node, where inference of the geographical information is carried out based on measurements of a massive number of nodes. In this architecture, no data compression to the deeper network layers is done, signalling overhead is maximal and often prohibitively large, but the sensing performance is likely to be best.

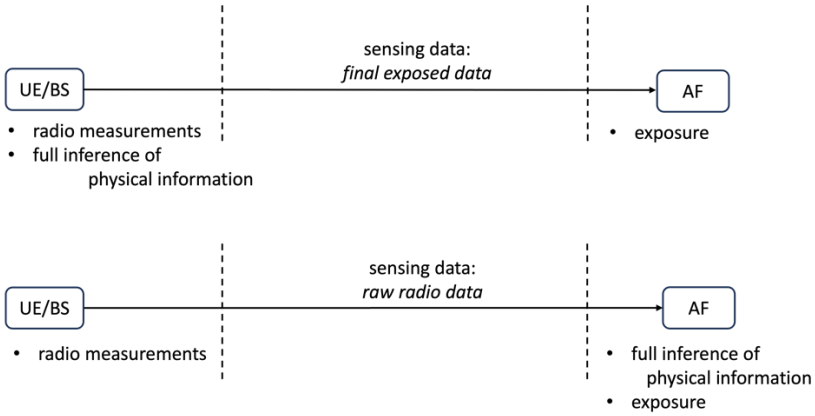


Figure 6-2 Canonical forms of sensing. Top: Inference of geographical sensing information from radio measurements is carried out in the node where the radio measurement takes place (UE or BS). Information transfer to the Application Function (AF) occurs in finally exposed form. Bottom: Radio measurements are transferred as raw radio data to the AF, where inference of geographical sensing information takes place.

6G is likely to comprise of a design where some trade-off of the above canonical ways is done in a *staged* process of inference of sensing information. While sensing is done on a best-effort basis, the network will be able to trade off sensing performance against data signalling overhead. A more centralized inference has to be compared to a distributed design in terms of (primarily) data overhead and performance. One way to address this challenge is to first, define a number of sensing information classes, where the sensing requests from the applications are required into a certain exposure format for each class.

In conclusion, the architectural implications of introducing device BCS data and capabilities exposure mechanisms are manifold. They require a holistic approach that not only addresses the increased data volume and computational load but also ensures data integrity, security, privacy, and trustworthiness. As 6G networks evolve, these architectural considerations will play a pivotal role in ensuring the seamless integration of a wide range of devices and applications, driving the next wave of digital transformation.

6.1.4 Summary

To enable effective beyond communication service delivery, exposure of data and beyond communication service characteristics and capabilities should be enabled in a secure, privacy-preserving and efficient manner. Also, authorisation and authentication should be provided for such data consumer functions. Architectural enhancements to the E2E system to support E2E data management both in-network, but also towards 3rd parties should be provided.

Table 6-1 Exposure and data management summary table

Description	Exposure and data management: This enabler encompasses various mechanisms that allow the exposure of data generated by various producers (including the RAN and sensing nodes) towards network-centric and application layers; besides exposure - storage, processing, trust are in focus	
Benefits	**KPI improvement**	Controlled data traffic/overhead via efficient data management, higher number of vertical services consuming (B)CS data, higher data availability, low latency (serving the data consumers from the nearest data provider), Novel services exploiting BCS data, 6G contributing to social aspects such as safety, trust, and sustainability increase due to novel services beyond communications (e.g., via JCAS in the connected mobility domain, environmental sensing for urban areas, or applications in the agriculture domain), scalability

	Design principles [HEX223-D21]	#1 Support and exposure of 6G services and capabilities #3 Flexibility to different network scenarios #4 Network scalability #7 Internal interfaces are cloud-optimized #8 Separation of concerns of network functions
	Dependencies / Basis for another enabler	JCAS protocols, signalling and procedures Compute offloading protocols, signalling and procedures, DataOps, MLOps, E2E context awareness management, Integration and orchestration of extreme edge resources in the computing continuum
Implications	**Requirements**	Exposure of data collection, aggregation, labelling between consuming functions in core and 3rd party applications Device capabilities (e.g., sensing, resolution, etc.) exposure to beyond communications service resource control, as well as management and orchestration entities. Requirements derived from [22.137] 3GPP TS 22.137 "Integrated Sensing and Communication": - … the 5G network shall be able to provide secure means to report sensing result to a trusted third-party requesting information about a target object when specific requested conditions are met) - … the 5G network shall provide secure means for a trusted third-party to request 5G wireless sensing service based on specific parameters (e.g., refresh rate, period of time, sensing KPIs, geographical location) and to receive the corresponding sensing results. - … the 5G network may provide secure means to expose to a trusted third-party the combined sensing result derived from the joint processing of the 3GPP sensing data and non-3GPP sensing data. Authorization and authentication (some form of authentication should be provided)
	Standard relations & regulations	3GPP TSG SA WG 1, TS 22.137 Integrated Sensing and Communication 3GPP TS 26.531 Data Collection and Reporting; General Description and Architecture 3GPP TS 29.503 5G System; Unified Data Management Services TS 23.288 Architecture enhancements for 5G System (5GS) to support network data analytics services
	Required resources	Storage resources for BCS data aggregation (post-processing, exposure to 3rd parties) Computing resources for raw BCS data pre-processing (prior to storage) Bandwidth to support flow of BCS data between entities/end points E2E communication and computation delay for Ultra-Reliable Low-Latency Communications (URLLC) use cases and non-delay tolerable services

6.2 JCAS protocols, signalling and procedures

6.2.1 Introduction

Sensing applies dynamic characteristics of the (radio) environment to detect, e.g., objects that are not connected to a network, in a specific area of interest. For sensing to work, BSs and/or UEs can participate as transmitters

or receivers of sensing signals, according to the sensing requirements, the network conditions, the sensing capabilities of involved sensing devices and the environment where the sensing takes place, etc. For integration of sensing services in a future communication system to, among others, support the various use cases presented in [HEX223-D12], enhancements of the 5G and beyond architecture and protocols are needed. 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. The SeMF can be designed as a dedicated NF, since it is enabling a new functionality for next generation networks. An alternative option would be to integrate the SeMF services as part of the Location Management Function (LMF). However, it should be noted that LMF is focusing on location or ranging services for a target UE only, which is not the case for the Sensing services, as presented above. The SeMF can also include sensing control, sensing processing and the sensing requests. Note that the extend of which functionalities are included within SeMF as well as their implications is for future study.

6.2.2 Architectural implications

In previous studies, e.g. [HEX223-D32], we have presented ideas on how the architecture needs to be evolved, i.e., new functions needed and new transport, to support JCAS when all involved measurement nodes are base stations. In this study we extend our work to also include UEs as measurement nodes.

The design of a solution with UE involvement has, in addition to technical requirements on resolution, etc., an objective to meet SA1 requirements [22.837]:

- The 5G 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 5G 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. As mentioned in previous work (cf., Figure 6-2), sensing starts with an application sending a sensing request over the CP to a request function. This function receives requests, makes sure that the requester is authorized and then forwards the request to the correct sensing control function. The sensing control indirectly configures measurement nodes, both BSs and UEs, and the sensing processing function; also, via CP. After measurements have been performed, results are sent to the processing function over the Data Plane (DP). Control plane and data plane separation enables building each with the right technology and capacity. When the UE is involved in sensing the UE needs to be connected to the network via a serving base station (Serving BS). Even though the latter is drawn as a separate node in Figure 6-3 the serving BS could be co-located with one of Tx or Rx base stations above.

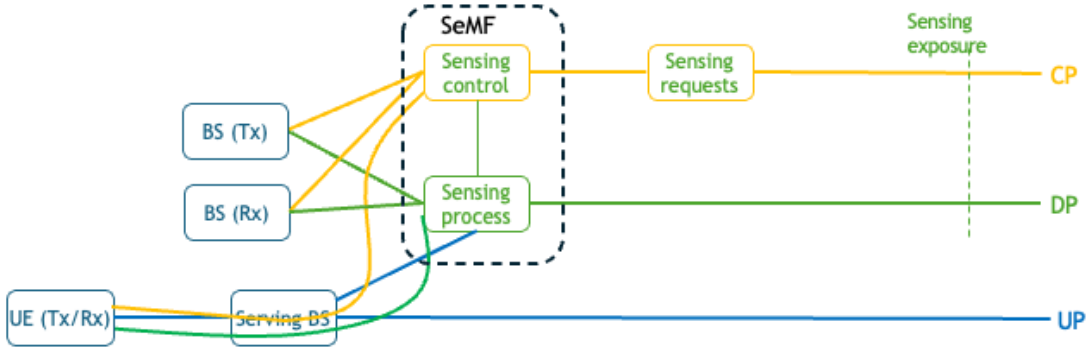


Figure 6-3 Functional architecture, network-based sensing with UE involvement for a bi-static sensing setup, i.e., Tx and Rx a located in separate nodes.

The sensing request and corresponding network functions are not affected very much, or at all, by adding UE assistance. The main piece of information in a request is still the area that the application wants to investigate or sense.

The basic Sensing Control Function (SCF) translates the requested sensing area into necessary measurement nodes, i.e., determining what nodes transmit beams that cover the requested area, and provide configurations to the identified nodes to enable actual measurement. With UE assistance the SCF also finds the UEs to be involved (mapping area into relevant UEs). Note that the SCF also configures the sensing processing function (see more in the next session) with information necessary for a correct interpretation, e.g., geometry of involved nodes. The SCF may also coordinate resources between sensing and communication, e.g., sharing in time, frequency, sharing available antennas, transmission power, processing capabilities, etc., however, the final decision on scheduling is taken by the scheduler in the base station. In a cell where resources are scarce, e.g., when the overall load is high, there will be a performance trade-off between the communication load and the sensing accuracy and/or frequency.

As mentioned in previous deliverables [HEX223-D32] sensing measurements are sent to the "sensing processing" function (SPF) for processing. In most cases the SPF interprets sensing measurements applying the geometry of the involved nodes, and possible other information about the surroundings, e.g., a 3D map of buildings. Further, the SCF provides the interpretation in a format meaningful to an external receiver, i.e., answers the "question" provided in the sensing request. In case of sensor fusion information, which would be handled by the SPF. There may be use cases where raw data, measurements, reported events, etc., are forwarded to an external processing unit, however, that is not discussed any more here. This SPF also ensures that the privacy, e.g., of bystanders is protected.

To summarize, involved measurement nodes comprise sensing capable RAN entities and UEs, e.g., base stations and UEs having the capability to send and receive Sensing Radio Signals (SeRS) used for JCAS as well as reporting measurement results. With involved UEs comes an additional dimension as the UE may be owned by a user and the use of the UE for sensing needs to be allowed by the user. This contribution focuses on an important part in the process of involving UEs, namely, how to find and activate UEs for sensing.

When the network receives a request to do sensing in a specific target area, it will send a request to one or more base stations, which serve UEs in this area. As described above, the network is, through the SCF, aware of the geographical areas of the base stations' service. Also, the SCF will have information on where gNBs have LoS to the investigated area and from the sensing request, determine whether UE assistance is necessary.

The concerned base station will in a first step check if there are any connected UEs within the area, in a second step it will check for inactive UEs within the area, e.g., based on UEs that are configured with SeRS for positioning in inactive. Inactive UEs are not connected to a particular base station but retains the configuration it had since the previous connection. Thus, the inactive UE needs to connect before it can transmit user data.

It is not enough to just detect UEs but the detected UEs need to be willing and capable to perform sensing. Whether a UE is capable can probably be some functions added to the existing list of UE capabilities. In this way procedures to signal capabilities between UE and gNB already exist. Regarding "willing", this is a bit more complicated since this may involve user consent, but willing could also be related to, e.g., battery status. Maybe this can be part of the subscription eventually.

If no UEs are found in these two steps the network has to page UEs, which is the ordinary way to let UEs know that something is about to happen. For sensing, since this is new service, it is likely that a new paging procedure is used. One example that distinguishes sensing paging from legacy paging, is to use a new paging Radio Network Temporary Identifier (RNTI) targeting UEs willing to do sensing. This P-RNTI may be mapped to a specific area of a cell, e.g., based on Synchronization Signal Block (SSB) or a specific area corresponding to a cell or beam. Thus, UEs that detect this new P-RNTI will receive a paging message indicating the area to sense and possibly other information.

When a UE meets the requirements of the paging message, e.g., capability and willingness to participate in sensing, the UE connects with the network to provide this information to the SCF. The SCF will then configure the sensing measurement, including all necessary nodes, and perform the measurements as described previously. If UEs are involved UE measurements will be forwarded via the data plane to the SPF, just as measurements performed by base stations would be.

Whether a UE can participate in sensing is not a binary on/off notion. In fact, sensing, as any other service, comes with a notion of quality-of-service and a UE can participate in sensing with a higher or lower quality.

In order to address this, a *sensing quality indicator* can be introduced. Such an indicator would be sent by the UE and indicate the expected sensing accuracy/quality.

As described above, the inference of sensing information from the radio measurements is likely to be carried out in a *staged process* within the architecture. The measurement node, for instance a UE, acquires raw radio measurements which it could transfer directly to a more centralized network node or, alternatively, it could pre-process and carry out initial stages of the inference. Often it will be desirable to carry out some preprocessing in the UE to avoid capacity issues in the RAN. In a further node, for instance the BS, pre-processed information will then be fused with pre-processed information from other measurement nodes and further refined, before being transferred to a centralized sensing processing node. This staged process of inference of sensing information, along with the various sensing information classes (exposure formats) and various fusion strategies imply the need for stage-specific measurement reports. Although, the SCF and SPF often are depicted as separate nodes in figures, the actual location of such functionality needs to be determined, i.e., the functions can be separate NFs or just part of already existing nodes.

6.2.3 Preliminary workflows and evaluation

The SeMF can receive sensing requests from a UE, an external Application Function (AF), or even an NF that include information about the requested sensing service, the area of interest, the sensing QoS requirements, information about the objects to be sensed, etc. Figure 6-4 describes an example of how a sensing service request by an AF could be supported by a communication system. The SeMF or another NF can authorize the sensing service consumer and verify privacy requirements of the requested sensing service. Thereinafter, the SeMF undertakes the coordination of sensing procedures, determining the appropriate sensing method (e.g., mono-static BS-based sensing, bi-static BS-based, UE-assisted sensing [HEX223-D42]) and configuration information, according to the requirements received from the requesting consumer as part of the sensing service request, and the available sensing capabilities in the area that the request defines. The SeMF discovers and selects the BSs and/or UEs to be involved in the sensing service. The sensing configuration indicates the one or more BSs and/or one or more UEs that could be involved in the sensing procedure, the sensing duration, the sensing region and optionally radio access network parameters for sensing e.g., the selected frequency.

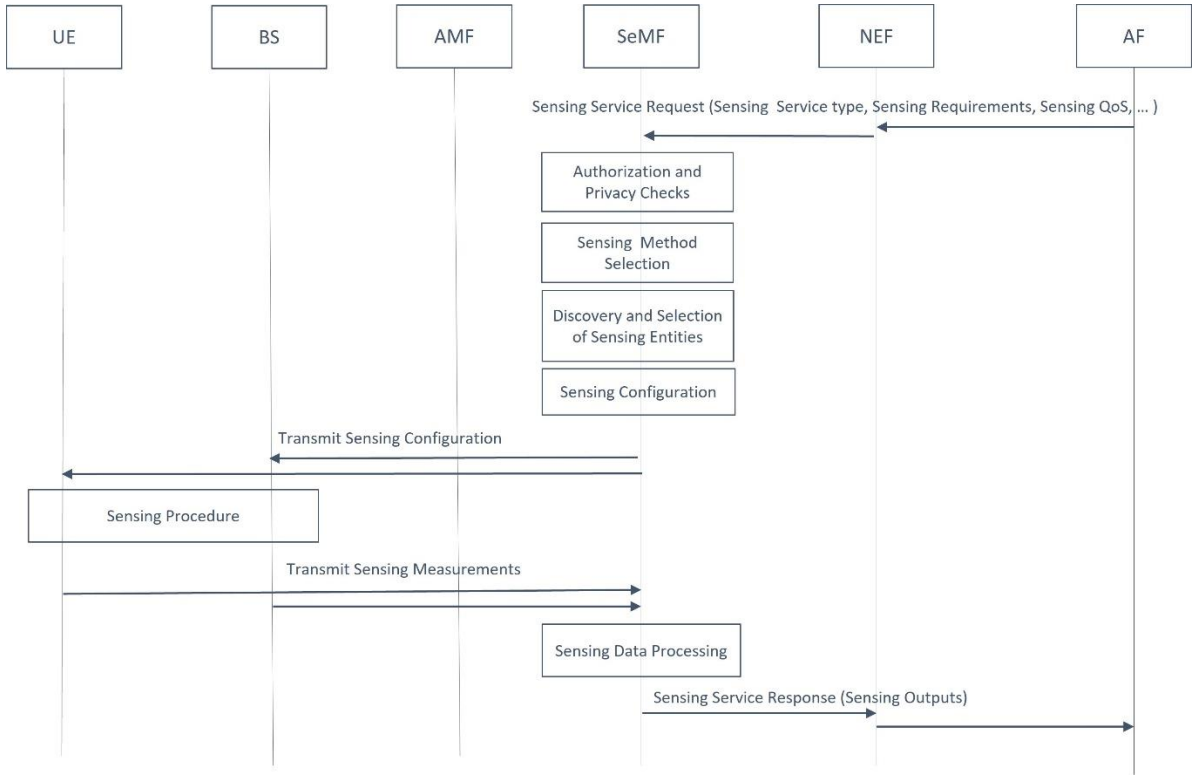


Figure 6-4 Example Deployment of Sensing Services in a Communication System

The SeMF firstly ensures the consent of the entities involved in a sensing procedure and then requests the involved BSs and/or UEs to initiate a sensing procedure, according to the determined sensing configuration. For instance, a BS allocates sensing relevant resources to realize the sensing procedure by transmit sensing signals, according to the sensing QoS requirements. Each BS and/or UE that has the role to receive sensing signals provides the sensing measurements to the SeMF for the processing of the sensing data and the determination of the sensing outputs. The output to the requesting consumer can include a list of objects or information about the tracking of the object or even sensing measurements that can be processed by a third party. The type of output depends on the requested sensing service.

6.2.4 Summary

Integration of sensing capability to the RAN nodes, especially on the receiver side is a key pre-requisite, along with the specification of the sensing resources, as well as the classes and modes that should be used for the sensing reporting. This enabler presents a new SeMF, that can incorporate multiple sensing functionalities such as sensing control, sensing requests and sensing processing. Through interactions with the other NFs, this new function can realize sensing service according to the specific requirements (e.g., sensing accuracy, resolution, latency etc.) posed by the service requester. The joint communication and sensing capability would enable a large set of use cases (i.e., listed in [HEX223-D12]), such as from robots to cobots, massive twinning, and immersive telepresence through providing high location accuracy and real time tracking.

Table 6-2 JCAS protocols, signaling, and procedures summary table.

Description	JCAS protocols, signaling, and procedures: This enabler encompasses architecture enhancements and protocols to integrate sensing services in a communication system	
Benefits	**KPI improvement**	Realize sensing as a service to achieve required quality of sensing (i.e., measured by sensing accuracy, sensing resolution, sensing latency etc.) according to the application/use case, Sensing capability could be used for object detection for safety purposes, for environment monitoring, human motion monitoring e.g., healthcare etc.

	Design principles [HEX223-D21]	#1 Support and exposure for 6G services and capabilities #3 Flexibility to different network scenarios (sensing flexibility) #6 Persistent security and privacy
	Dependencies / Basis for another enabler	Exposure and data management Compute offloading protocols, signaling and procedures, Architectural means and protocols, Multi-connectivity, 6G Network modularisation, E2E service design in modular 6G
Implications	**Requirements**	Integration of sensing capability to the RAN nodes (esp. on the receiver side), Identification of sensing resources, Identification of sensing report class/format/content/mode 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 SA1 TR (requirements).
	Required resources	Sensing resources, Bandwidth for transferring sensing data, computation, and processing resources to process sensing data and generate outputs, storage resources in case data/output are stored, new paging identifier

6.3 Compute offloading protocols, signalling and procedures

6.3.1 Introduction

As stated in [HEX223-D32], the introduction of compute offloading in the next generation of networks should not increase the complexity of the communication protocol. This can be achieved by tight integration and true convergence of communication and computing, by introducing novel architectural components for distribute computing. To satisfy the strict requirements on the computation and communication latency, trustworthiness, power consumption and data accuracy, it is important to introduce a 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.

Additionally, the proposed compute offloading protocols, signalling and procedures enabler should ensure that both the QoE for the communication as well as the required resiliency and quality of computation are achieved, such that the 6G design principles of (#5) Resilience and availability and (#1) Support and exposure of 6G services from [HEX223-D21] are met. Finally, this enabler aims at defining the basis for *Application-/Device-specific BCS data consuming functions* and *Distributed Compute-as-a-Service* enablers, discussed in 6.3.4 and 6.4.4, respectively.

6.3.2 Architectural implications

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 Nodes (CompN), network nodes 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.

When a device, acting as an ON, decides to offload a computation, it will have to discover and select the candidate CompNs, capable of performing the requested computation while satisfying the associated KPIs. Each CompN should estimate the task(s) execution complexity and resources demand (i.e., computation and storage) based on a common characterization of the offloaded compute workloads (i.e., compute tasks) determined by the following requirements:

- Traffic class (i.e., one-time vs multi-iteration, one-node vs multi-collaborative-nodes)
- Computation complexity (e.g., number of FLOPS and memory)
- Communication requirements (e.g., size of compute payload to be transferred)
- Precision requirement (e.g., quantization level of the compute data and operations)
- Quality of compute service classes (QoCS), such as latency sensitive, precision sensitive, …

These characterizations serve as the basis for the capability exchange and compute offloading coordination between the different nodes.

The general architecture for computational offloading is introduced in [HEX223-D32]. Using it as a foundation for further study, the computation offloading procedure is foreseen as a staged procedure, as illustrated in Figure 6-5. In the first stage (i.e., Node Discovery Phase 1), the compute node capabilities are identified. It is followed by the second stage (i.e., Node Discovery Phase 2) where the request for computation offloading is performed. Finally, the third stage (i.e., Computational Offload Procedure) comprises sending of computing tasks and receiving of compute results.

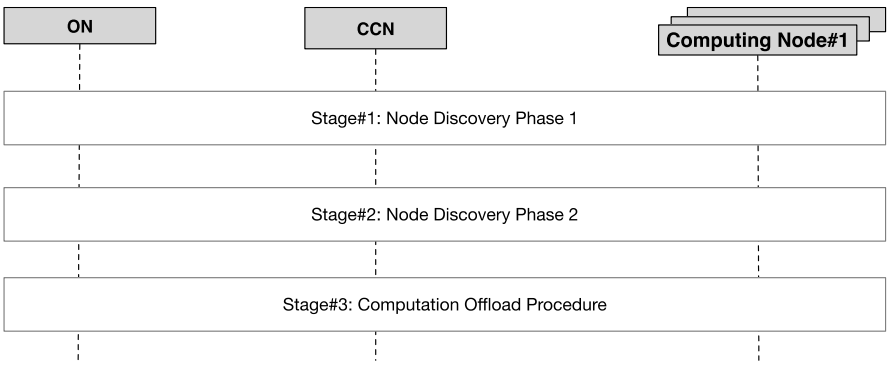


Figure 6-5 Computational Offloading Procedure - High Level Flow.

Device offloading enabled via Compute-as-a-Service (CaaS) moves computation from a mobile device to a network node with more suitable compute and storage capabilities, see Figure 6-6. This may reduce the mobile device's computational needs, heat generations, and power consumption, however at a cost of increased usage of network communication resources. Additionally, CaaS may enable better application scalability, new CPU hungry services (e.g., XR and digital twins), and improved operation times for battery-driven devices. This entails the need for a connection between the device offload functions and the network offload functions. Further analysis is however needed, to conclude on whether such network functionality needs to be standardized or not.

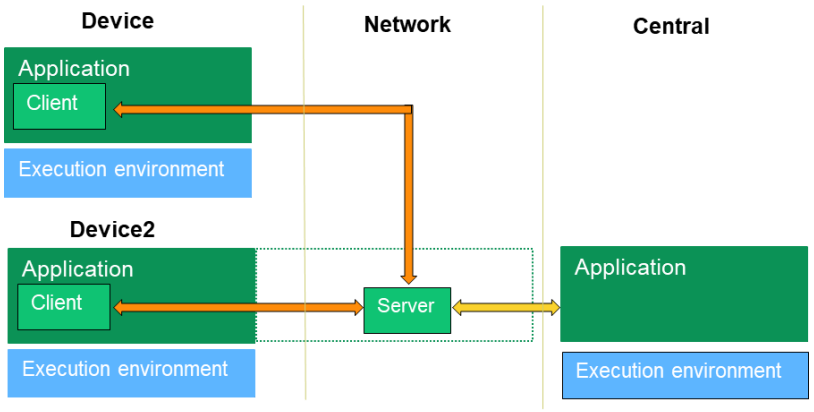


Figure 6-6 Overview of computational offloading of a common coordination task, e.g., to realize collaborative perception or to save overall bandwidth

6.3.3 Preliminary workflows and evaluation

In the architectural implications section above, two different architectural options are presented, with one focused on device offloading enabled via CaaS moving computation from a mobile device to a network node, while the other is focused on a general architecture and procedural flow for compute offload without emphasis on deployment.

In [HEX223-D32], dynamic offloading with the aim to offload application functionality dynamically from a mobile device to network embedded computing is proposed. As a continuation to this work, a demonstration on how to perform dynamic offloading and the resulting benefits and challenges are in focus here. For this study, a scenario is of robot cars, trying to map an outdoor or a disaster area, where the device (i.e., the car) needs to analyse the environment with an onboard camera or radar, is used as a reference use-case.

For efficient dynamic computational offloading, the runtime execution and the time-critical modules of the application running in the device must be replicated in the network. However, it is not expected that software and hardware platforms will be the same in the devices and the hosts that are handling the offload. To counter this, there are several technologies to package and deliver applications, for instance Virtual Machines (VM), like Java VM, or containers. In this demonstration, the emerging WebAssembly runtimes is used, which is selected due to being specifically portable, lightweight, secure, and polyglot. Furthermore, it is assumed that the application is built in a modular fashion, with well-defined modules representing microservices, objects, or even functions. These offload-able modules, perform a specific task, preferably, with as little interaction with the rest of the application on the device as possible, to avoid the unnecessary use of network communication resources. The offload-able modules are dynamically scheduled for offloading depending on situational changes, during application runtime. This is handled by an offload manager in the network and an offload handler in the device.

In this work it is assumed that the device (UE) and offloading cluster (i.e., the network) have a pre-established connection between the offload handler in the device and the offload manager in the network (e.g., via RRC control plane functions), but eventually this should be supported by a discovery service. Additionally, it is assumed that there is a single network node, so there is no need for any compute offloading mobility here. However, in reality CaaS mobility must be supported within the PLMN network, but this is not treated here.

In the experiment, the offloading is manually triggered. In a real service, there would be an automated trigger based on relevant status and context from the network, the device, and applications. The purpose of the demonstration is to examine the performance characteristics of offloading, especially the power consumption, execution time and network utilization.

Figure 6-7 (left) shows the device power consumption for offloading (remote – in light grey) vs no offloading case (local – dark grey). 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.

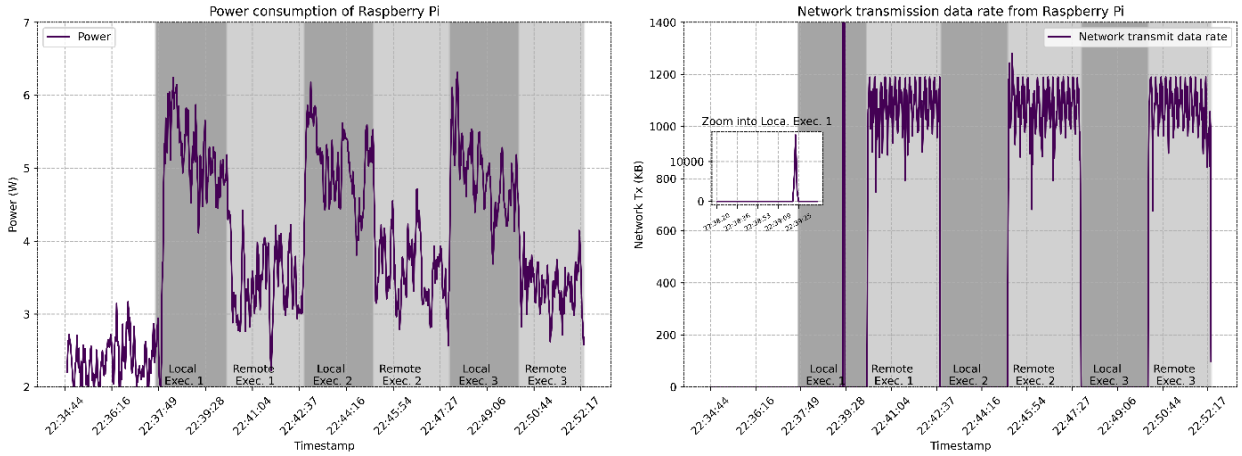


Figure 6-7 Dynamic offloading of a CPU demanding operation from the device to the network. The left figure shows the consumed power when there is local (when there is no offloading) and remote (offloading to the network). The right figure shows the device network transmission (i.e., uplink) due to offloading (remote) vs no offloading (local).

However, this comes at the cost of the use of network communication resources, highlighted in Figure 6-7 (right), to ensure that the module running in the execution environment in the network is synchronized with the remaining application running on the device.

Moreover, in [HEX223-D32], the general architecture for computational offloading is introduced, which is further expanded upon in the above section with Figure 6-6 illustrating the computation offloading as a staged procedure. The messaging exchange foreseen in the node Discovery Phase 1 and Phase 2, are shown in Figure 6-8 and Figure 6-9, respectively. These messages are protocol agnostic, meaning that they just illustrate the messaging exchange flow among nodes, with their exact implementation depending on where the different nodes would be deployed (i.e., NAS, RAN, etc.).

Node Discovery Phase 1, illustrated in Figure 6-8, starts with the initial registration of the Computing Node (CompN) and ON within the CCN. Then, CompNs update the CCN with their available compute capabilities. Based on this information, CCN sends the *Compute Capabilities Update* message (e.g., memory size, computation capabilities, etc.) to the ON. The offloading can be either controlled by the CCN (i.e., the message contains the overall aggregated compute capability of all registered CompNs) or controlled by the ON (i.e., the message contains the compute capability per CompN).

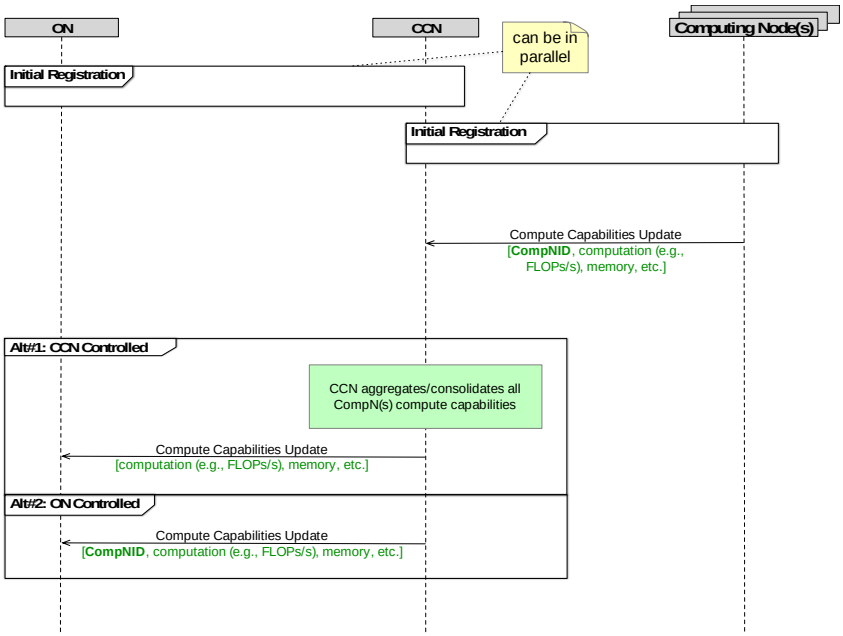


Figure 6-8 Node Discovery Phase 1

The messaging exchange in the Node Discovery Phase 2 is depicted in Figure 6-9. In CCN controlled procedure, ON requests a general compute offload request that can be mapped by the CCN to any of its registered CompN(s). Then, the CCN evaluates the requested computation tasks, and, if approved, sends it to one or more CompNs accordingly. In the ON controlled case, ON requests a specific CompN, and CCN only forwards this request to the specified CompN. The CompN then evaluates the compute request by the CCN and sends a Computation Offload Response to the ON. If one or more CompN(s) reject the Computation Offload Request, depending on the latency requirements of the task(s), CCN may request these task(s) from other available CompN(s), or sends a reject in the *Computation Offload Response* message to the requesting ON. Based on this response, ON follows the Computation Offload Procedure in the third stage, which will be further described in the next deliverable.

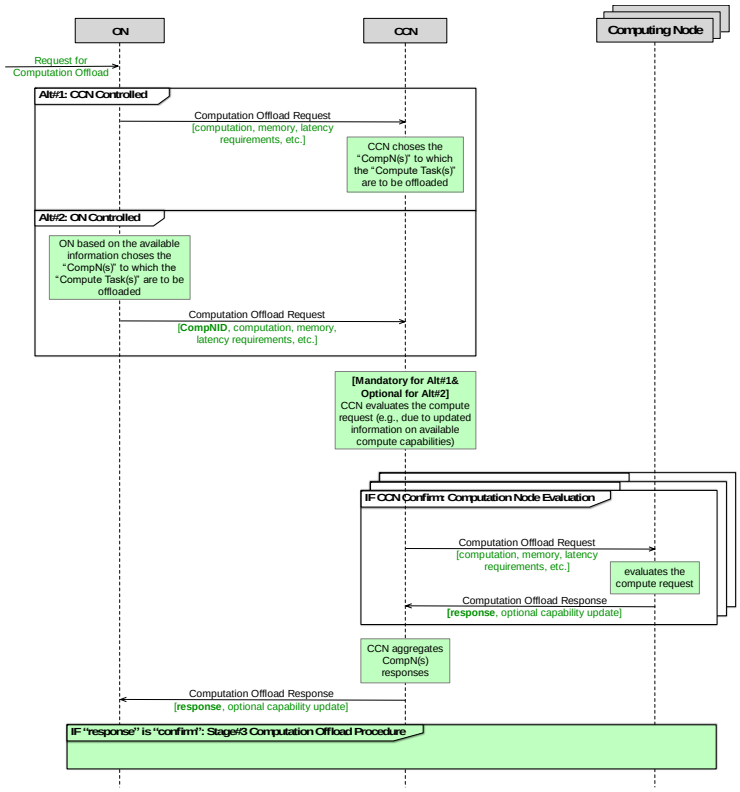


Figure 6-9 Node Discovery Phase 2

6.3.4 Summary

Novel architectural components for compute offloading may have an implication on the network signalling and procedures. This includes discovery of computing nodes, synchronization and coordination of offloading and computing nodes and novel data and compute offloading procedure. Table 6-3 summarizes the main benefits and implications of the *Compute offloading protocols, signalling and procedures* enabler.

Table 6-3 Compute offloading protocols, signalling, and procedures summary table.

Description	Compute offloading protocols, signalling, and procedures: Architecture enhancements and protocols to support compute offloading from mobile devices to network nodes, without increasing the complexity of the communication protocol and satisfy latency, trustworthiness, power consumption, resilience	
Benefits	**KPI improvement**	Energy savings in the device Increasing application scalability and availability (Extended Reality (XR), digital twin, etc.) Communication and computation latency Privacy/security Power consumption Computation Resiliency Quality of computation/ data accuracy Device complexity
	Design principles [HEX223-D21]	(#5) Resilience and availability (#1) Support and exposure of 6G services

	Dependencies / Basis for another enabler	Depends on Exposure and data management enabler, Architectural means and protocols, Integration and orchestration of extreme edge resources in the computing continuum, Multi-domain/Multi-cloud federation Present basis for Application-/Device-specific BCS data consuming functions and Distributed Compute-as-a-Service enabler
Implications	**Requirements**	There is a need to support a connection between the device offload functions and the network offload functions. If the network functionality has to be standardized need to be further analysed. Identify (RAN) sensing procedure (incl. best-effort procedures, managing sensing QoS, sensing scheduling procedures, sensing admission management, etc.) Identify network signalling needs to support procedures New roles of network components (UE, core network, RAN)
	Standard relations & regulations	38.331 (RRC), 3GPP TS 23.501 System architecture for the 5G System (5GS)
	Required resources	Compute resources are needed. Can be placed in the edge network (gNB), far edge (UE) or can also be centralised compute resources if the application is not delay sensitive.

6.4 Application-/Device-specific BCS optimisation architectural enablers

6.4.1 Introduction

In the broader context of 6G network capabilities, the focus on Application-/Device-specific BCS data consuming functions signifies a major shift in network functionality and service provision. This section examines initially the complexities of these functions, which are essential in efficiently managing and utilizing the substantial data generated by a variety of devices within the network. A key aspect is the transition from conventional data handling to a more refined, application- and device-specific method. This change is critical for maximizing the capabilities of 6G networks, facilitating them to effectively support a diverse array of applications.

At the core of this development is the ability to selectively manage and process data in a manner that is precisely aligned with the specific requirements of different applications and devices. This approach not only improves network efficiency and performance but also elevates data privacy and security, which are of utmost importance in the rapidly expanding digital age. The discussion emphasizes the need to develop advanced algorithms and innovative technologies adept at navigating the complexity and volume of BCS data in 6G networks. These technologies are expected to not only optimize data throughput and reduce latency but also to underscore the importance of sustainability and energy efficiency in network operations. As 6G networks evolve, the establishment of robust frameworks for data exposure and processing becomes essential, ensuring that the network remains resilient, secure, and adaptable to the ever-evolving challenges of digital communication and data management.

At the same time, incorporating sensing capabilities into a communication network represents a highly promising domain for the BCS that brings forth numerous possibilities and challenges. There are practical applications aimed at enhancing the network's own performance, as well as captivating scenarios where spatial sensing can be extended as a service to improve the performance of existing applications. Architectural enablers should be introduced that contribute to JCAS as BCS concept, which are essential in efficient sensing of the surrounding environment and in the introduction of the quantum sensing technologies. A key concept that is stressed out in regarding the JCAS are the self-sensing capabilities of the UEs together with the quantum-enhanced communication and sensing capabilities incorporated into the network architecture.

6.4.2 Architectural implications

6.4.2.1 BCS data-consumer application placement optimisations

Optimizing BCS data-consumer application function placement requires a flexible architecture adaptable to various service needs. This necessitates a network infrastructure that can dynamically assign application functions, supported by intelligent algorithms. These algorithms must consider computational load, network resources, latency, and privacy to optimize system performance and energy efficiency efficiently.

The architecture must ensure energy-efficient registration and connectivity checks for BCS, essential for devices with limited energy. An innovative registration mechanism is needed to minimize overhead and conserve energy, while keeping the network responsive, especially for devices in BCS activities like JCAS. These devices require rapid response in varying states. The architecture should balance effective registration protocols with resource efficiency, addressing both connectivity challenges and seamless mobility.

Furthermore, as BCS extends to various data-intensive applications like massive twinning and Compute/AI-as-a-Service, the architecture must support vast, seamless data exchanges while maintaining strict privacy and security standards. This requires the integration of advanced data management frameworks within the architecture that not only regulate data flow but also ensure that the data is processed and stored in a trusted environment. The incorporation of new network functions aimed at supporting trust and security is essential. These functions must be embedded within the architectural fabric, possibly through a layered approach that separates critical data paths from general network traffic, thus establishing clear boundaries for data processing and exposure.

Further to the above, sustainability of operations is also critical. The architecture should not only be robust and secure but also energy-conscious, promoting sustainability in line with 6G directives for energy efficiency. To achieve this, the architectural design must prioritize not just the operational efficiency of data transmission and processing, but also the lifecycle management of the network infrastructure itself. To this end, it should facilitate the seamless introduction, scaling, and retirement of application functions in response to the fluctuating demands of BCS application/UE requests.

Different JCAS applications are associated with stringent requirements. On the one hand, heavy computations need to be performed on the sensed data, coming from different sources, to provide with the information on their localization. On the other hand, JCAS applications could also be associated with delay-strict requirements, where the results are expected to be received in real-time (e.g., stopping a robot machine after detecting a human). In addition, although a far edge cloud is located near the end user and comes with promise of reduced communication delay, it can suffer from scarcity of compute resources which induces high processing delay. The trade-off between network metrics and compute metrics would call for a new approach that enables the Integration of Network and Compute (INC) to perform coordinated optimization.

Figure 6-10 illustrates an architecture, where network communication is provided by a CSP (Communication Service Provider), whereas compute resources are offered by cloud providers (edge cloud and far edge clouds). Note that the CSP and the cloud providers could belong to different organizations. The user application (e.g., JCAS application) should therefore be deployed in a compute site with the required compute resource, while ensuring the associated network performance. The general idea of the INC approach is that instead of controlling or optimizing the use of network and compute resources separately, both are considered as part of the same system and governed by common processes. To this end, an INC server is considered to ensure coordinated optimization between network and compute processes. It therefore collects network metrics (e.g., maximum delay and minimum throughput to an edge cloud) from CSP as well as compute metrics from cloud providers (number of available CPUs/GPUs and memory). This would require standard interface to expose such metrics to an INC server. The availability of such metrics to the INC would allow performing an optimized decision on the placement of user applications in a way to reach the requested network and compute needs (note that usually it is enough to just select the compute site, which can select the target compute node locally). The decision would therefore be followed by a request to the selected compute site to deploy user application, and by another one to the CSP to steer the traffic to the selected compute site while enforcing the request QoS.

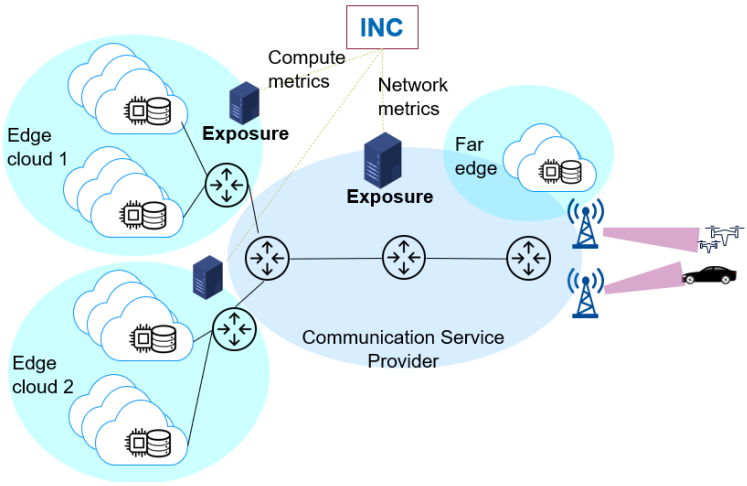


Figure 6-10 Integration of Network and Compute (INC) server collects network and compute metric to decide optimized placement of application.

From architectural point of view the JCAS can be categorized in two operating modes: bistatic and monostatic. In the traditional bistatic mode, the transmitter and receiver are separated, while in the monostatic mode they are located on the same device. For example, in the traditional bistatic localization and tracking, the base station can accurately localize and track the UE. On the other hand, in the monostatic mode the UE is capable to sense the environment and track moving objects in a so-called joint radar and communications.

In addition to the joint radar and communication mode, the monostatic sensing can be performed using a combination of Time-of-Flight (ToF) and Angle-of-Arrival (AoA) measurements. For example, in IEEE 802.1 extracting the Channel State Information (CSI) of the path between an Access Point (AP) can produce accurate AoA estimations. The Fine Time Measurement (FTM) protocol [FTM+16] can offer accurate distance estimations by using the Time of Departure (ToD) and Time of Arrival (ToA) sent between the AP and the client. Combining these two techniques together can give us very accurate estimations of the surrounding environment.

A possible architectural approach in monostatic sensing is the concept of self-sensing where the mmWave radios that are integrated in the device primarily for communication purposes can be re-used to enhance the environmental sensing when the optical sensors experience performance degradation (i.e., in case of LiDARs with translucent material.). Figure 6-11 illustrates the concept of self-sensing where a device (i.e., robot) moves and scans freely the indoor environment using dedicated sensors, while the mmWave radios periodically exchange AoA and ToF estimates between themselves (self-sensing) by bouncing the signal in the environment, thus enabling accurate estimates of the target object/material surface.

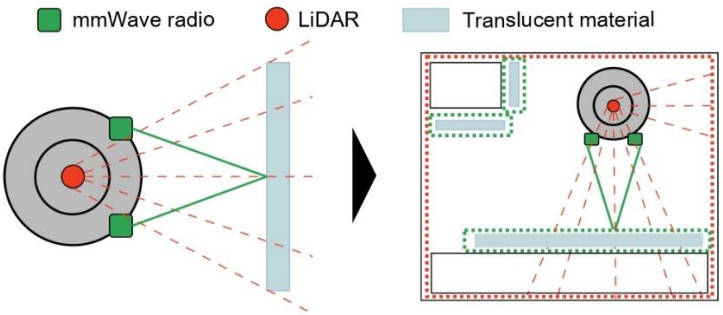


Figure 6-11 Key intuition behind self-sensing

The architectural implications of introducing the monostatic self-sensing concept and sensing data exposure are twofold. On one side, they require an architecture and protocols where the radios that perform self-sensing can exchange the sensing information with the infrastructure in a secure and efficient way, not degrading the communication performance. On the other hand, the same radios that are primarily designed and optimized for communication will need to evolve so they can support frequent sensing and addition to the communication.

6.4.2.2 Quantum-based network architecture optimisations

Key aspects of the quantum-based network architecture are quantum sensors for increased precision and sensitivity and quantum techniques such as Quantum Key Distribution (QKD) for secure communication. These enhancements are important for maximizing the applicability of JCAS, facilitating them to efficiently support a various types of use cases. At the same time, focusing on the JCAS as BCS, the integration of quantum-based technologies into the network architecture introduces unique aspects that exploit the principles of quantum mechanics for enhanced communication and sensing capabilities. The network will also benefit from advances in quantum computing to meet the increasing demand for computing power. The architecture may include hybrid layers that integrate classical and quantum communications. Classical layers will handle conventional data transmission, while quantum layers will support quantum communication for specific applications, creating a two-tier network.

6.4.3 Preliminary workflows and evaluation

6.4.3.1 BCS data consumer application-driven optimization

In the context of 6G, the optimization of BCS data consuming application function placement considering certain data exposure interfaces from the network side is critical in order to satisfy various QoS (communication service and beyond) requirements. The focus of this evaluation is to analyse and compare different application placement strategies within a simulated network environment. The evaluation leverages a Python-based simulation environment incorporating network graphs, application requirements, and node capabilities, to explore the efficacy of these strategies under varying conditions.

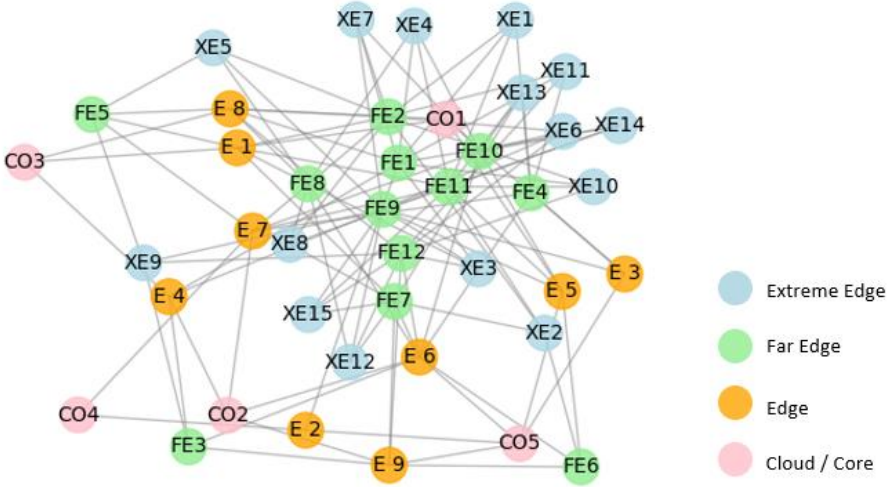


Figure 6-12 Simulated environment of a multi-layer network for application placement optimization

The simulation environment, as shown in Figure 6-12, models a multi-layer network with nodes representing different layers such as Extreme Edge, Far Edge, Edge, and Core. Each node possesses attributes like CPU, memory, and storage, reflecting a realistic network scenario. Each layer is configured to mirror the increasing resource availability observed as we delve deeper into the network. Starting at the Extreme Edge, node capabilities are limited (CPU capabilities start modestly, ranging from 10 to 200 MHz, paired with memory capacities that vary from 0.5 to 2 GB, and storage options extending from 100 MB to 4 GB), encapsulating the constraints of edge computing devices. As we progress towards the Core layer, resources expand significantly, using nodes with CPUs operating at the GHz level and memory scaling to several GBs, and storage capabilities ranging from a few GBs to multiple TBs, reflecting the robust processing environment of central data centres. Application's instances with specific requirements (CPU, memory, storage, latency sensitivity) are placed within this network, and their performance is evaluated based on several metrics.

Two main application placement strategies are evaluated:

1. **Genetic Algorithm**: This strategy employs a genetic algorithm to optimize the placement of application instances across the network. The algorithm considers factors such as latency, energy consumption, data exposure, and resource utilization to determine the optimal node for each application instance.

2. **Random Placement**: As a baseline comparison, a random placement strategy is also evaluated. Here, application instances are randomly assigned to nodes within the network, without considering the optimization factors.

Evaluation Metrics

The performance of each placement strategy is assessed based on the following metrics:

- **Latency**: Measures the time delay experienced in the network, a crucial factor for latency-sensitive applications.
- **Energy Consumption**: Evaluates the energy efficiency of the application placement, an important consideration for sustainable network operations.
- **Data Exposure**: Assesses the extent of data exposure across the network, i.e., the distance from the data source, which has implications for security and privacy.
- **Resource Utilization**: Indicates how effectively network resources are utilized by the placed applications.

The results of the evaluation scenarios (Figure 6-13) reveal significant differences between the GA and random placement strategies. The GA consistently achieves lower latency and energy consumption, suggesting a more efficient utilization of network resources. In contrast, random placement, leads to suboptimal performance, particularly in terms of latency and energy efficiency. Data exposure and resource utilization metrics also show the GA's ability to balance network load and manage data more securely and efficiently. This is crucial in 6G networks where data management and security are paramount.

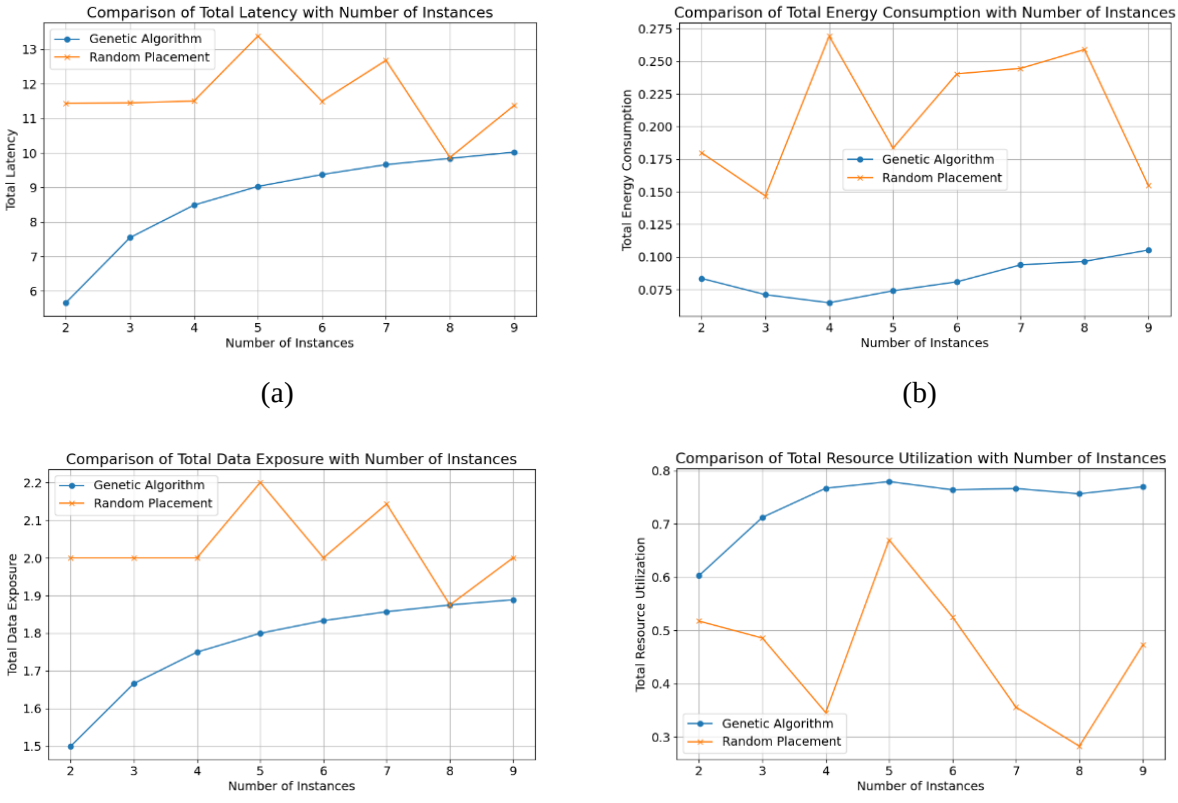


Figure 6-13 Application component placement optimisation evaluation: E2E Latency (a), Energy Consumption (b), Data Exposure (c), Resource Utilisation (d)

6.4.3.2 Application- and Device-driven optimization for BCS

Figure 6-14 shows a generic call flow of the INC approach for a coordinated of network and compute optimization.

- Network and compute metrics are continuously exchanged with the INC server. This can be performed via exposure entities and require standardized interfaces.
- A request for placing the target application is formulated to the INC server. Such a request is mainly formulated by application provider and includes both compute and network requirements. The INC server will thereafter resolve the request and decides the compute site where the application should be placed. The decision is performed in a way to meet both compute and network requirements of the application to place, which can be performed thanks to the metrics received from the network and the compute providers. Once the request is resolved and the compute site is selected, the INC formulates a request to the selected compute site to deploy the target application, followed by a request to the CSP to steer user traffic to the selected compute site while enforcing the requested network requirements.

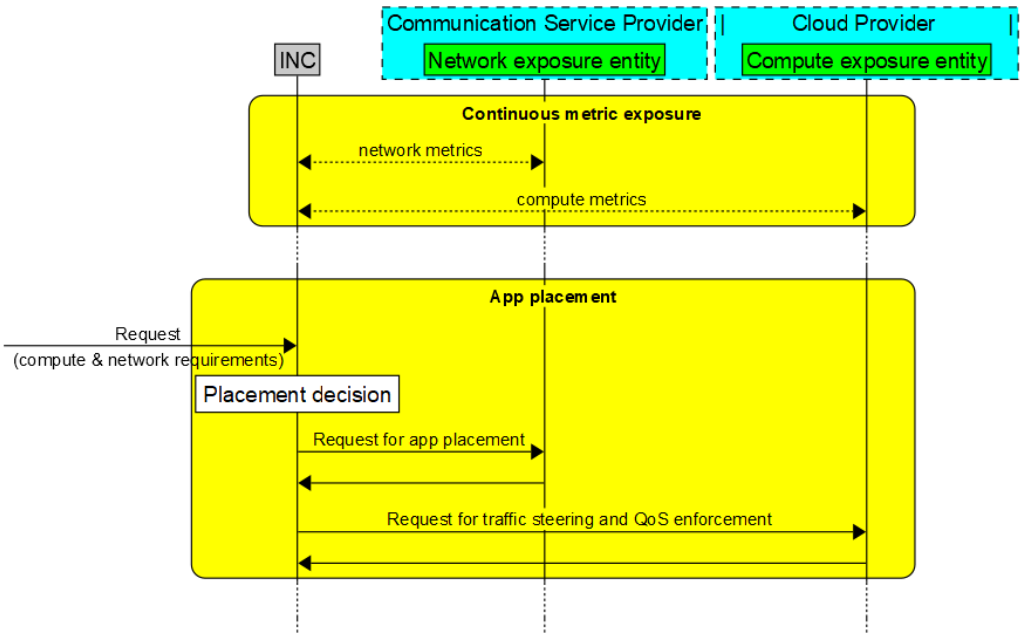


Figure 6-14 A generic call flow for coordinated network and compute optimization

6.4.3.3 Enhancing JCAS capabilities – Indoor mapping

To evaluate the feasibility of the self-sensing concept we developed a simple prototype illustrated in Figure 6-15. Two mmWave devices with the same orientation are placed on a table, separated by absorbing material that eliminates undesired paths, prevents interference, and increases the directivity of the antennas. In addition, a Raspberry Pi is mounted that connects directly to the mmWave devices. For the mmWave devices, we used COTS devices, namely the MikroTik wAP 60G and MikroTik wAP 60Gx3. Since the manufacturer installs a proprietary operating system with IEEE 802.11ad stack that has limited access, we use the implementation in [BMG+22], which ported OpenWRT to this platform and replaced the firmware. This allowed us to use a customized version of the open-source wil6210 Linux driver that exposes FTM and CSI.

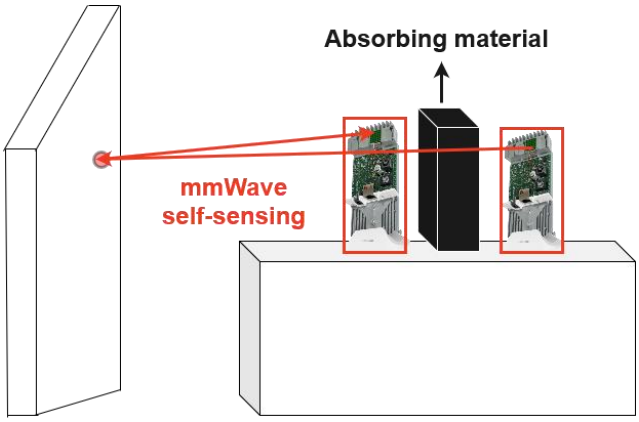


Figure 6-15 Self-sensing prototype

In order to grasp the capabilities and constraints of the self-sensing concept, we performed an analysis to delve into the accuracy of the estimations relative to the distance between the devices and the walls, and the angle assessments. We assess the estimated distances by positioning our self-sensing system facing a wall and progressively increasing the distance between mmWave devices and the wall, from 1 meter to 7 meters, with a 2 meter step. Likewise, for angle estimations, we conduct Angle of Arrival (AoA) measurements at various angles (-40 to 40) at distinct positions from the wall i.e., 1m, 3m, and 5 m. For every distance and angle estimation, we conduct 50 measurements and produce the graphs depicted in Figure 6-16 and Figure 6-17.

Figure 6-16 shows the Empirical Cumulative Distribution Function (ECDF) of the self-sensing distance errors using FTM. We observe that with the self-sensing concept we can attain centimetre-level precision across all distances, maintaining distance error under 10 cm in 80% of instances. Additionally, Figure 6-16 also indicates that the FTM distance error rises with increasing distance. This is because, as the distance increases between the mmWave devices and the wall, the attenuation is higher and more susceptible to noise. Moreover, the farther you move from an obstacle, the greater the likelihood of encountering second-order or higher reflections within the system, leading to larger errors, as with yellow line (7 m) indicates. Nevertheless, it can be observed that at 7 meters the error is bounded in the order of centimetres.

Figure 6-17 shows a box plot of self-sensing error in the azimuth plane. We observe that the angle estimation error also escalates with the enlargement of the rotation angle of the mmWave devices. The reason is that the aperture of the antenna is 60 degrees, therefore the higher the rotation is, the less receive power, which degrades the azimuth estimation. It is worth mentioning that for distances above 7 m and angles above ±40 the current self-sensing concept encounters challenges in data collection due to the amplified number of reflections captured. These encouraging findings indicate that the angle and distance estimations derived from off-the-shelf mmWave devices can be precise and beneficial for sensing the surrounding environment. While angle estimations indicate that when performing self-sensing the mmWave devices should consistently face the object/material surface at angular orientations ranging from ±40, and the estimated distances indicate that we can obtain up to 7 meter-range, which is comparable to commercially available Lidars.

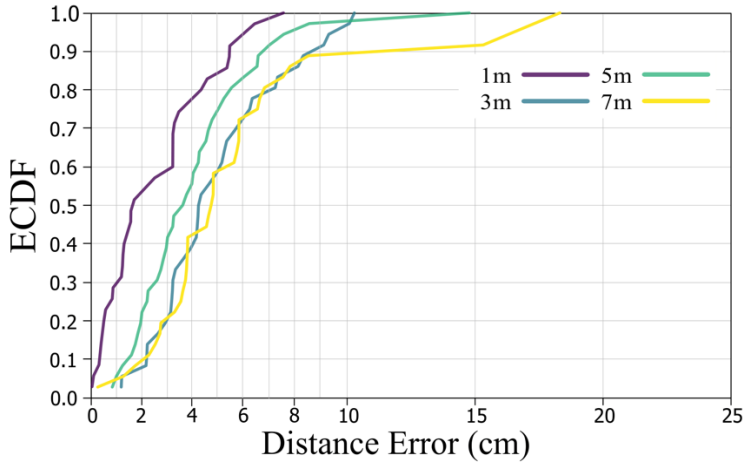


Figure 6-16 ECDF for the FTM distance errors [MGB+23]

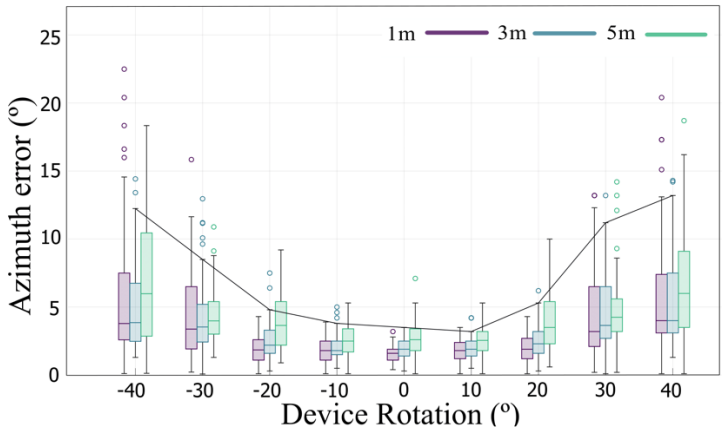


Figure 6-17 Azimuth errors for different rotation [MGB+23]

6.4.4 Summary

Advanced algorithms and technologies for BCS data processing will be required. To this end, application function optimization will be critical depending on the availability of computational and communication resources, as well as potential security/privacy constraints. Efficient resource allocation will be required for the system to be able to accommodate diverse use cases, with different BCS architectures and QoS characteristics. Also, new device capabilities, such as new UE roles and responsibilities in JCAS, coordination with the infrastructure in order to share the self-sensing information, new radios that can provide frequent sensing capabilities will be required. Table 6-4 summarizes the main benefits and implications of the *Application-/Device-specific BCS* enabler.

Table 6-4 Application-/Device-specific BCS data consuming functions summary table

Description	Application-/Device-specific BCS data consuming functions: Mechanisms for optimising the placement and resource allocation to BCS data consumers (Beyond-comm functions, network-centric application components, etc.)	
Benefits	**KPI improvement**	Data privacy, security, trustworthiness, integrity E2E computation and communication delay, network load, BW/energy efficiency Number of services enabled in the network with optimized support JCAS capabilities optimisation: angle estimation, distance estimations, sensing accuracy.

	Design principles [HEX223-D21]	#4: Scalability to adapt to a range of devices and AFs. #1: Efficient data management and exposure for various Afs #6: Emphasis on security, privacy, and trustworthiness in data handling #10: Sustainable and energy efficient in network operation
	Dependencies / Basis for another enabler	Exposure and data management Protocols, signalling and procedures Distributed Compute-as-a-Service, AIaaS, Intent based Management (Zero Touch), E2E context awareness management, MLOps, DataOps
Implications	**Requirements**	Advanced algorithms and technologies for data processing and application function optimization Robust mechanisms for data exposure, thus privacy and security Efficient resource allocation strategies for diverse use cases New UE roles and responsibilities in JCAS, coordination with the infrastructure in order to share the self-sensing information, new radios that can provide frequent sensing capabilities. Quantum sensing and computing
	Standard relations & regulations	3GPP TS 23.501 System architecture for the 5G System (5GS) 3GPP TS 23.502 Procedures for the 5G System (5GS) 3GPP TS 23.503 Policy and charging control framework for the 5G System (5GS) 3GPP TS 23.288, Architecture enhancements for 5G System (5GS) to support network data analytics services IEEE 802.11 az
	Required resources	Computational resources for data processing and AI/ML tasks Infrastructure capable of handling diverse and dynamic data streams UE with multiple mmWave radios Modular and scalable infrastructure capable of handling diverse and dynamic data streams

7 Virtualisation and Cloud transformation

The off-the-shelf cloud i.e., Amazon and Microsoft are suitable for a big subset of multimedia human-scale applications, but it has its limitations when it comes down to supporting the upcoming latency sensitive 6G use cases. This document describes an initial concept on how to transform the cloud, so it fits applicable 6G requirements such as management, latency, security, and connection reliability. Four architectural enablers for the transformation of the virtualisation and cloud were identified in [HEX223-D32], namely "Integration and orchestration of cloud continuum resources", "Multi-domain/Multi-cloud federation", "Network module placement" and "cloud transformation in 6G-quantum architecture", along with the identified problem statements related to each of these enablers. In this deliverable, the "Network module placement" enabler and the "Integration and orchestration of cloud continuum resources" enabler are merged under a single enabler named "Integration and orchestration of extreme edge resources in the compute continuum". The architectural proposals required for the successful operation of the defined enablers, as well as their initial workflows and evaluations are presented. Each architectural enabler is mapped to the 6G E2E system blueprint of Figure 7-1 [HEX223-D22], as explained in the introduction of each enabler's section.

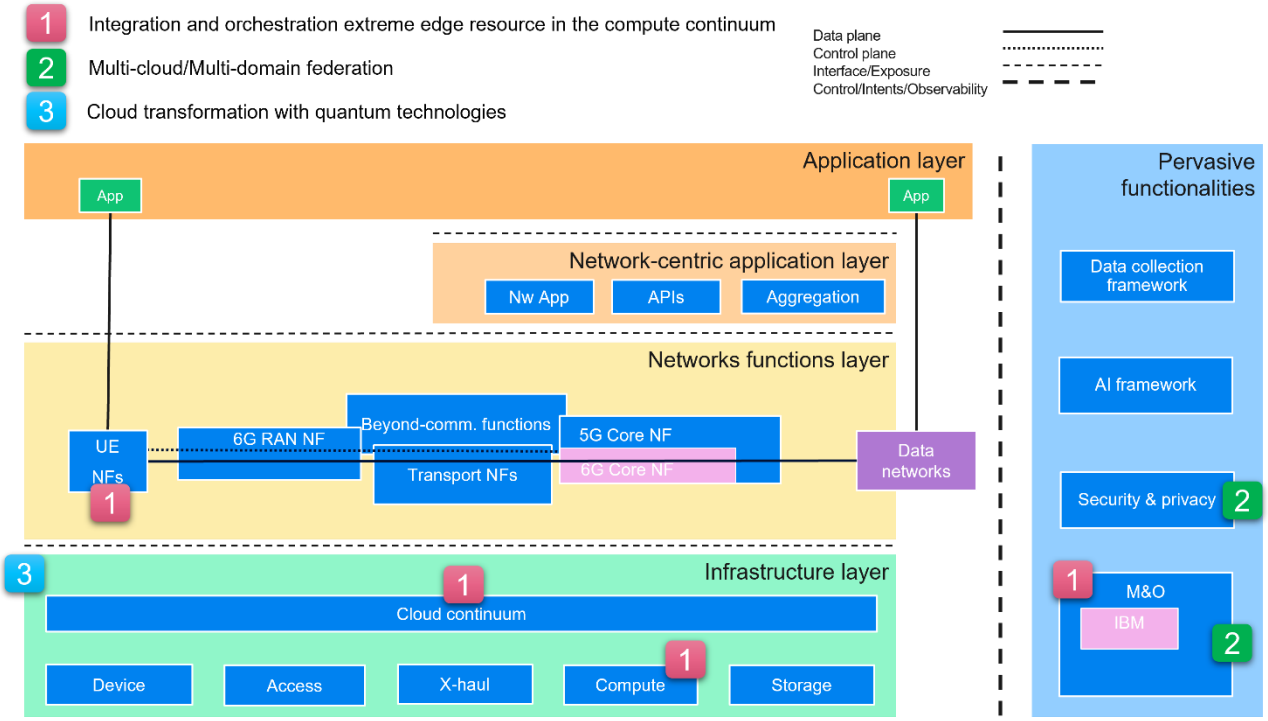


Figure 7-1 Mapping of the virtualization and cloud transformation enablers to the 6G E2E system blueprint of [HEX223-D22].

7.1 Integration and orchestration of extreme edge resources in the compute continuum

7.1.1 Introduction

The integration and orchestration of extreme edge resources in the compute continuum define the needed architectural interfaces and components for seamless orchestration and management of compute resources that are placed beyond the radio access part of the network. Figure 7-2 shows the extended cloud continuum that is composed of powerful centralized cloud resources, distributed edge resources, and heterogeneous, mobile, and volatile extreme edge resources. The integration of such extreme edge resources brings a set of new key challenges that are studied in this section and introduces the concept of decentralized end-to-end orchestration as an alternative to the MNO-centric Management and Orchestration (M&O) approach that was already introduced in the 5G networks. Finally, such an extended cloud continuum could be used to run vertical

applications, but also network modules. This enabler defines a new split of the user plane function that can be used to handle the traffic at the vertical site.

Section 7.1.2 states the identified architectural implication of this enabler, while Section 7.1.3 presents initial evaluation performed in the context of this enabler. Finally, Section 7.1.4 summarizes the benefits and implications of this enabler.

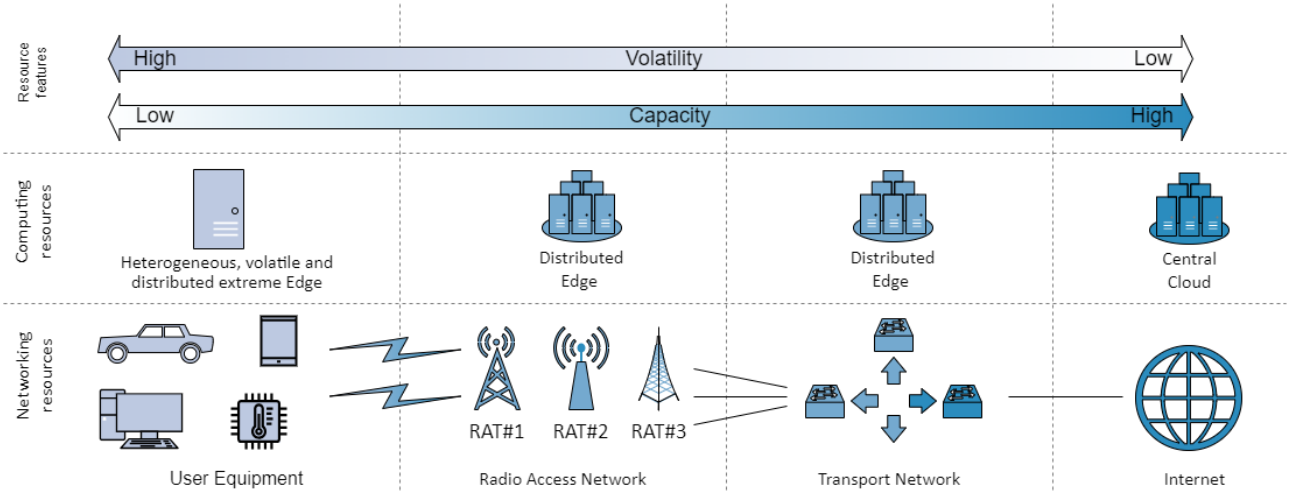


Figure 7-2 Volatility and Capacity of the compute and network resources in the compute continuum

7.1.2 Architectural modifications

ETSI developed one of the first edge computing frameworks called MEC [MEC003] that was originally conceptualized as edge computing platform that telecom operators and verticals can use to relieve end-devices from local task processing by offloading computation to the virtualized platform. The prevailing deployment choice for MEC involves Network Function Virtualization (NFV) [NFV002] and enables application developers and content providers to leverage cloud computing capabilities at data centres located at the edge of the network. The computing resources beyond the Radio Access Network (e.g., user equipment's) offers diverse computing and data capabilities, presenting a potentially valuable addition to conventional MEC systems for supporting latency- or data-sensitive applications. However, these resources primarily consist of devices with restricted computational capabilities with respect to data centres, sometimes being battery-powered and mobile. This limitation complicates the direct support of comprehensive MEC solutions.

In [RGO+23] 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. This conclusion is supported by the research on inter-MEC system connection and federation [MEC035], The MEC Orchestrator (MEO) is regarded as the crucial facilitator for numerous workflows, although cMEC might lack support for it. However, a particular cMEC has the

option to share various resources with different tMECs by leveraging its orchestrating abilities, or even with peer cMECs. Dependency on a tMEC System: In instances where the cMEC lacks the implementation of a particular MEC function, it must depend on the upper layer tMEC system to provide the lacking functionalities. New workflows, guidelines for MEC application development, and specific interfaces must be established to address the deficiency in functions.

- **End-User Device Co-location and Awareness:** The cMEC system has the flexibility to either operate within the same end-user device as the MEC application or to run on a **constrained device in** its immediate vicinity. The end-user device can engage in the cMEC integration in the following manner:
  1. cMEC-aware: the end-user device and cMEC are either within the same local network or have knowledge of each other's identity (e.g., the cMEC operates on that specific end-user device). The end-user device has the capability to examine the available cMEC systems and request the deployment of a MEC application, leading to the establishment of a connection between the cMEC and a tMEC.
  2. cMEC-unaware: The end-user device lacks awareness of any nearby cMEC and consequently requests the deployment of a MEC application directly to the tMEC. However, upon recognizing the presence of a nearby cMEC deployment, the tMEC opts to instantiate the application on the interconnected cMEC.

Given the aforementioned points, Figure 7-3 details the architectural implications to interconnect the cMEC with the tMEC, without the MEO being present in the cMEC system. The main components of the system are the virtualization infrastructure that hosts the MEC applications (MEC App) and the Multi-access Edge Platform (MEP). The virtualization infrastructure furnishes computing, storage, and networking resources for MEC applications, encompassing a data plane. for routing the traffic among applications, services, local/external networks, and mobile edge platform. The mobile edge platform offers a setting in which MEC applications can discover, advertise, consume, and offer mobile edge services. Finally, MEC applications run on top of the virtualization infrastructure provided by the mobile edge host and can interact with the mobile edge platform to consume and publish mobile edge services via the Mp1 reference point. How these MEC applications retrieve the data to be published as a service is left unspecified by current ETSI MEC specifications. ETSI MEC defines a set of exemplary services. For example, the Radio Network Information service (RNIS) provides radio network-related information, such as up-to-date radio network status, metrics, and statistical data pertaining to the user plane, and information related to users served by the radio nodes. At the system level, an operational support systems (OSS) tool usually oversees the initiation and cessation of MEC applications as requested by a user application lifecycle management (UALCMP) proxy. These requests are received from either an end user or a customized portal. The presence of a MEC orchestrator (MEO) affords a comprehensive view of the entire MEC system, facilitating package onboarding and determining the most appropriate host for deploying the application.

At the host level, the MEC platform manager (MEPM) directly manages the lifecycle of applications while configuring traffic, security, and DNS rules based on the application's requirements. Meanwhile, the MEC Execution Platform (MEP) provides the environment for MEC services to MEC applications and implements DNS and traffic control rules for these applications. Eventually, the computational, network, and memory resources of the platform are, managed by the virtual infrastructure manager (VIM).

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 Operations Support System (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.

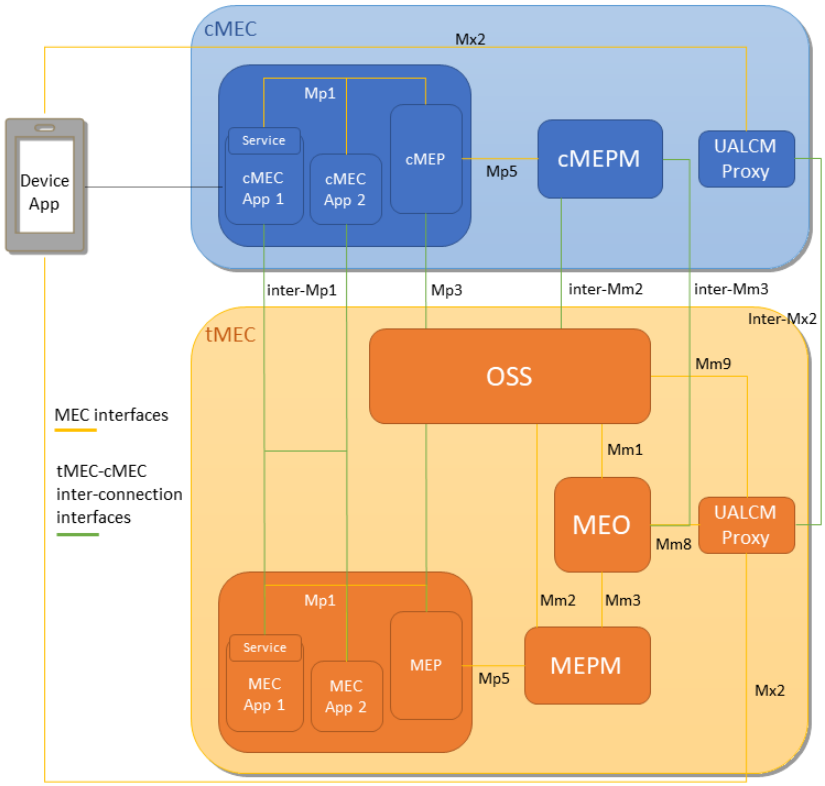


Figure 7-3 Architectural scheme of constrained MEC together with traditional MEC [RGO+23]

The extended cloud framework that includes the constrained devices is enhancing the traditional cloud capabilities and could be used to run not only vertical applications, but also network functions. Figure 7-4 illustrates a scenario of an industrial factory, where robots execute different tasks, and are connected to a local network. The factory is equipped with local compute servers (extreme-edge, Far edge and Cloud) and is also connected to a CSP. The underlying applications are associated with different requirements, ranging from very strict delay requirements to medium delay requirements. As the extreme-edge cloud could suffer from scarcity of compute resources, some vertical applications could dynamically be deployed at the edge cloud and the extreme-edge cloud. Note that optimal placement of such applications would require coordinated processes between the M&O framework of the local virtual network and the CSP, by considering their network and compute capabilities. Consequently, this would require functionalities at vertical site to handle user traffic and to steer it locally or to the CSP.

Handling user traffic is performed by a network module, which is UPF (User Plane Function). Different UPF types with specific functionalities are being standardized, including Uplink Classifier (UL-CL) and Intermediate UPF (I-UPF). These network modules are deployed in a chained fashion to ensure a common objective. For instance, an UL-CL can be inserted to steer a specific traffic to a local edge, while routing the remaining traffic to a central UPF anchor. In order to handle user traffic at the vertical site, a UPF type network function that is called Sub-UPF (as it operates in a sub-network environment) could therefore be deployed for this purpose. Sub-UPF (which could be owned by the vertical) can therefore be customized and configured to steer user traffic as per the location of the target application. Ensuring efficient user traffic steering in a way to reach the target QoS would require optimized placement of the chained user plane functions across the cloud continuum.

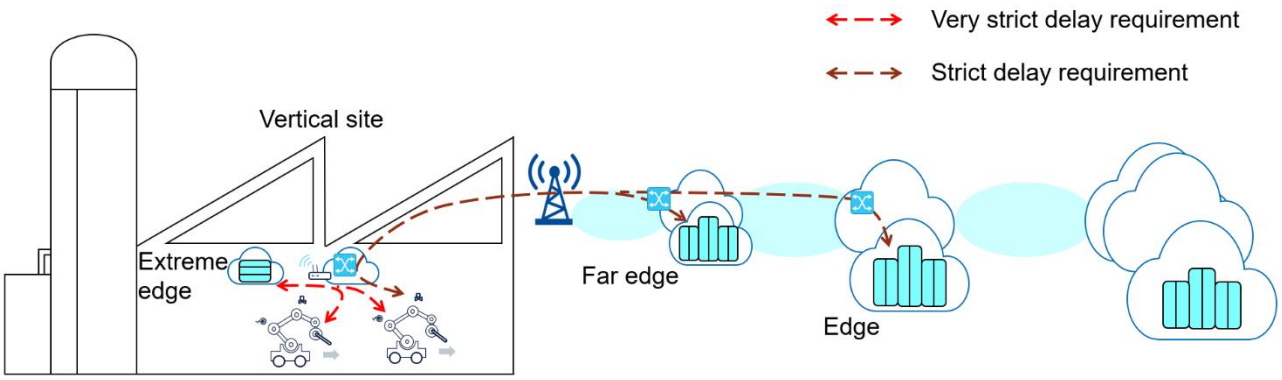


Figure 7-4 Scenario for handling user traffic in cloud continuum environment

The management and orchestration of the network functions like the Sub-UPF at the vertical side needs to (i) take into account heterogeneous virtualization platforms (e.g., Kubernetes, K3s, Microk8s, OpenStack, etc.) and extreme edge devices (e.g., IoT devices, Sensors and Actuators, Robots and Cobots, etc.), (ii) improve the placement mechanisms to efficiently deploy, migrate and distribute network functions that have proximity constraints (e.g., applications components that need to be executed near data sources to fast react to related events) and/or requirements exposing a unified interface, and (iii) to automate the discovery mechanisms of virtualization platforms and extreme edge devices information. To support such M&O framework a driver-based approach can allow the orchestrator to retrieve the platform specific computing capabilities and device specific information through per-platform and per-device drivers. These drivers, implemented as specific resource management software components that execute workflows within the M&O and share the same interface, are managed by a platform and device agnostic manager to perform the dynamic discovery, continuous monitoring, and inventory of the resources across the Extreme-Edge, Edge, and Cloud Continuum. The driver-based approach designed for the compute continuum M&O allows to plug and unplug specific drivers, even dynamically at runtime, depending on the scenario and on the types of the devices involved. Adding a new compute resource (i.e., that it could be hosted in a device) does not introduce any signalling cost to the discovery and monitoring processes since the new compute node join operation (e.g., to a Kubernetes cluster) is performed by the legacy virtualization platform-specific procedures (e.g., the Kubernetes ones), with no impact to the above-mentioned workflows.

Figure 7-5 depicts a possible architecture of the compute continuum M&O that highlights the extreme edge, edge and cloud resources discovery, monitoring, and inventory workflows as well as the virtualized services orchestration workflow.

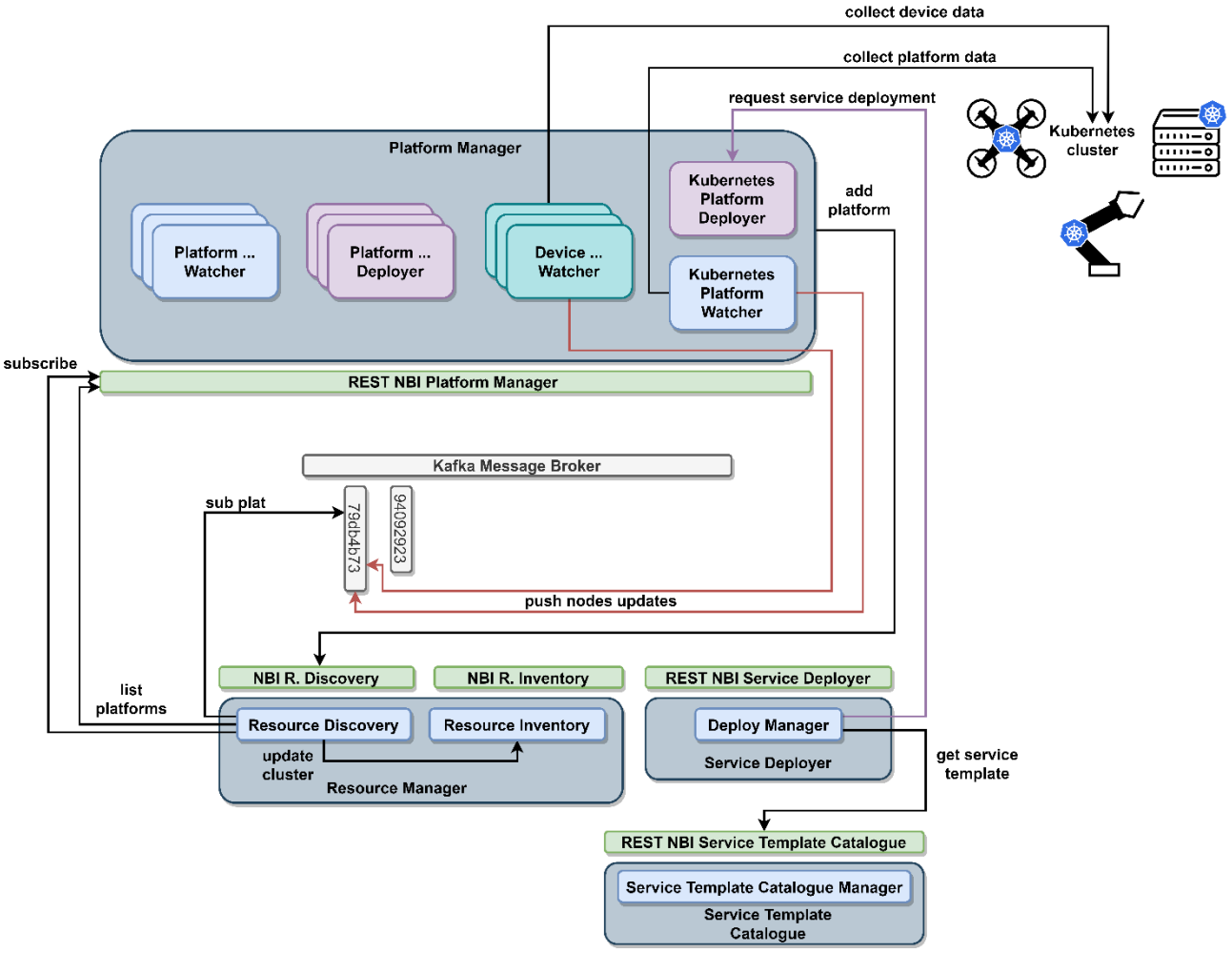


Figure 7-5 High level software architecture of the compute continuum M&O

The platform has been designed to be a multi-module architecture where each sub-module (in grey in the Figure 7-5) contributes to the functionalities of the orchestrator.

- The Platform Manager enables the dynamic discovery and continuous monitoring of the resources across the extreme edge, edge, and Cloud Continuum through the management of platform-specific and device-specific drivers, embedded in an agnostic-designed skeleton, which can be develop, plugged, and unplugged depending on the underlying virtualization infrastructures and the involved devices. It also takes care of the virtualized applications orchestration operations by the means of platform-specific drivers.
- The Resource Manager module registers to the Platform Manager to receive updates on the discovered and monitored virtualization platforms (along with device-specific information) to expose a unified interface to retrieve information about the extreme edge, edge and cloud continuum resources and thus enable the inventory functionality of the platform making use of different information models.
- The Service Deployer module exposes a unified interface to manage platform-agnostic virtualized service applications orchestration requests, leveraging the different information model, and translates them in the platform-specific orchestration requests that can be handled by the platform-specific drivers available in the Platform Manager to perform the orchestration operations requested. The Service Deployer works alongside with the Service Template Catalogue to retrieve the platform-specific orchestration templates to be used to fulfil specific orchestration operations.

The introduction of such a new M&O platform is focused on going beyond monolithic approaches of standards solutions like ETSI NFV MANO and aims at clearly separating the orchestration of the (legacy) network services from the orchestration of the vertical applications. In particular, it introduces a fully cloud-native

approach, with transparent integration of extreme-edge resources, to address the strict requirements of the application layer domain (in terms of deployment constraints, dynamicity, communication patterns, etc.).

If we consider standards solution like ETSI Network Functions Virtualization Management And Orchestration (NFV M&O) the introduced platform can play the role (i.e., be mapped in the ETSI NFV M&O architecture) of a particular enhanced VIM due to the fact that its functionalities are mainly focused on the management of the resources of the extreme-edge, edge and cloud continuum layer (infrastructure layer) yet, at the same time, it offers advanced functions like the management of multiple domains, the dynamic discovery of the infrastructure layer resources and their continuous monitoring, and the possibility to orchestrate service application over the managed resources. In the context of the Hexa-X-I M&O framework [HEX22-D62] the highlighted M&O platform could fit in as infrastructure layer M&O since it covers all the requirements needed by such entity (functionalities of the infrastructure layer M&O) providing, at the same time, advanced functionalities (i.e., multi-domain support, dynamic discovery of continuum resources, etc.)

In the same line of going beyond the legacy monolithic MNO-centric M&O approaches, another alternative architectural modification towards 6G would involve the development of the *decentralized M&O* concept. This approach is considered to have the potential to solve the new set of challenges with respect to the aggregation of devices beyond the MNO own premises mentioned above, i.e., the diversity of stakeholders in this domain (vertical industries, hyperscalers, end-users…), the high heterogeneity of devices (including reduced capability or battery power devices,), the potential high volatility of those devices (they could unexpectedly move, drastically change their available computing or processing resources or even be unexpectedly disconnected), and the size of this domain, that can be huge in scale. In this regard, Appendix 11.4 describes how a decentralized M&O approach could be implemented. This approach relies on the deployment of multiple instances of a reduced set of network elements that would be distributed through the entire network continuum, so addressing the M&O problem considering the network as a diverse ecosystem of resources and stakeholders (as it is envisaged towards 6G), rather than approaching it only from a single MNO perspective.

7.1.3 Preliminary workflows and evaluation

This section presents some preliminary workflows and interactions of some of the features of the integration and orchestration of extreme edge resources in the compute continuum. The general idea that the architectural implications have been presented in the previous section are elaborated and evaluated in detail here. The driver-based M&O approach has three main functions, namely: (i) the dynamic discovery, (ii) continuous monitoring and inventory of the compute continuum resources and (iii) the virtualized service applications orchestration.

The dynamic discovery, continuous monitoring, and inventory of (i.e., extreme-edge, edge and cloud) continuum resources functionalities, depicted in Figure 7-6, are fulfilled by the Platform Manager and Resource Manager modules of the M&O Platform as described in the previous section. In the first step of the workflow the Resource Manager performs a subscription to the Platform Manager to keep track of the new clusters (i.e., and their resources) that could be onboarded to be monitored in the future. Next, the Resource Manager queries all the available platforms from the Platform Manager to start receiving updates for them; the updates are produced by platform-specific and device-specific drivers of the Platform Manager that collect the platform and device information and characteristics from the target platforms. The workflow in Figure also details the operations executed by the Platform Manager when a new platform is onboarded to be monitored. Upon receiving a new platform, the Platform Manager spawns dedicated drivers to monitor the specific information of the underlying resources. As the last step the Platform Manager notifies the Resource Manager of the newly added platform so (i.e., the Resource Manager) it can keep track also of these resources in its inventory.

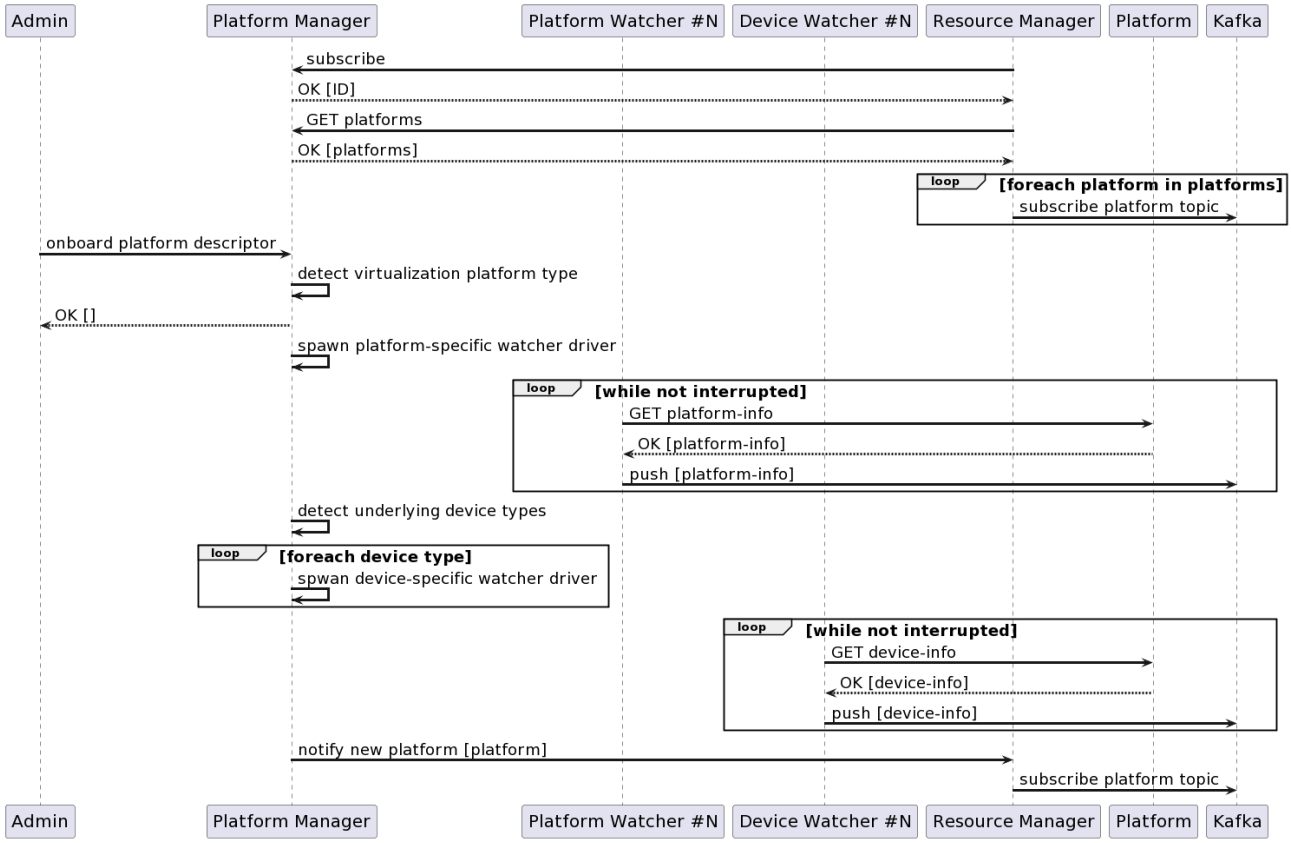


Figure 7-6 Dynamic discovery, continuous monitoring, and inventory workflows

The virtualized service applications orchestration workflows are depicted in Figure 7-7. The involved modules of the M&O Platform that fulfil these functionalities are the Platform Manager, with its platform-specific deployment drivers, the Service Deployer, the Resource Manager, to retrieve the deployment target platform information, and the Service Template Catalogue to retrieve the templates to be used to deploy or update a given virtualized service application. In the deployment workflow the Service Deployer module of the M&O Platform inspects the service components of the received deployment request, on its Northbound interface, and, for each of the requests, retrieve the specified orchestration template from the Service Template Catalogue. Next, it proceeds to customise the template and produces the platform-specific service component deployment request that will be handled by a platform-specific deployer (driver) of the Platform Manager to execute the deployment of the application component in the target virtualization platform.

The service application update and delete workflows are very similar to the deployment one. In the update workflow the request includes the identifiers of the service components that need to be updated and the new configuration that need to be applied: the involved modules of the M&O Platform are the same as in the deployment one. In the delete workflow only the identifiers of the service components that need to be terminated are specified. It is worth mentioning that all the components of the M&O Platform and workflows highlighted in these paragraphs have been tested and integrated in an internal lab environment and in the PoC B testbed.

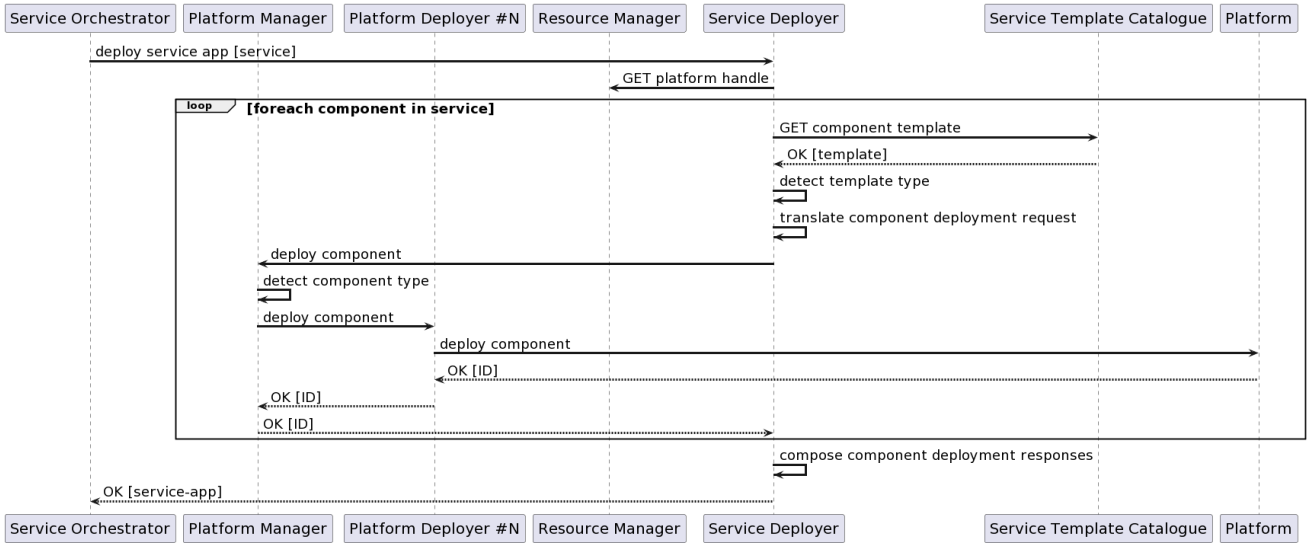


Figure 7-7 Platform-agnostic service applications orchestration workflow

Regarding the decentralized M&O approach also introduced in the previous Section 7.1.2, Appendix 11.4 introduces the high-level workflow showcasing the overall operation of the concept. Regarding its evaluation, works targeting some of the technical challenges associated to this concept are being addressed in the context of the System-PoC B, and are planned to be reported in Deliverables D2.4 and D2.6, reporting the interim and final end-to-end system evaluation results, respectively.

7.1.4 Summary

As stated in [HEX223-D32], integration and orchestration of the extreme edge resources in the compute continuum enables the M&O capabilities of compute resources beyond the radio access part of the network. It provides the needed architectural components, interfaces, and mechanisms to orchestrate and manage the volatile and resource constrained extreme edge devices as part of the compute continuum. This enabler directly contributes towards the fulfilment of the improved QoE, seamless service continuity, efficient M&O, and privacy protection requirements of all the six use case families defined in WP1 [HEX223-D12]. For example, in the Immersive Experience, Trusted Environments and Fully Connected World Use Case Families this enabler contributes towards the improvement of the privacy and security protection. Different applications from the above-mentioned use cases will be allowed to handle their sensitive information (e.g., eHealth) in the device where it has been generated. In addition, in the Collaborative Robots and Digital Twins Use Case Families this use case contribute towards improved M&O and service continuity capabilities by providing mechanisms for flexible resource inclusion and allocation in the compute continuum.

This enabler has implications on the infrastructure layer of the 6G E2E system blueprint where new interfaces and mechanisms need to be defined for: 1) exposing the capabilities of the extreme edge devices and 2) for interactions between the extreme edge, edge, and cloud resources. Moreover, implications on the Management and Orchestration component of the 6G E2E system blueprint are expected by defining new end-to-end multi-technology resource orchestration concepts to orchestrate resources over the volatile and constrained extreme edge devices. Table 7-1 shows a summary of the main benefits and implications of the Integration and orchestration of extreme edge resources enabler in the E2E 6G system blueprint that is under development in the project.

Table 7-1: Benefits and implications of "Integration and orchestration of extreme edge resources" enabler

Description	Interfaces, architectural components, and mechanisms to allow the integration, orchestration and management of the extreme edge devices into the compute continuum.	
Benefits	**KPI improvement**	Latency, energy consumption, service availability, increase data privacy.
	Design principles [HEX223-D21]	Full automation and optimization (#2)

		Flexibility to different scenarios (#3) Internal interfaces are cloud optimized (#7)
	Dependencies / Basis for another enabler	Programable network monitoring and telemetry Orchestration mechanisms for the computing continuum Programmable and flexible network configuration
Implications	**Requirements**	New UE roles and responsibilities, coordination and interaction of the UE with the edge and cloud, communication between functions deployed in the edge/cloud and the UE, UE resource orchestration and management.
	Standard relations & regulations	3GPP TS 23.558 [23.558] ETSI GR MEC 036 [MEC035]
	Computing resources	Cloud computing, edge computing and UE computing resources will jointly automated and optimized.

7.2 Multi-domain/multi-cloud federation

7.2.1 Introduction

Multi-domain/Multi-cloud federation is the capability to aggregate cloud services provided by multiple domains and providers into a single, coherent cloud. The federation concept is not only about the capability of spanning across the administrative domains of different legal entities (e.g., network operators, nations), but it is also strictly related to the multi-cloud capability, i.e., the aggregation of underlying cloud resources built on different cloud technologies, including private and public cloud.

In 5G, federation is mainly focused on sharing cloud/compute resources. The key concept is that in a 6G system, federation should extend the basic 5G federation concept, providing a joint offering of cloud and network services alongside with beyond communication services (e.g., sensing). From an operator point of view, this facilitates the seamless deployment of telco and non-telco applications, aligning with the "Beyond communications" paradigm. From a customer point of view, users can expect a consistent level of service across different operators and countries within the federated cloud. This can be achieved via network services pairing across federated domains allowing easy portability of applications between these domains. The architectural solution has three key innovations points that give value to a future implementation:

- **Native** federation: In 5G system there is no native concept of federation. Federation mechanisms (e.g., GSMA OPG) in 5G are standardized in objects that are outside the 5G system M&O layer. This leads to higher costs of integration and maintenance and loose coupling between the 5G system and those external objects. The requirement is to have a direct management of the federation in the M&O layer of the 6G system (possibly by means of a dedicated module of a modular M&O layer).
- **Broader resource sharing via federation**: 5G federation concept is focused on aggregation of cloud resources. The whole federation and related brokering systems are designed to offer cloud services. The proposal it to redesign federation concepts by including network services and beyond communication services (e.g., sensing), the latter of which is a key concept in HEXA-X-II project (see Chapter 6), allowing the federation customers to benefit from an aggregated cloud, network and beyond communication service. As an example, the integration can lead to optimized and coordinated services such as coordinated edge roaming: local breakout of data, intrinsic to network technology should be coordinated with application mobility, in order to grant seamless service to users across the federations (coupling of network local breakout and mobility of application is not natively synchronized in 5G system). The requirement is the evolution of federation mechanisms and interfaces towards the integrated cloud, network and beyond communication services.
- **Intent-based interfaces**: Intend-based management is one key enabler of Hexa-X-II [HEX223-D22]. We are depicting an architecture in which all federation interfaces (including the interfaces to customer

and the interfaces to lower layer services to be shared in the federation are intent based. This leads to a federation of services rather than a federation of bare cloud resources.

Section 7.2.2 categorizes the federation approaches that can be followed in a 6G architecture and describes architectural implications of the enabler for a specific approach. Section 7.2.3 offers a look on the initial evaluations obtained for this enabler and finally, Section 7.2.4 summarizes the concepts of this enabler organizing them in a table.

7.2.2 Architectural modifications

Multi-domain/Multi-cloud federation greatly impacts the Orchestration functionality since 6G M&O layer needs to encompass a dedicated module to manage federation. Federation models can vary depending on how the relation between M&O Layer and services is implemented. The following models have been identified:

- **Direct model**: Cloud services directly communicate with M&O layers of different Network Operators, regardless of the domain in which they are deployed. This model is the simplest one, but it has limited scalability.
- **Cloud service broker model**: A common resource layer acts as a single Cloud service broker for all federated network operators. All Operators can exploit Cloud services from all the federated domains. This model has higher scalability than the first one, but it intrinsically has a governance problem (i.e., who owns the service broker)
- **Peering model**: A dedicated peering interface is established between 6G M&O layers of Network Operators. Federation interfaces manage both business and technical aspects. Beside Cloud Service from federated operators, other cloud resources from external cloud entities can be aggregated and exposed. This model offers good level of scalability without relying on a centralized service.

Regarding the interfaces, intent based APIs layer should be able to aggregate cloud services, network services, and beyond communication services provided by the operator and by all federated partners. The intent of services shall encompass the combination of all three types of services, i.e., with one intent-based API call, interfaces of the different services can be called. Finally, security and privacy network functions and policies should be extended to also consider the security related aspects of federation interfaces and federated resources.

The federation multi-domain aspect of data centres is a critical factor in the evolution of the current state of the deployment architectures for large-scale city-wide or even country-wide deployments. However, we must look at more than just data centres for their traditional factor, i.e., a large pool of computing resources allowing big on-prem implementations. We are to look at them as a federation of nodes that also includes the edge nodes scattered around the city and, with the advent of 6G, the extreme edge nodes that can also be powerful computational units. Hence, the workloads that will be running shall be cloud-native and be able to bootstrap themselves on different nodes, regardless of their nature (e.g., CPU architecture). On the other hand, nature will be of utmost importance when choosing where to run a specific workload to benefit from the nature of the data centre. The critical factor is ensuring that the underlying infrastructure is homogeneous and based on cloud-native technologies so that an orchestrator can reach it and choose to deploy specific workloads there.

Therefore, the concepts that we discuss here, in a 6G environment, will entail the extension of the cloud principles until the edge and the extreme edge. In this way this extension aims to foster a base configuration for the Advanced Orchestration and Management of the resources in every node of the federation ensuring an efficient resource allocation and utilization. The architecture should also amplify the lifecycle management of the applications using Life Cycle Assessment tools and solve the 3GPP premises while adding a flavour for the correct placement of the workload or resource. Finally, access to such a robust network of computing nodes will lower the overall cost of running an application, since the necessary resources to perform the tasks can be deployed across the different nodes. The 6G ecosystem needs to evolve towards amplifying the M&O capabilities of the 5G ecosystem (and previous ones), making sure every node (cloud, edge, extreme edge, on-prem, etc.) has the same bootstrap recipe and the exact implementation principles for it to be orchestrated seamlessly.

Such aspects are directly related to the heterogeneous integration of diverse technologies while optimizing the latency requirements (critical vs sensitive should be one of the placement decisions for the app). Moreover, the federation allows for “free” distributed intelligence with by design privacy since all nodes are on a

federation and (should) trust each other. Lastly, it touches on massive connectivity since all nodes must expose the same standard APIs for better interoperability while fostering new topologies for in situ processing or running concurrent tasks on energy-optimized nodes.

The direction of the 6G ecosystem for the Multi-Domain Data Centre should treat it as another federated cloud node with specific characteristics that can confer its advantages over other nodes on the network. The app developer or resource holder should be on the lookout for making the best product possible and then tell the 6G ecosystem the network requirements for their developments. The network should decide where to place the workloads that makeup such a product.

The following paragraphs present a modification of the ETSI NFV MANO framework using the **Cloud service broker** federation model.

Network virtualization typically applies ETSI NFV specifications to orchestrate Network Services (NS) [NFV002] and assumes that the orchestrator operator is also owner of the infrastructure therefore no business interfaces to the NFVI, composed of NFVI-PoPs [INF001] are defined. The multi-provider approach, however, provides numerous benefits, such as reducing the cost of the resources by selecting the cheapest provider, extending the geographic infrastructure span, increasing infrastructure capacity, and contributing to higher infrastructure reliability. Moreover, the NS security can be improved, as each provider may handle only a part of the NS. The concept called Cloud Continuum [HEX22-D13] assumes integration and uniform exposure to applications of all available resources. It includes the unreliable and constrained devices, i.e., the resources of IoT nodes or smartphones, and the resources of NTNs, of which some, like Low Earth Orbit satellites (LEOs) or UAVs, may be short-lived.

In this approach, a modification of the ETSI NFV M&O framework allows for dynamic interconnection of data centres of various infrastructure providers and exposing the aggregated resources to services. The key component of the concept is the resource layer that handles all infrastructure-related operations and acts as a proxy between Modified NFVOs (M-NFVO) and Modified VIMs (M-VIM). The modified ETSI NFV M&O components, ecosystem actors, and infrastructure components interact with resource layer via secure interfaces, as illustrated in Figure 7-7. For simplicity, it is assumed that Edge Cloud data centres and the Central Cloud data centre provide computing, storage and intra-data centre networking virtualization, as well as mechanisms to cope with faults and load balancing. Each data centre is described by the Data Centre Features (DCF), which consists of, among others, parameters concerning data centres' geographic location, total capacity, delay of links between this data centre and other data centres, and resources cost. Some data centre parameters such as resource consumption, energy consumption, power status (in the case of battery-powered data centres), and estimated reliability are updated by data centres or resource layer in real time. In the case of mobile data centres (LEOs, UAVs, cars, pedestrians), details concerning data centre mobility patterns are also provided or calculated by the resource layer; therefore, DCF supports the Far-Edge resources usage. The main features of the proposed approach are the following:

- Resource layer has mechanisms for dynamic adding and removal of data centres of multiple Infrastructure Providers using secure interfaces. The dynamically updated DCF describes each data centre.
- Exposing all resources to orchestrators would make the orchestration not scalable; therefore, the resource layer exposes to orchestrators only a Partition of Infrastructure resources (NS-PoI) with its topology. The NS-PoI is created using NS requirements that may include data centre location, cost, energy efficiency, reliability, inter-data centre delay, etc. Please note that the resource layer computes the partition topology; this is no more the role of the orchestrator.
- Based on the requirement of each NS (for example, location, inter-data centre delay), the resource layer may group several data centres to create an Aggregated Data Centre (ADC). The ADCs simplify NS-PoI topology and stabilize it (in the case of Infrastructure dynamic changes), improving that way orchestration scalability.
- The resource layer supports multiple orchestrators that are no longer linked with a dedicated Infrastructure domain; each handle dynamically created, NS-specific NS-PoI. The orchestrators for specific NS or a set of NSs can be orchestrated.
- The resource layer can autonomously trigger the VM migration within an ADC in a way invisible to orchestrators. The operation can be triggered by ADC modification (adding or removing a data centre

from ADC). In case when an isolated data centre is added, the resource layer updates NS-PoIs if needed.

- The resource layer has resource-oriented functions that enable not only exposure of information about resource consumption but also resource consumption predictions, reliability estimation of a data centre, etc. resource layer internal components (described below) interact using a message bus. New resource layer components can be orchestrated, for example, to support mobile Infrastructure nodes (prediction of mobile nodes' position), Far-Edge, NTN, resource brokering, etc.

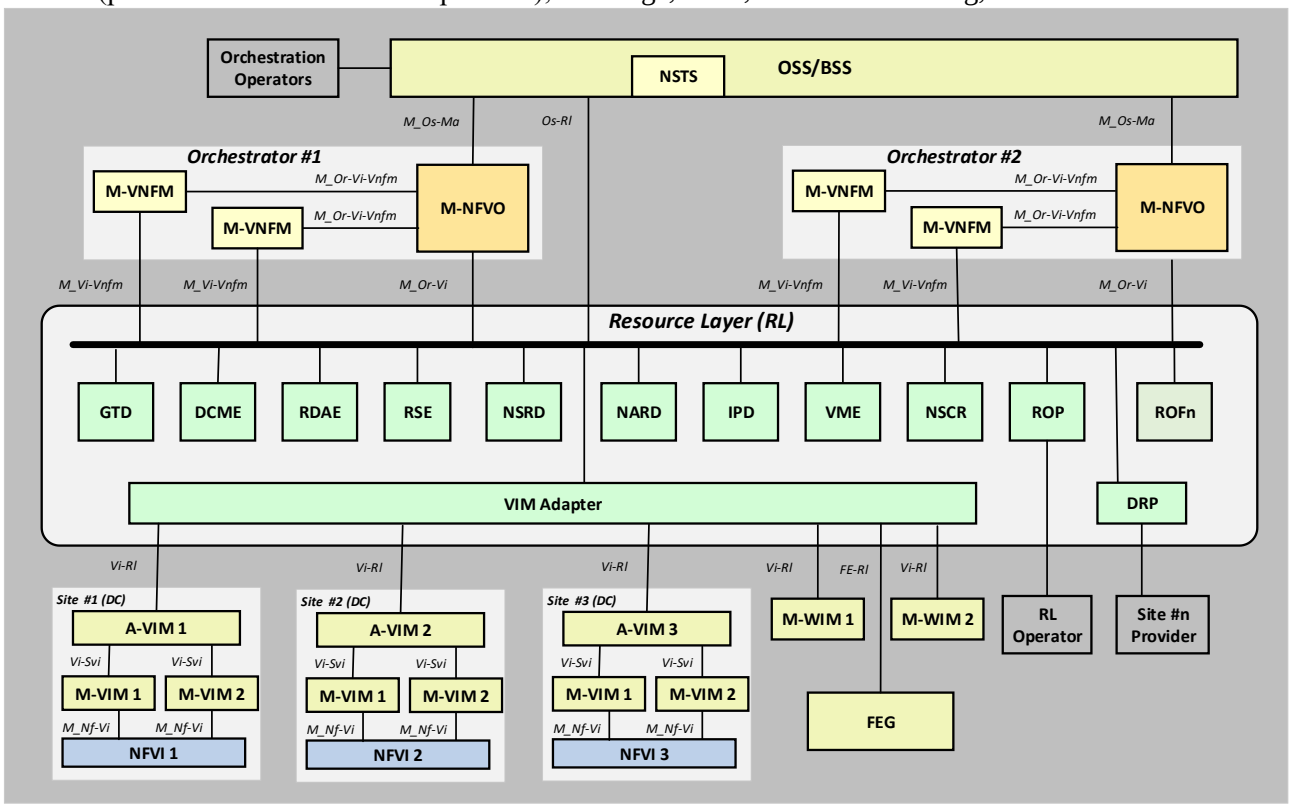


Figure 7-8 Multi-provider based Cloud Continuum approach.

The overall architecture of the concept is presented in Figure 7-8. The resource layer provides secure interfaces towards data centres and Orchestrators. The interfaces are an extended version of the existing VIM, VNFM and VNFO interfaces. Their modifications concern security mechanisms supporting mutual authentication, encryption, and data integrity checking. Moreover, the exchange of the DCF parameters between the resource layer and modified VIMs and NFVOs (called M-VIM and M-NFVO) is supported. In the case of NFVO, the modifications allow passing to resource layer requirements concerning resources to create NS-PoI and the ability to use NS-PoI and exploit the DCF parameters. The internal components of resource layer are the following:

- *Global Topology Database* (GTD) keeps the topology of all connected or being able to be connected DCs. For each DC, the DCF is recorded in GTD and periodically updated by DCME.
- *Data Centre Monitoring Engine* (DCME) evaluates in real-time each data centre status, i.e., energy consumption, performance, and reliability, updating its DCF.
- *Resource Layer Data Analytic Engines* (RDAE) predicts resource usage of all data centres and their related indicators (energy, reliability, etc.). M-NFVO can use this feature to optimize the decision about VNF placement and for decisions concerning VM migration.
- *Resource layer Security Engine* (RSE) provides mutual authentication of framework entities and operators. It enforces encrypted communication with integrity protection between the resource layer and other entities.
- *NS Requirements Database* (NSRD) consists of, provided by each NS, requirements concerning the cost of resources (maximum, average), preferred geographical locations of some virtual functions, and defining acceptable delay within a data centre and between data centres or other metrics.

- *NS Allocated Resources Database* (NARD) provides information about resources allocated to each NS and information about NS handling orchestrator.
- *Infrastructure Partitions Database* (IPD) keeps NS-PoIs topology as seen by relevant M-NFVO. It has a similar form to GTD but also consists of ADCs, i.e., data centres integrated, according to the criteria of NSRD.
- *VNF Migration Engine* (VME) is autonomously triggering VM migration between data centres of an ADC to optimize the resource load balancing inside the ADC. M-NFVO can also trigger VNF migration.
- *NS Charging Records* (NSCR) keeps real-time charging records concerning Infrastructure related resource usage and related events.
- *Resource Layer Orchestrated Function* (ROF) is a function that can be orchestrated by one of M-NFVOs on request of the resource layer operator to extend resource layer functionality. ROF, using the resource layer message bus, can interact with other resource layer components. The resource layer operator interacts with OSS/BSS using the *Os-Rl* interface to orchestrate AOFs.
- *Resource Layer Operator Portal* (ROP) is used by the resource layer Operator to manage the resource layer and to interact with Infrastructure (site) Providers. It provides resource layer status information (faults, performance, real-time and historical resource usage/charging records) and some trends from RDAE.
- *Data Centre-Resource Layer Portal* (DRP) is used for the resource layer interactions with data centre Providers. It is involved in adding and removing a data centre. A part of the process is the authentication of the data centre by RSE, and this operation updates all relevant NS-PoIs. The process may include VNF migration before data centre is removed or after it is added.
- *VIM Adapter*, which primary role is interaction with A-VIMs, and, if necessary, it interconnects dynamically multiple sites using M-WIMs. The A-VIM is a single point of interaction with a Site (data centre) and may interact with numerous VIMs of a Site (data centre).

Note that resource layer in not the only new component in the modified architecture. The Network Slice Topology Selection (NSTS) component, part of OSS/BSS, interacts with resource layer to create an appropriate NSRD record which includes the identification of serving M-NFVO. The Aggregated VIM (A-VIM) can be used to interface all M-VIMs of the same provider or site. The Far-Edge integration depends on Far-Edge technology and is provided by the Far-Edge Gateway (FEG).

The procedures of the resource layer include:

- *Adding a Data Centre:* For adding a data centre to resource layer resource pool, two mechanisms have to be considered. The GTD and the NSRD are updated, and eventually, also ADCs. Finally, the VME checks if the load balancing between data centres of modified ADCs has to be done. If the result changes the topology of data centres handling NS, the M-NFVO can trigger VM migration between data centres and ADCs.
- *Removing a Data Centre:* The removal of data centre can be requested by data centre Provider, resource layer Operator or by an automated process of resource layer; for example, a microsite can be switched off for energy consumption reduction. If the data centre is a member of ADC, and the remaining ADC resources are insufficient, the resource layer may interact with OSS/BSS to trigger VNF migration handled by M-NFVO.
- *VNF Migration within an ADC:* The ADCs are seen externally as a single data centre; however, they are a composition of interconnected data centres with no embedded load-balancing mechanism between them. To that end, it is proposed to use a resource layer-controlled load balancing mechanism that moves VMs between different data centres of the same ADC. The M-NFVO can also trigger VNF migration to provide load balancing between separate data centres/ADC to optimise the traffic by the VNFs following the mobile network users' mobility, etc.
- *Creating Resource Partition.* This procedure is the first step in the NS orchestration process and is in details described below.

The infrastructure comprises many data centres of different sizes, located in different location and registered in GTD, see Figure 7-9. For each data centre, regularly updated DCF is provided, and the NSTS sends to resource layer a list of Infrastructure requirements of each NS. The subset of data centres handling NS (NS-

PoI) in the first step is selected according to the delay between data centres and location constraints described in NSRD. In the next step, some data centres that do not fulfil other constraints like energy efficiency, cost, size, or reliability are removed from the subset. The transmission delay is typically about 6 μs/km; so, for 50 km distant data centres, the one-way delay is about 300 μs – a delay negligible for most non-time critical NS (the delay between VMs of the same server [OBT+16] can be about 100 μs). The selected data centres are next checked against lower delay (comparable with intra-data centre delay), and isolated groups of data centres fulfilling the condition are grouped as ADCs that M-NFVO and M-VNFMs see as a single data centre. The information about individual data centres and ADC members (i.e., data centres) is stored in GTD. The delay criterion is, however, not restrictive enough for Edge data centres, and constrained location in such a case is much more stringent. The Edge data centres, located close to the traffic source, can reduce the traffic volume. For example, a video analytics application that handles video streams to find image features installed on the network Edge significantly reduces the traffic. Therefore, some explicitly marked data centres can be excluded from the data centre grouping. The creation of ADC implies a need for the estimation of their geographical location and cost.

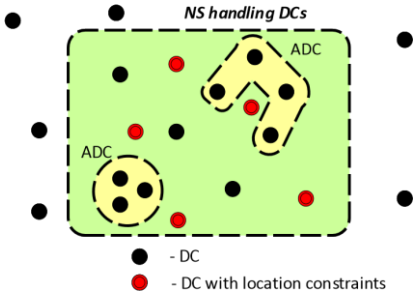


Figure 7-9 Example of topology of data centres and ADCs that will handle an NS. The data centre selection is based on delay and location constraints.

The problem is that ADC can be composed of distant geographical data centres; therefore, this is no longer a single point on a map. To provide ADC coordinates, we propose to use an algorithm that will use the centre of gravity approach based on the data centres resource amount. Another problem is linked to the different costs of resources of data centres of various providers. In such a case, the average price of ADC resources has to be calculated. A similar problem concerns the estimation of ADC reliability and energy efficiency. Different algorithms can be used to handle the issues, but their selection is out of the scope of this document. Data centres that are not a part of NS- PoI can be used in particular situations, for example, in the case of data centres or their link failures.

7.2.3 Preliminary workflows and evaluation

The distributed Cloud Continuum infrastructure has to cope with 5G/6G requirements and be cost-effective. The density of Cloud Continuum nodes , servers or data centres, impacts the elasticity of local deployment of the 6G network functions or end-user applications. The cost of dense Cloud Continuum can be high and may require network deployment for interconnecting Cloud Continuum node. On the other hand, a sparse Cloud Continuum grid may not be able to fulfil the latency requirements of 5G/6G networks or their services. This subsection evaluates the number of Cloud Continuum nodes deployed to ensure low latency of 5G/6G networks in Poland. The analysis is built upon a population map of Poland. The population data includes 956 cities. The country's population is 38 million citizens, and the area is about 313,000 km^2 in a rectangle of 650x600 km. It has been assumed that Cloud Continuum nodes will be located in cities with a preference for large cities. The analysis takes into account the delay requirements of ultra-low latency services. In this exemplary analysis, it has been assumed that end-to-end Control Plane latency should be smaller than 10 ms and User Plane latency should be less than 2 ms. The end-to-end latency includes the radio interface and RAN transport delay, as well as the 5GC/6GC transport, SBA delay and processing time by nodes. There are no maximal values of 5GC/6GC transport delays defined by 3GPP. Depending on the type of traffic, the 3GPP defines different delay values based on 5QI [23.501]. The 5G QoS Identifier (5QI) is a scalar used to reference 5G QoS characteristics. For low-latency applications, the packet delay budget is defined as 10 ms.

In this analysis, only two types of delay related to the core network have been evaluated. The first is related to the virtualisation stack, and the second is related to the propagation delay between the nodes (the queuing

delay is ignored). It has been assumed that the virtualisation delay is about 50 µs [CBC+21] and the distance-related transport delay is 6 µs/km [DFM+21]. It has also been assumed that the virtualisation and transport-related delay for the ultra-low latency service should be lower than 1 ms. This requirement limits the radius of such service to ca. 70 km (shortest path). Due to fibre duct topology linked, e.g., with the road topology, the geographic distance between Cloud Continuum nodes has to be reduced to about 50 km; therefore, each Cloud Continuum node service radius should be about 25 km.

A two-stage algorithm has been proposed to evaluate the number of needed Cloud Continuum locations. In the first stage, it tries to use the minimum subset of cities, with a preference for the largest ones, to provide country-wide coverage. Later on, it reduces very close locations using the DBSCAN algorithm [MYL+23]. The operation can be linked with the aggregated Cloud Continuum node exposure to the orchestrator.

An evaluation of the placement of Cloud Continuum node with a geographic service diameter equal to 50 km (ca. 1 ms delay between Cloud Continuum nodes) is presented in Table 7-2, and Figure 7-10 presents the obtained results, Table 7-3 and Figure 7-11 present results for geographic service diameter 150 km (3 ms delay).

Table 7-2 Cloud Continuum nodes placement with a geographic service diameter equal to 50 km

	Number of available Cloud Continuum node locations	Population of the smallest city	Number of locations selected by the algorithm	Number of locations after close Cloud Continuum nodes aggregation	Number of cities covered by
Scenario 1-25	900	1 818	312	241	953
Scenario 2-25	600	4 792	310	241	910
Scenario 3-25	500	6 700	294	229	878
Scenario 4-25	400	10 241	258	204	802
Scenario 5-25	300	15 471	211	175	697
Scenario 6-25	200	22 246	159	133	559
Scenario 7-25	100	44 980	83	73	356

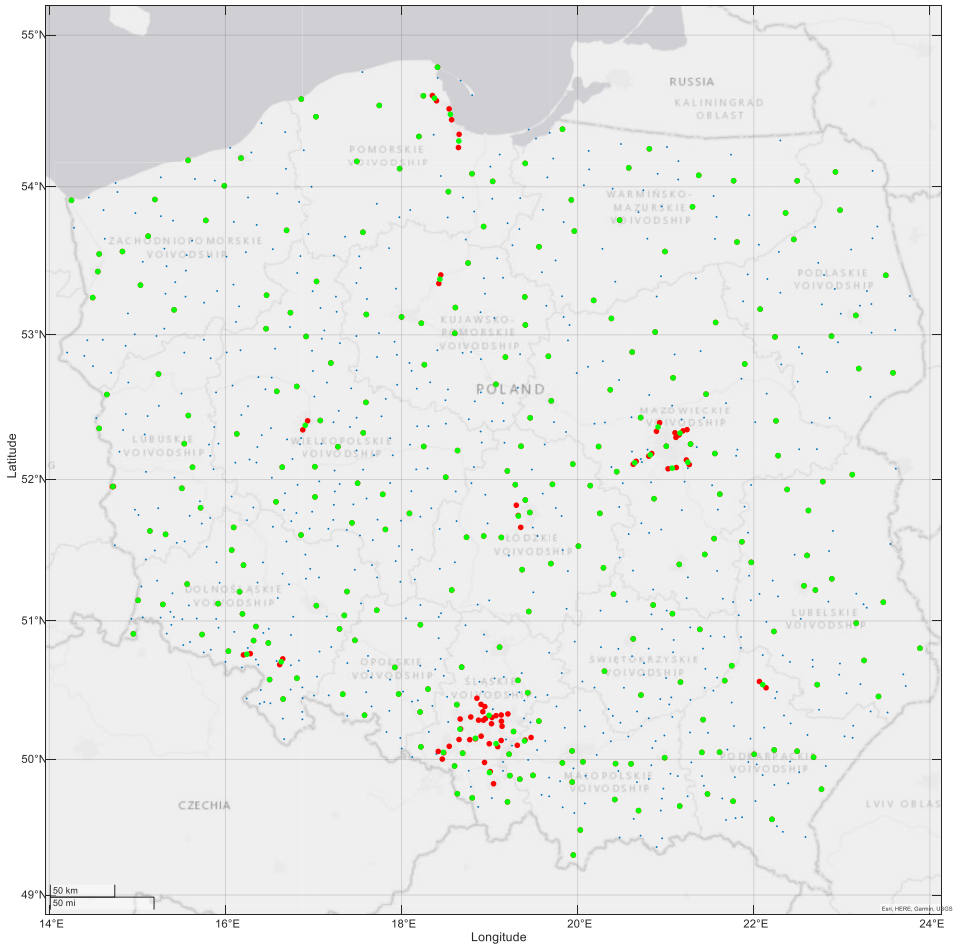


Figure 7-10 Selection of Cloud Continuum node locations for Scenario 2-25 (green points). Small blue points are city locations, and the red points mark the initially selected, close locations that were aggregated by clustering. In this scenario, 46 cities are out of service.

Table 7-3 Cloud Continuum nodes placement with a geographic service diameter equal to 150 km

Scenario	Number of available Cloud Continuum node locations	Population of the smallest city	Number of locations selected by the algorithm	Number of locations after close Cloud Continuum nodes aggregation	Number of cities covered by
Scenario 1-75	900	1 818	49	38	956
Scenario 2-75	600	4 792	50	38	956
Scenario 3-75	500	6 700	50	38	956
Scenario 4-75	400	10 241	52	40	956
Scenario 5-75	300	15 471	50	38	956
Scenario 6-75	200	22 246	52	40	952
Scenario 7-75	100	44 980	51	39	924

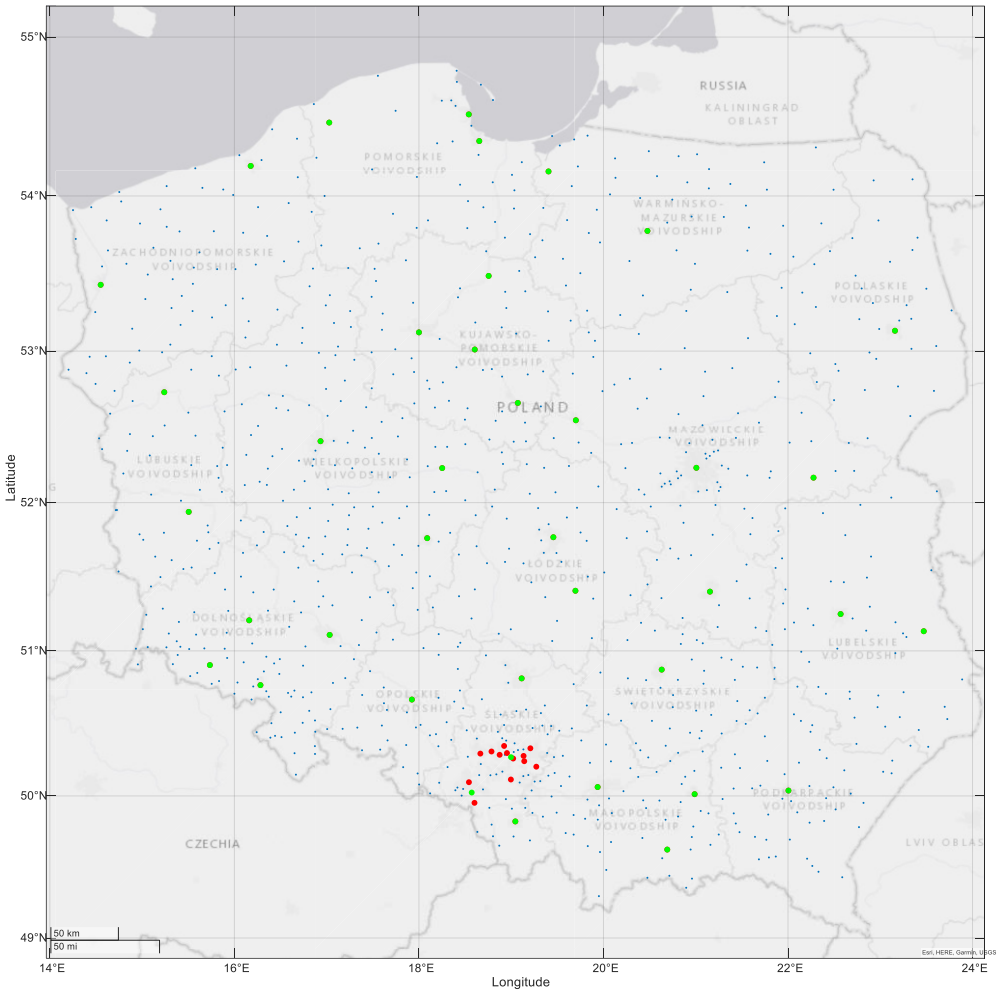


Figure 7-11 Selection of Cloud Continuum node locations for Scenario 5-75. The red points mark the initially selected, close locations that were aggregated by clustering. In this scenario, all cities have access to the service.

Table 7-2 shows that ca. 240 locations of Cloud Continuum nodes deployed even in small cities are needed to provide virtualised ultra-low latency service available 'anywhere'. It is, however, expected that the deployment of ultra-low latency services will be on an island basis instead. Table 7-3 shows that relaxing threefold time constraints reduces the needed locations to about 40, which can be big and medium-sized cities.

The concept discussed in this section assumes that multiple providers will build the Cloud Continuum infrastructure. Therefore, there will be no careful planning of Cloud Continuum nodes' locations, and the operator's goal will be to fill the gaps where they are needed. Although based on evaluation results, some of the factors can be identified as having a low impact on Cloud Continuum nodes distribution and others as having a significant impact. Work on such multi-criterion optimisation that also takes into account RAN node placement will be provided in the D3.4 deliverable of the Hexa-X-II project.

The end-to-end delay requirements analysed in this evaluation study are the following [22.621] and are described in [LCO+23].

7.2.4 Summary

Federation represents a transformative approach in which multiple, independent service providers collaboratively pool their resources to deliver seamless and scalable services. In 6G, federation represents an opportunity for the core network ecosystem as a whole. Federating Network Operators offers a strategic

approach that enables the expansion of services across diverse national domains. Overall, the federated model emerges as a comprehensive solution that not only benefits Network Operators and customers but also fosters a more cost-effective and integrated environment for technology providers. Table 7-4 shows a summary of the main benefits and implications of the Multi-cloud/multi-domain federation enabler in the E2E 6G system blueprint that is under development in the project.

Table 7-4 Benefits and implications of "Multi-cloud/multi-domain federation" enabler

Description	The main aim of the federation model 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.	
Benefits	**KPI improvement**	Latency, network load, reduce complexity for cross-domain deployment, improve QoE
	Design principles [HEX223-D21]	(#1) Support and exposure of 6G services and capabilities (#2) Network Scalability (#7) Internal interfaces are cloud optimized
	Dependencies / Basis for another enabler	Cloud continuum should provide intent-based interfaces for cloud services. 6G Core network should provide intent-based interfaces for network services. Beyond communication functions module should provide the intent-based interfaces for BYC services
Implications	**Requirements**	Intent-based interfaces for every component M&O layer that is able to harmonize all the different parts. Federation interfaces
	Standard relations & regulations	This enabler will impact the orchestration of network functions. Impacts includes but are not limited to [23.501], [23.502].
	Required resources	Cloud resources

7.3 Cloud transformation with quantum technologies

7.3.1 Introduction

Quantum computing (QC) represents a ground-breaking computational approach that leverages the principles of quantum mechanics to perform specific calculations much more efficiently than traditional computers, such as laptops and desktops used in our daily lives [VCL24]. Unlike classical computers, which process information using bits in states of 0 or 1, quantum computers utilize quantum bits, or qubits, capable of existing in multiple states simultaneously due to the quantum property known as superposition. This empowers quantum computers with the capability to concurrently explore numerous potential solutions to a given problem. Quantum parallelism, a crucial aspect of QC, enables the simultaneous exploration of extensive solution spaces, offering the potential for more efficient resolution of complex problems compared to classical computers. Additionally, the entanglement phenomenon establishes correlations between the states of multiple qubits, allowing them to be interconnected in a way that modifications to one qubit instantaneously influence others, regardless of spatial separation. This property is harnessed in quantum algorithms to execute complex calculations with heightened efficiency.

The convergence of 6G and QC not only represents a significant advancement in technological evolution but also heralds the emergence of transformative applications poised to revolutionize various sectors, including telecommunications, healthcare, finance, and AI. The synergy created by combining the ultra-fast, low-latency

communication capabilities of 6G with the unparalleled processing abilities of quantum computing is set to redefine the possibilities and applications of both technologies.

7.3.2 Architectural modifications

The realization of a softwarized continuum, in which network operations are realized as microservices or intelligent agents, will require unprecedented resources for in-network computing, which will be a pillar of communication networks. In parallel, the employment of AI in the future 6G architecture will also create an explosion for computing at the control plane in order to target multiple objectives. First, it could be used for management and orchestration, traffic classification, and time-series forecasting. In particular, referred to management and orchestration, some examples of use cases could be traffic forecasts, automated Virtual Network Function (VNF) placement and network slicing, network self-healing, and hidden patterns discovery, to proactively assist network planning and sizing.

With this immense computing load associated with softwarization and AI, QC may become a suitable candidate for core processing components within O-RAN, particularly for computations reliant on the RIC [LBS+22]. A QC-enabled core RIC holds the potential to deliver real-time outcomes for Near-Real-Time (Near-RT) services. Further, O-RAN can integrate quantum communication to bolster its security measures [AAU+23]. Critical components such as DUs and CUs within O-RAN infrastructure need to be both easily deployable and securely connected. Failure to ensure this may expose the network to eavesdropping, man-in-the-middle attacks, and various other security threats that could compromise user traffic flowing through O-RAN nodes. QKD offers a solution by establishing secure keys between O-RAN components, effectively mitigating these risks.

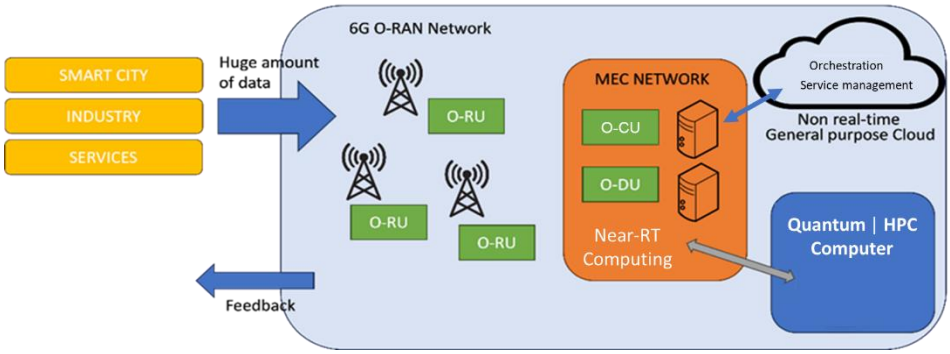


Figure 7-12 O-RAN quantum edge architecture

Quantum-enabled Open RAN networks can also support innovative applications such as quantum optimal algorithms, quantum machine learning, and quantum-assisted blockchain technology. These applications can unlock new possibilities for network services and functionalities.

Leveraging quantum computing demands the development of unique models tailored to its operations. In certain cases, standard computations like radio resource management and allocation may need to conform to QC models or algorithms for problem-solving. Specifically related to the RAN and the edge of the network, open software has obtained significant attention in order to realize customized solutions for specific businesses and use cases (for example industrial IoT in campus networks with focus on low latency). Furthermore, integrating QC hardware into the network architecture requires careful consideration of the limited capabilities of today's quantum hardware and the complexity of multiple computing layers. As compute and storage requirements increase, the efficiency of QC integration decreases in comparison to classical HPCs. Therefore, a phased approach is critical, with quantum capabilities added incrementally, starting at the edge (the computing resources closest to end-user devices with limited capabilities of state-of-the-art noisy quantum computers) and eventually extending to the cloud (centralised cloud data centres providing massive computing and storage resources).

In the above context of necessary unprecedented in-network computing resources, research and standardization are ongoing to create an open source High-Performance Computing (HPC) stack [SRK+22] including open Hardware and Software. This can play a pivotal role in future 6G open source MEC. This open-source approach

can help in ensuring Europe's global competitiveness and data sovereignty, since it creates the capability to handle the entire data life cycle, which in turn relies on the underlying in-network edge computing infrastructure.

Here, we aim to investigate the integration of quantum communication and computing technologies into O-RAN and open source HPC stack for MEC (see Figure 7-12). This to improve the performance (in terms of, for example, latency, energy usage, connectivity, resilience) of the classical 5G and upcoming 6G RAN-MEC. Specifically, the idea is to integrate current available quantum computing platforms (with limited number of qubits), that are available today, with MEC classical platforms. Centralized and distributed quantum classical MEC can be considered.

Furthermore, we target designing and developing open-source solutions for hybrid quantum-classical computing that have to support RAN and MEC network functionalities/microservices/agents. This is an important pillar in future 6G automated industrial networks, in which thousands of devices (robots, sensors, etc.) will significantly stress the computing at the RAN-edge of the network. Evaluations will consist of emulation of computing environments, exploiting the benefits of an extended quantum-classical RAN-MEC open source stack (socket to application) together with some specific experimental evaluations using available computing (HPC and quantum) platforms in the consortium.

The hybrid classical model is expected to exploit the unique properties of quantum mechanics. In the midterm (upcoming 3-5 years), quantum capabilities may be used for secure information distribution, while heavy computing tasks remain within the domain of classical computers due to their current superiority over quantum counterparts. We will assess chosen Open RAN functionalities, comparing power consumption and latency in both centralized and distributed hybrid quantum-classical architectures. We will assess chosen Open RAN functionalities, comparing power consumption and latency in both centralized and distributed hybrid quantum-classical architectures.

The hybrid classical model is expected to exploit the unique properties of quantum mechanics. In the midterm (upcoming 3-5 years), quantum capabilities may be used for secure information distribution, while heavy computing tasks remain within the domain of classical computers due to their current superiority over quantum counterparts. We will assess chosen Open RAN functionalities, comparing power consumption and latency in both centralized and distributed hybrid quantum-classical architectures. We will assess chosen Open RAN functionalities, comparing power consumption and latency in both centralized and distributed hybrid quantum-classical architectures.

7.3.3 Summary

The impact of quantum technology on cloud transformation in 6G networks depends on the maturity of quantum technologies and their seamless integration into the network architecture. The incorporation of quantum technologies will involve their partial integration as sub-modules or in a hybrid-classical combination. With an emphasis on KPIs such as latency, security and robustness, quantum technologies will only be strategically integrated where their inclusion offers advantages over purely classical scenarios. The imperative for a hybrid architecture is to seamlessly integrate quantum capabilities into existing network infrastructures, particularly at the edge of the cloud.

The hybrid classical model is expected to exploit the unique properties of quantum mechanics. In the midterm, quantum capabilities will be used for secure information distribution, while heavy computing tasks will remain within the domain of classical computers due to their current superiority over quantum counterparts. We will assess chosen Open RAN functionalities, comparing power consumption and latency in both centralized and distributed hybrid quantum-classical architectures. We will assess chosen Open RAN functionalities, comparing power consumption and latency in both centralized and distributed hybrid quantum-classical architectures. Table 7-5 shows a summary of the main benefits and implications of the Cloud transformation with quantum technologies enabler in the E2E 6G system blueprint that is under development in the project.

Table 7-5:Benefits and implications of "Cloud transformation with quantum technologies" enabler

Description	Cloud Transformation with quantum technologies

Benefits	**KPI improvement**	Reduce latency, increase robustness of the network Hybrid (classical-quantum) open source
	Design principles [HEX223-D21]	Integration of small-scale quantum computing platforms Hybrid quantum-classical centralized / distributed computing
	Dependencies / Basis for another enabler	Interfaces for the entanglement distribution Modular architecture
Implications	**Requirements**	Quantum hardware (computers, routers, repeaters,..) Interfaces for hybrid integration of classical and quantum components Quantum open sources
	Standard relations & regulations	3GPP specification [TS 38.300]
	Required resources	Quantum hardware (s.a.)

8 Proof of Concepts

In this section, we present the component proof of concepts (Component-PoCs) that are studied within WP3.

8.1 Component-PoC #B.2: Distributed Model Training and Inference

8.1.1 Remote controlled robot use case

The PoC dataset is collected from an internal network simulator. The simulation consists of a remote-controlled robot, with a camera attached to it, in a mining use case. An actuator robot with a UE attached to it is located at a distant location and is controlled by a remote controller that has another UE. The actuator robot streams out video from the environment over its UE and the communication link to the controller UE located at the other end. The UE receives the video packets, renders them, and based on the observed video sequences; it sends control signals back to the remote actuator robot to operate the robot. Both UEs are connected to a base station, where the base station transmission power can be configured.

The data logs related to data input Physical, MAC, IP, and video application (data output labels) are recoded during a simulation of video streaming. The input attributes are as follows. Physical entity had attributes related to SINR, received power, received information size; MAC entity had uplink transmission data volume; IP entity had uplink and downlink throughput as input features. The number of input attributes were not too many. To stress the computation and communication of neural network training with higher input feature dimensionality, additional dummy input features (all were set to zero) were added as input to the neural network. Therefore, at the end, Physical, MAC and IP entities had 259, 258, and 257 input attributes in total. The two use cases are deployed at two entities, as two NN models, in the output node. The use case entities in the output node collected dataset at the application level related to the video quality indicators:

- *videobitrate*: measurement of the video bitrate of the received video frames at the control terminal client,
- *videodelay*: measurement of the inter-frame time of the decoded video frames received at the control terminal.

The goal is to estimate *videobitrate* and *videodelay* with the decentralized input attributes in the input node.

In the inference phase, these two NN models are transferred from output consumer to UPF such that UPF can assess the QoE.

8.1.2 Distributed AI-Enabled Technology: Split Learning

This document on the proof of concept (PoC) presents an AI-enabled technology called split learning and showcases benefits and trade-offs on different scenarios. The implementation is performed as a multi-headed, multi-tailed parallel split learning in Kubernetes. Neural networks that belong to different entities are placed in distributed manner to multiple Kubernetes pods.

The PoC consists of three main types of nodes: input node, generalization node, and output node. The global NN model is split into these node types, and every node type contains portion of the global NN model. A node type in this demo is defined as the group of at least one neural network model. The neural network models that are deployed at the input and output of the global neural network are called input and output nodes, respectively. The input node contains the input dataset, i.e., ML features; and the output node contains the target labels. The intermediate NN model is deployed in the generalization node. These nodes are illustrated in Figure 8-1.

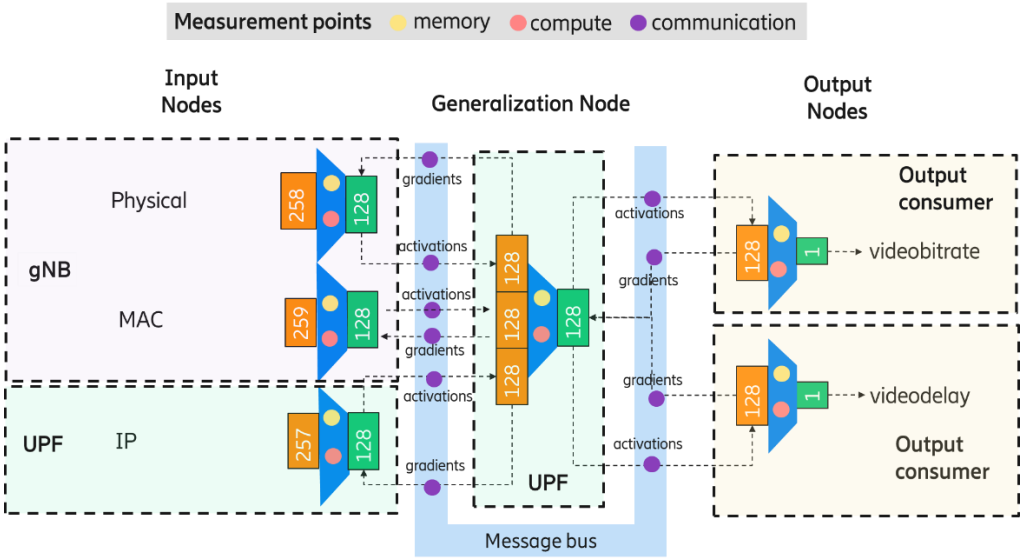


Figure 8-1 Illustration of cross-network function training and model generalization via joint optimization for multiple use cases in a split learning setting. Blue trapezoids represent Kubernetes pods, and each pod hosts an NN model.

Input node consists of three entities that each contains input data and compute capability. The three input entities are Physical, MAC, and IP, and each entity has one local NN model that trains on the local input data. The generalization node does not have access to raw dataset; it is rather an intermediate entity with a NN model (serving as a compute element) with the goal of taking high compute tasks related to the training of a split learning model. It also serves with the goal of generalizing the intermediate learned representations from input nodes, so called encodings or activations, to multiple use cases deployed in the output node. The output node consists of two NN models serving for two use case tasks, i.e., video bitrate estimation and video delay estimation. The Physical and MAC entities can be co-located in gNB; the IP entity and the generalization node can be co-located in UPF; and the output consumer entities that contains video bitrate and video delay data can be co-located in output consumer such as external application or some other external entity.

The PoC demonstrates three AI enablers as follows.

8.1.2.1 Cross-network function training

This enables model fusion with models that are trained on datasets obtained from different network functions.

8.1.2.2 Model generalization

Model generalization is studied in two scenarios:

Scenario 1 This enables part of the neural network ML model to generalize to multiple use cases (depicted as output consumers), i.e., reusing substantial portion of the neural network for *video bitrate* and *video delay*

estimation. This then is expected to yield less models to manage. The split learning model architecture of scenario 1 is illustrated in Figure 8-2. The scenario consists of training jointly two use cases with the datasets obtained from multiple different network functions such as gNB, UPF, and other consumers that are potentially AFs or may be something else, e.g., PCF. The input ML features are obtained from the gNB and UPF in the input node; and the labels are obtained at the consumers at the output node. The datasets obtained at the gNB, UPF and other consumers (e.g., applications) are not shared in between, instead they train local NN models jointly in a split learning setting. The split learning allows these distributed and split NN models to be trained collaboratively by means of only exchanging activations and gradients.

Scenario 2 Moreover, the modularization of NN model enables *data domain adaptation* to reuse a model between data domains or to jointly train high performing models when there are distributional differences between domains. This also allows us to train a model for a domain with missing labels when there exists a different domain for the same task with existing labels. There may be different data domains potentially caused by factors including base station reconfiguration, upgrade in the system, or to transfer models between heterogeneous networks, etc. This approach enables adaptation of an ML model in a data drift scenario to sustain model efficacy. This scenario is a collaborative NN model training consisting of only one use case, *video bitrate* estimation. In this scenario, the datasets are obtained via network simulation where there was a data drift during a video streaming session. This typically occurs since at the beginning of a video streaming, there is a video initialization phase where video packets are downloaded into the video playout buffer. The video client increases the playout bitrate until it reaches an optimum point, from then on the streaming is in steady-state. In *data domain 1*, the measurement dataset is obtained from the initial phase of the video streaming, i.e., between the duration when the video streaming is initiated and the time when it reached a steady-state. In data domain 2, the measurement datasets are obtained after a steady state has been reached. The data distributions between data domains 1 and 2 are different. Conventionally, an ML model that is trained in a dataset with a certain distribution under-performs for another dataset of different distribution. Therefore, conventionally additional model fine-tuning needs to be performed at the target domain. Instead, the goal with this scenario is to learn jointly a good representation of the two data domains via the generalization node with the assistance of additional domain classifier NN model. This way, distributed ML model training for multiple data domains can be performed once, and one model can serve for two data domains, simultaneously. The split learning model architecture of scenario 2 is illustrated in Figure 8-2. Data domain ids (1-data collected at video initialization phase; or 2- data collected at steady state video streaming) are the target labels that the domain classifier trains on. In training, it learns to differentiate these two different data distributions, by reversing the sign of the gradients in the backward propagation after the domain classifier node. This entity needs to obtain these target variables in training phase.

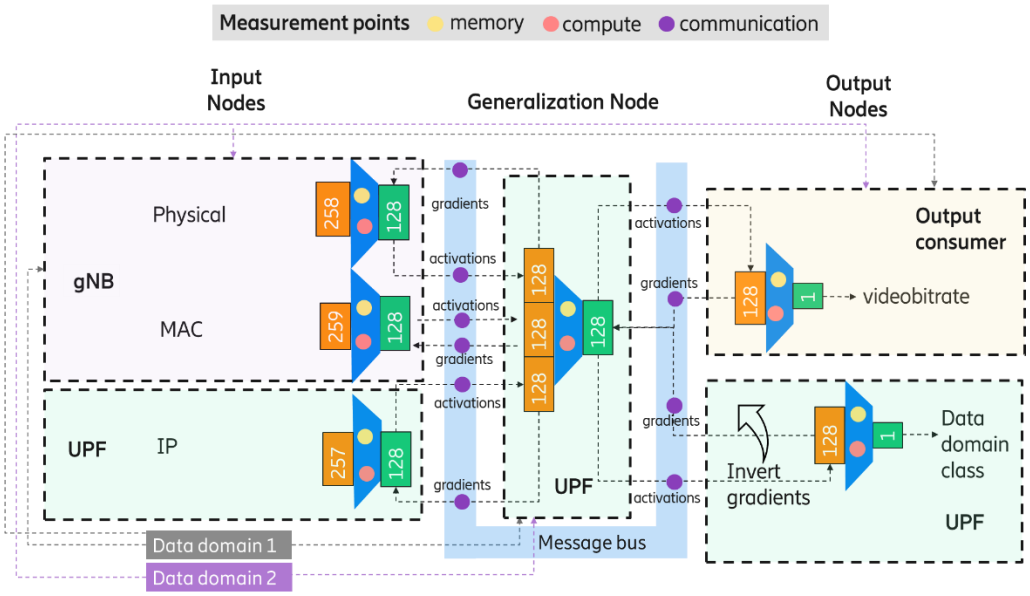


Figure 8-2 Illustration of model generalization via domain adaptation in a split learning setting. Blue trapezoids represent Kubernetes pods, and each pod hosts an NN model.

8.1.2.3 NN model layer offloading

This enables offloading a portion of NN model layers between the nodes, e.g., from output node to the generalization node. For instance, when a consumer output node (e.g., a battery depended on computation device with limited compute) cannot train the local NN model, then it can offload some of the NN model layers to the generalization node, e.g., a UPF in core network, as illustrated in Figure 8-3 and Figure 8-4.

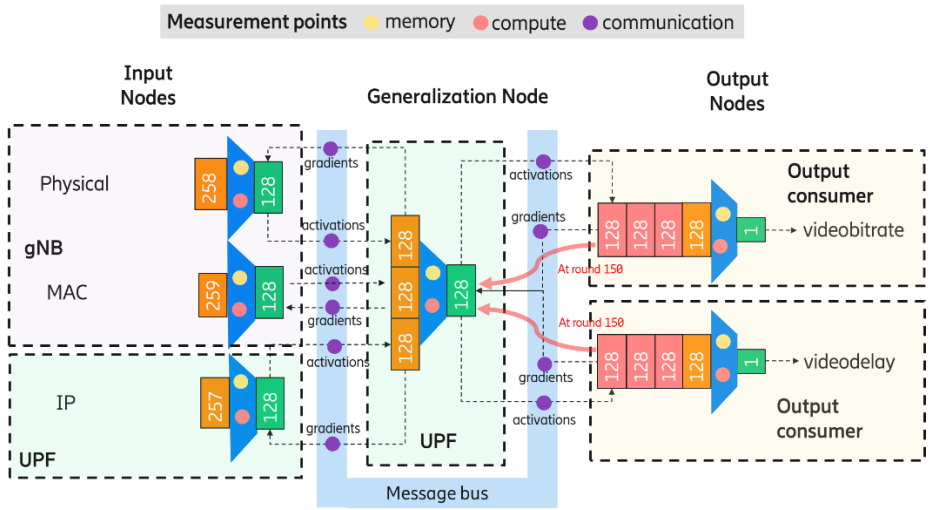


Figure 8-3 Illustration of three NN model layers being offloaded from output nodes to the generalization node. Blue trapezoids represent Kubernetes pods, and each pod hosts an NN model.

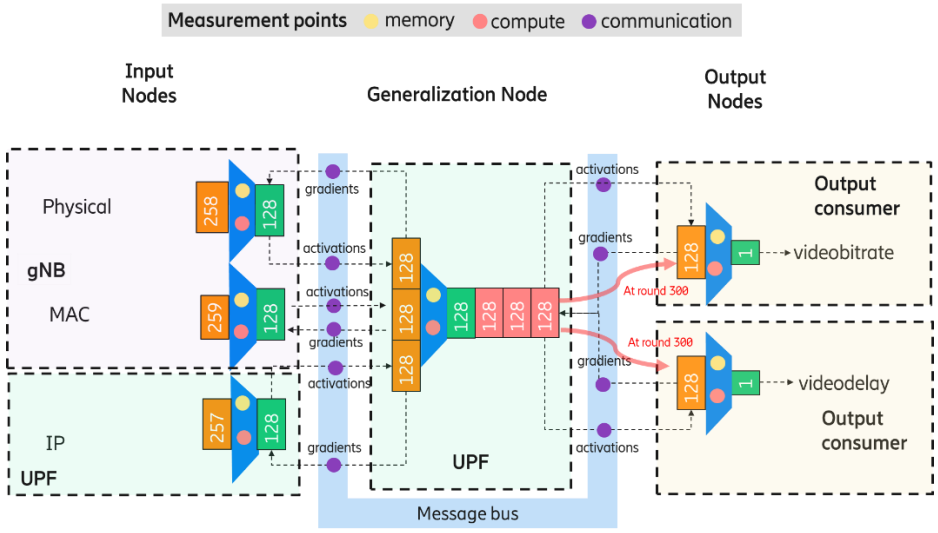


Figure 8-4 Illustration of three NN model layers being offloaded from generalization node to the output nodes. Blue trapezoids represent Kubernetes pods, and each pod hosts an NN model.

8.1.3 The input node

The last output layer of each entity in the input node is connected to the input layer of the generalization node.

For every training data batch (of size 256 samples):

1. Pre-condition: The entities at the input node and the output node aggregate local datasets on shared key, e.g., on pre-determined time interval or other identifier, and generate global sample ids. These entities align on these sample ids in training and inference. Every batch executed in all NN models in a training round consists of the same sample ids, i.e., datasets samples from the same time interval.
2. The Physical, MAC, and IP nodes select a batch (with batch index 0, and batch size of 256), go to step 4,
3. The Physical, MAC, and IP nodes receive the common batch index generated and broadcasted by the generalization node,
4. The physical, MAC, and IP nodes execute forward-propagation on the dataset at the given batch index and predefined batch size (e.g., 256),
5. The input nodes provide activations (encoded representation at the output of local NN models) to the generalization node simultaneously,
6. The input node receives specialized gradients from the generalization node (each input node receives different gradients),

7. The input node performs backward propagation on the received gradients,
8. The input node updates the local NN model weights,
9. Repeat steps 3-8 until training ends.
   Training is terminated when a pre-determined number of rounds has been reached. The termination message is then sent to the generalization node, and then from the generalization node to the output node. This way, all nodes terminate and release the compute resources.

There are three input entities in the demo for each input data source, and they are Physical, MAC and IP. The three input modules are considered as cross-network functions which establish the reason for cross-network function training. The input nodes are referred to also as *head* nodes.

8.1.4 The generalization node

The first layer of the generalization node is connected to three input entities in the input node, and the last layer of the generalization node is connected to the two output entities (use case entities) in the output node. The input layer of the generalization node is 384 neurons, i.e., the first 128 neurons are connected to the output of the first input entity (Physical), the second 128 neurons are connected to the output of the second input entity (MAC), and the third 128 neurons are connected to the output of the third input entity (IP).

The generalization node repeats execution of steps 1-9 until training ends.

1. The generalization node receives activations from different input entities (from Physical, from MAC, and from IP entity) in parallel,
2. The generalization node concatenates all three activation matrices into one matrix,
3. The generalization node performs forward pass on the concatenated matrix,
4. The generalization node sends the activations (output from its final layer) to the output entities. All output entities receive the same activations,
5. The generalization node receives gradients from the output entities,
6. The generalization node averages the gradients that are received from multiple output entities,
7. The generalization node performs backward pass on the aggregated gradients,
8. The generalization node sends the gradients at its first input layer to the input node, i.e., the output of 128 gradient values for each input entity. Note that the gradients that are sent to the input entities are not identical.
9. The generalization node acts as a coordinator in between the input node (head) and output node (tail); therefore, it generates a random batch index at every round of training, and then sends the same batch index to all input entities along with the gradients. The input entities execute on this batch index in the next training round.

8.1.5 The output node

The first layers of the two entities in the output node are connected to the input of the generalization node. The output node is also referred to as *tail* node.

1. The output entities perform forward pass in parallel on the received activations from the generalization node,
2. Compute the loss values depending on the defined loss function. Loss functions defined in this proof of concept are *cross-entropy loss* between the estimated class predictions for
   a. video bitrate classifier {high or low bitrate}
   b. video delay classifier {high or low delay}
   c. data domain classifier {data domain 1 or data domain 2},

   Here, what is meant by data domain is data distribution, and data domains 1 and 2 have two significantly different data distributions.
3. The output entities perform backward propagation based on the computed loss value,
4. The output entities in parallel sends the gradients to the generalization node.
   - Special condition: if the output entity is a domain classifier (as in Scenario 2), the sign of the gradients is altered before being delivered to the generalization node,
   - Special condition: in scenario 2, the output node for video bitrate estimation serves to estimate the video bitrate for two data domains. Therefore, there are two forward pass

operations performed when the activations are received, and then the loss values are calculated for 2 domains, separately. The two loss values for the two domains are then averaged. And the gradients are obtained via the backward propagation given the averaged loss value. Therefore, in this case, there are two forward pass and one backward pass operation at the output entities.

5. Repeat steps 1-4 until training ends.

The output node can serve multiple purposes, and in the scope of the PoC, an output node was one or more of:

- a video bitrate estimator output entity,
- a video delay estimator output entity
- domain estimator output entity.

8.1.6 Kubernetes pods and settings

There are 6 pods implemented and executed in parallel: 3 for the input node; 1 for the generalization node; 2 pods for the output node. In the first scenario, there are 2 pods for the two use case entities at the output node (one for the video bitrate estimation and one for the video delay estimation); and in the second scenario, there is one pod at the output node for the video bitrate estimation use case entity for two data domains; and 1 other pod at the output node for the domain classifier entity.

The CPU and memory are configured in design time via *yaml* configuration file. We experiment with different CPU and memory settings. We allocated 2 CPU's and 1Gi memory on each Kubernetes pod at the input and output nodes; 2 CPU's and 2Gi memory at the generalization node.

8.1.7 Measurement Points and Evaluation Method

Memory:

The size of the NN models in each entity is recorded at every round of training. This is especially important as it provides input to the orchestration and coordination units. Moreover, we record the memory footprint via accessing the value at */sys/fs/cgroup/memory.current* every 5s.

Compute:

Forward- and backward-pass time are recorded at every round of training at every entity. In addition, CPU statistics are obtained every 5s by accessing information at /sys/fs/cgroup/cpu*

Communication:

The communication cost is obtained via calculating the message size of the transmitted data in every delivery of information from any node to any other node.

Model Efficacy:

Accuracy is calculated as the ratio of correct predictions to all predictions; and the f1-score is calculated as (2 * *precision* * *recall*)/(*precision* + *recall*), where precision quantifies the ratio of correctly predicted high values to the sum of all values predicted as high; while the recall quantifies the ratio of correctly predicted high values to sum of all actual high values.

8.1.8 Results

With *cross-network function training*, we demonstrate collaborative training without moving and sharing data in between the entities. This, up-to-some extent, assisted preserving privacy and enabled data protection. Moreover, as the NN models at the input nodes are also encoders, they assisted in data size compression by reducing the dimensionality of the input attributes. This reduced the communication overhead in parallel. In addition, cross-network function training enables an ML model to train with richer input ML attributes obtained from different layers in the network stack such as RAN, MAC, and IP. This way, the model efficacy of the ML model can be improved. In this study, we observed an increase of ML model efficacy from 0.44 to 0.68 (+54%).

There are also clear benefits of *model generalization* with respect to reduction in computation, memory, and storage overhead. With model generalization, the reduction in cost and memory becomes proportional with the number of use cases. In this work, two use cases were trained simultaneously via joint optimization, hence the reduction of compute, communication, and memory were 50% (1/2), while the model efficacy was sustained. However, this highly depends on the similarity of the tasks and use cases. Communication overhead is also indirectly reduced due to a smaller number of models (thus less training and communication rounds) as the models for serving multiple tasks are trained simultaneously via a common generalization node. Moreover, domain classifier reduced the training requirements when training ML models for different data distributions of the same task. An unsupervised domain adaptation-based approach yielded 11% increase in model efficacy as compared to direct model transfer from a source data domain to a target data domain. A supervised domain adaptation approach yielded faster model convergence and achieved faster model generalization.

By *NN model layer offloading*, significant reduction (16%) in the CPU pressure at the output nodes during model training is quantified. But this yielded 9.7% increase in the CPU pressure at the generalization node where the model layers were offloaded to. The forward- and backward propagation time are also highly influenced by the number of NN layers, and this was quantified by significant change in forward- and backward propagation times obtained at the nodes before and after offloading. Moreover, 99% reduction in memory footprint was observed after NN model layers were offloaded. Offloading from output node to the generalization node also reduced the total training and inference time (sum of FP time at the generalization and output node). The total forward pass time was reduced from 0.04s to 0.01s (75%) and 0.033s to 0.018s (45%) in *videobitrate* and *videodelay* estimation entities, respectively. This reduction has been caused by higher memory allocation at the generalization node in design time as well as its already relatively larger model size before offload, which did not result in significant impact after offload. The reduction in CPU utilization and memory at the output consumer entities relaxes the compute and energy requirements as these output consumers may be deployed at end devices with limited battery-power, memory, and storage. During the offloading, no significant impact on the model efficacy and the training performance were observed. However, the impact on the model efficacy highly depends on the offload strategy and use case, therefore, should not be considered as a conclusive statement. The role of offloading here becomes significant when there are not available compute resources for additional training tasks at edge nodes. One example could be offloading of model layers from output consumer smartphones that are running application clients to the core network located at cloud with large compute capabilities. Another example could be offloading model layers from base stations to the core network.

8.2 Component-PoC #B.3: Trustworthy flexible topologies in 6G, leveraging on “beyond communication” aspects

8.2.1 Inventory management use-case

Flexible network topologies enable versatile, robust and dynamic applications not only intended towards public use but also for industry needs. In the context of this PoC, the design, functional and technical requirements are formulated under a warehouse inventory audit scenario. Inventory audit is an ideal area of application due to the diverse nature of network coverage caused by high ceilings, signal bouncing off metallic surfaces and even mobile metallic structures. The impact is shown through regions in the infrastructure that have either no network coverage or unstable connectivity. These areas significantly benefit from adaptable infrastructures capable of providing network and computing resources precisely where and when they are required. Productivity downtime can also be minimized due to uncertain circumstances causing network loss, assuming the infrastructure is operational in a “local” connectivity (i.e., no-internet connection) state.

The inventory audit scenario utilizes a cluster of three Worker Nodes (WN), namely two Autonomous Mobile Robots (AMR) and one autonomous Unmanned Aerial Vehicle (UAV), implemented in a 2x3x2.5m (WxLxH) operating area. The area consists of several zones emulating pallet racks holding different types of packages that the worker nodes must inspect, identify and localize in all three spatial dimensions. It is important to note that the scenario operates completely autonomously, orchestrated from an edge server installed in the

warehouse premises for real-time remote monitoring, task allocation and more. Therefore, any network downtime or poor performance will cause failsafe procedures to take place, disrupting the workflow.

8.2.1.1 Worker nodes

All worker nodes are equipped with cameras, localization sensors and a robotics software stack (ROS2) for autonomous capabilities. The UAV relies on the open-sourced PX4 autopilot for its core autonomous capabilities, ROS2 is supplementary. They also feature two-way communications with the edge server using DDS and MQTT protocols, through the local network, with multi-connectivity capabilities. More specifically, the robot-devices for this use-case are the following:

- **AMRs:** 2 units
- **Autonomous UAV:** 1 unit

Function

The AMRs and UAV work in a collaborative manner in order to inspect the warehouse racks in a bottom-up approach, identifying the packages on the shelves using a custom-trained computer vision algorithm. The AMRs have a custom backplate that acts as a docking station for the UAV, in order to carry it around the warehouse for energy efficiency and safety reasons. When the WNs are tasked to inspect a zone, the AMRs are responsible for inspecting the ground-level shelves, while the UAV for the higher-level shelves that are less accessible.

8.2.2 Flexible topology use-case

For the demonstration of the flexible topology PoC, an additional Flexible Topology Node (FTN) is utilized, as shown in Figure 8-5. This takes the form of an autonomous UAV and similar to the worker node UAV, it also features ROS2 and PX4 autonomy software stacks with multi-connectivity capabilities. It will communicate with the edge server using a 5G network interface routed via a VPN in order to receive the latest information on the WNs location and task status. This assumes that the infrastructure supports a reliable public 5G connection.

Scenario

In our extension of the inventory audit scenario with the additional flexible topology functionality, we assume two potential triggers for the dynamic network allocation utilizing the FTN:

1. Local connectivity loss: The local network becomes suddenly “undiscoverable” due to unforeseen reasons.
2. Out-of-coverage intent: An inspection task is received that is knowingly out-of-bounds to existing local coverage.

In both cases, the WNs enter a “paused” state that stops any current operations in a safe manner. For example, if the UAV was mid-flight, it triggers an autonomous behaviour to dock to the closest AMR. It is also noteworthy that in this state, each WN must rely on its onboard autonomous capabilities in order to achieve that state. Additionally, all affected WNs enter a “network discovery” state that triggers their network interfaces to search for a network with the stored FTN’s credentials. Normal operation is allowed only once a reliable network connection is established.

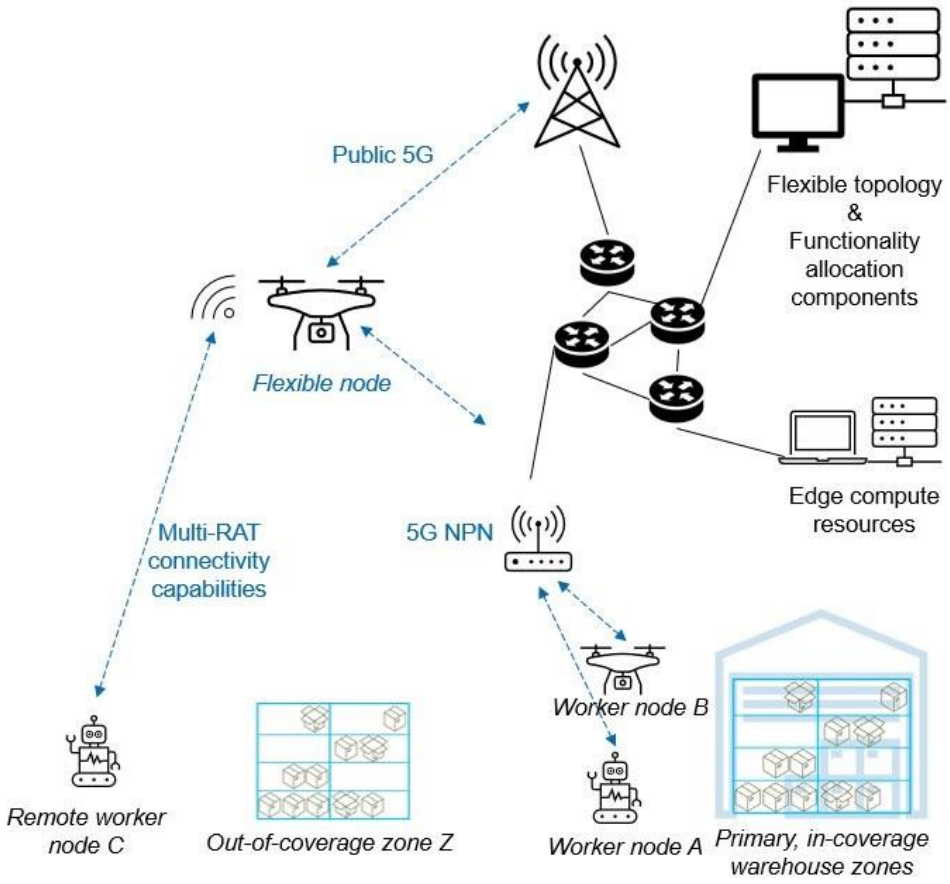


Figure 8-5 Flexible topology architecture

It is then up to the flexible topology component to evaluate the topology and allocate the UAV to the desired location. The flexible topology is pivotal to the PoC B.3 in demonstrating the adaptability and resilience of the 6G network in the warehouse scenario. The FTN plays a crucial role in this setup, acting as a dynamic link within the network topology. Operating autonomously, the FTN ensures continuous connectivity and optimal network performance, even in challenging environments characterized by obstacles and variable signal strengths. The end-result of this topology allocation is the restoration of stable internet connectivity, allowing previously isolated devices to communicate with the edge server effectively.

8.2.2.1 Topology Formulation in Warehouse Environment

Upon discovery, the topology formulation process begins. Here, the flexible network utilizes AI/ML techniques to dynamically create a flexible and resilient network topology. This formulation considers factors like node trustworthiness, energy consumption, capacity in terms of throughput, and the specific demands of the warehouse operations. The topology is designed to ensure that FTN nodes with higher trust scores, based on reliability and performance metrics, are prioritized for crucial data transfers. More specifically, for this scenario, the unique FTN finds the best strategic position each time to provide connectivity to the nodes.
In this PoC, the FTN's ability to adapt its position and network parameters on-the-fly is critical. For instance, if a worker node moves to a previously uncharted area of the warehouse, the FTN recalibrates its position to ensure uninterrupted connectivity. This dynamic response is key in maintaining a high level of operational efficiency and minimizing the impact of potential network disruptions.

8.2.2.2 Coordination and Trust Evaluation Among Nodes

Coordination amongst the Worker Nodes and the FTN is pivotal for the seamless execution of tasks in the warehouse. Each node operates isolated and the communication is facilitated through the FTN. The FTN, as it is equipped with dual network interfaces (WiFi and 5G), serves as a pivotal link in the topology, providing

coverage extension and computational support to nodes that may be in connectivity blind spots through Wi-Fi whereas the communication with the rest of the network is realized through 5G.

Trust evaluation is an ongoing process, where each node's performance and behaviour are assessed based on historical data, response times, data integrity, and security credentials. The dynamic nature of the warehouse environment, with its variable demands and its potential for connectivity issues, makes this continuous trust evaluation critical. Nodes with higher trust scores are preferred for task allocation, especially in crucial operations, ensuring a secure and reliable network environment. For instance, in an event where an AMR experiences a loss of connectivity due to its position in the warehouse, the FTN, through its dynamic topology, can quickly reposition itself to provide the necessary network and computational support. This adaptability not only minimizes downtime but also ensures that the warehouse operations continue seamlessly, with tasks being reallocated as needed based on the current network topology and node status.

9 Summary and Conclusions

The 6G architecture should be a platform for mobile communication as well as new beyond communication services such as sensing, compute offloading and AI. Further on, the 6G architecture must be flexible enough to support different types of network deployments, such as D-MIMO networks, local mesh ad-hoc networks, and satellite support. The architecture should also support the ability to scale networks based on current needs, to improve the efficiency. However, for this to happen, the architecture should be cloud-native, modular and easily extendable. The objective of this deliverable is to analyse the enablers that support such a 6G architecture.

The AI enablers for the 6G **data-driven architecture** comprise architectural means and protocols, MLOps, DataOps, AIaaS, and intent-based management. The AI enablers form a robust framework for seamlessly integrating AI into the compute continuum of 6G networks. MLOps focuses on making ML models operational by ensuring smooth deployment, versioning, and monitoring within the overall architecture. The key requirements identified for MLOps in this deliverable are i) access to high-quality data (i.e., DataOps), ii) scalable data storage, iii) efficient communication between data and model computation nodes, iv) computation for data processing, model training as well as inference and finally v) security and trust. DataOps enables efficient data collection and integration, serving MLOps with data at the right time and with right quality. The key requirements for DataOps are a robust data quality management functionality, an E2E data pipeline that can serve MLOps with data, and a version control for the collected data. AIaaS framework builds on MLOps, DataOps and intent-based management to provide AI services to different parts of the network and to end-users. The AIaaS framework requires new APIs for exposure in-network and to end-users. It is also identified that AIaaS requires security measures and compliance with regulations and feedback loops for continuous improvement and resource optimization.

To enable flexibility without increasing complexity, 6G needs an easily deployable architecture of modules (e.g., network functions) that can scale to current needs. The **Network modularisation** enablers focus on different granularities of a module, and the evolution of the modules based on different deployments and use cases with the purpose to achieve a network with more flexibility and higher efficiency and thereby be able to improve e.g., latency, throughput and sustainability. The enabler also investigates the streamlining of interactions between different modules/domains. Out of various modularization strategies, the two major methods presented in the document focus on optimizing the 6G NFs or modules based on either the 5GC procedures that they are involved in or to certain KPIs such as latency (E2E), procedure completion time, amount of CP signalling, and at the same time reduce complexity. Through mathematical modelling, the trade-off between flexibility and performance has been investigated; a higher degree of granularity leads to higher flexibility but the E2E delay could be higher for some of the use cases (e.g., immersive telepresence). The E2E modular enabler focuses more on the E2E design of different functions and nodes, such as the CP/UP design, analysis of the RAN and CN interfaces, as well as modular orchestration. The key findings are that the 6G RAN architecture needs to support flexible deployments, such as D-MIMO, either by continuing using the higher layer CU/DU split as in 5G or by using a lower layer split with RAN and the radio unit. Further on, it is found that there are no significant advantages to utilize a service based interface for the RAN – CN connection. A concept for how to design a more modular UPF is proposed, splitting the UPF in smaller parts which can be scaled up or down, depending on traffic demands, to improve the usage of the network resources and improve sustainability. Finally, the increased flexibility and the opportunity to deploy the 6G NFs and modules, brings out the need for a new orchestration mechanism. In network autonomy and adaptiveness via modularization, the required orchestration mechanism to effectively manage the network modules is analysed. As a candidate solution, intent-based network orchestration is considered. The distributed nature of the envisioned 6G NFs/modules also necessitates highly accurate and effective synchronization which is quite challenging with conventional methods.

New access and flexible topologies enablers consist of network of networks, multi-connectivity (MC) and E2E context awareness management. The network of networks enabler mainly aims to improve coverage and mobility performance with the use of NTN and subnetworks. One main component in the network of networks enabler is to inherently support NTN in 6G, with improved interoperability between satellites and integration with the terrestrial networks. Subnetworks would use a management node, i.e., a more capable UE, to control other UEs. The aim is primarily to reduce the complexity of the UEs in the subnetwork and increase battery

life. However, increased device functionalities will be required for the management node. At the same time, trustworthiness between the nodes of a subnetwork is of the highest importance, when forming the subnetwork. Multi-connectivity enables simultaneous connections to different physically separated nodes and/or aggregating resources from different frequencies. The MC can be both within 6G or with 6G and other access networks, such as WiFi, to increase the system's robustness and reliability. Aggregating carriers via CA/DC is only beneficial if the throughput is relatively equal from all carriers (i.e., similar bandwidths and coverage). However, higher reliability may still be achieved even with unequal carriers. Further on, it is identified that faster addition of secondary cells compared to 5G is needed, in order to reduce the power consumption. Context awareness is a mechanism to allow network components to dynamically adapt to the context to ensure the expected E2E QoS. One way to introduce context awareness to the network is that a resource orchestrator creates an abstracted view of transport resources and employs a software defined transport controller for resource management, ensuring that the QoS associated with a network slice is met. With context awareness, the use of the available communication and compute resources would be optimized by allowing the resource orchestrator to take into consideration the requirements of the user's tasks (e.g., maximum affordable delay).

The **beyond communication** enabler deals with how to enable new 6G services such as sensing and compute offloading, and how to expose resulting data and relevant service capabilities in a secure, privacy-preserving and efficient manner. The exposure and data management enabler therefore aims to reduce the overhead from data exposure by aggregating and fusing data while ensuring data privacy and trust. This enabler supports the creation of novel services that contribute to societal benefits like safety and sustainability, supported by its capability to efficiently handle and expose data from various producers, including the RAN and sensing nodes. Further on, there is a need to develop novel interfaces and APIs, enabling third-party applications to request, control, and manage data securely and efficiently, thereby reducing data traffic and overhead significantly. Exact information on the position of base stations and UEs can be used for sensing to enable some kind of QoS-based sensing. However, if there is any doubt about the position accuracy of measurements nodes, e.g., a UE position with some uncertainty, it is likely that sensing services will have to be provided by the network on a best-effort basis. One way to improve sensing quality is obviously to carry out more radio measurements prior to exposure of the measurement report to the requesting application, preferably measurements that are geographically distributed. Involving more network nodes (UEs or base stations) naturally leads to architectural challenges of centralized vs. distributed inference and processing of the measurements. The provision of sensing services by next generation communication systems necessitates the introduction of a Sensing Management Function (SeMF) that will be responsible for facilitating an efficient coordination of sensing procedures, considering various aspects such as sensing requirements, sensing capabilities, sensing constraints, etc. The SeMF can be designed as a dedicated NF, since it is enabling a new functionality for next generation networks. An alternative option would be to integrate the SeMF services as part of the location management function of 5G.

The off-the-shelf cloud is suitable for a big subset of multimedia human-scale applications, but it has its limitations when it comes to supporting the upcoming latency-sensitive 6G use cases. Therefore, there is a for a **cloud transformation** to a cloud platform better suited for 6G. One important aspect of this is to the integration of compute continuum and the extreme edge. The integration and orchestration of extreme edge resources in the compute continuum defines the needed architectural interfaces and components for seamless orchestration and management of the complete compute continuum composed of cloud, edge and extreme edge resources. Therefore, new architectural mechanisms are proposed for the MEC framework to incorporate the extended compute continuum and account for the extreme edge devices in order to improve e.g., latency and energy efficiency. The Multi-domain/Multi-cloud federation enabler aims to aggregate cloud services provided by multiple domains and providers into a single, coherent cloud in order to reduce complexity for cross-domain deployment. Federation represents a transformative approach in which multiple, independent service providers collaboratively pool their resources to deliver seamless and scalable services. To this end, the cloud continuum should provide intent-based interfaces for cloud services and the core network should provide intent-based interfaces for network services.

10 References

[22.137] 3GPP TS 22.137 “Integrated Sensing and Communication”, V19.0.0, December 2023

[22.621] 3GPP, TS 22.621 “Service requirements for the 5G system;”, V19.5.0, December 2023.

[22.837] 3GPP TR 22.837 “Feasibility Study on Integrated Sensing and Communication”, V19.1.0, September 2023.

[23.288] 3GPP 23.288 “Architecture enhancements for 5G System (5GS) to support network data analytics services”, V18.4.0, December 2023.

[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;(Release 19),” V19.0.0, 2023.

[23.700-82] 3GPP TS 23.700-82 “Study on application layer support for AI/ML services”, Release 19 (v0.3.0), March 2024 (in progress).

[24.501] 3GPP TS 24.501 “Non-Access-Stratum (NAS) protocol for 5G System (5GS); Stage 3”, V18.5.0, December 2023.

[26.531] 3GPP TS 26.531, “Data Collection and Reporting; General Description and Architecture”, V18.0.0, September 2023

[29.502] 3GPP TR 29.502 “Technical Specification Group Core Network and Terminals; 5G System; Session Management Services; Stage 3 (Release 18)”, V18.1.0, December 2022

[29.503] 3GPP TS 29.503 “5G System; Unified Data Management Services; Stage 3”, V18.4.0, February 2024

[29.510] 3GPP TS 29.510 “5G System; Network function repository services; Stage 3”, V18.5.0, December 2023.

[37.340] 3GPP TS 37.340 “NR Multi-connectivity”, V18.0.0, January 2024.

[38.300] 3GPP TS 38.300 “NR and NG-RAN Overall Description; Stage 2”, V17.5.0, June 2023.

[38.331] 3GPP TS 38.331 “NR Radio Resource Control (RRC)”, V18.0.0, January 2024.

[38.401] 3GPP TS 38.401 “NG-RAN; Architecture description”, V18.0.0, December 2023.

[38.410] 3GPP TS 38.410 “NG-RAN; NG general aspects and principles”, V18.0.0, December 2023.

[38.413] 3GPP TS 38.413 “NG-RAN; NG Application Protocol (NGAP)”, V18.0.0, December 2023.

[38.801] 3GPP TR 38.801 “Study on new radio access technology: Radio access architecture and interfaces”, V14.0.0, March 2017.

[38.817-01] 3GPP TR 38.817-01 “General aspects for User Equipment (UE) Radio Frequency (RF) for NR,” v16.4.0, September 2022.

[38.817-02] 3GPP TR 38.817-02 “ General aspects for Base Station (BS) Radio Frequency (RF) for NR,” V15.11.0, September 2023.

[38.821] 3GPP, “Solutions for NR to support non-terrestrial networks (NTN),” v16.2.0, April 2023.

[802.1Qbv] IEEE 802.1Qbv-2015 IEEE Standard for Local and metropolitan area networks -- Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic.

[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.

[AAU+23] M. Z. Ali, A. Abohmra, M. Usman, A. Zahid, H. Heidari, M. A. Imran, and Q. H. Abbasi, “Quantum for 6G communication: A perspective,“ IET Quant. Comm. Vol.4: no. 112–124, 2023.

[AKE+24] O. U. Akgul, S. Kuklinski, M. Ericson, H. Harkous, R. Querio, A. Varvara, B. Arar, B. M. Khorsandi, S. Wänstedt, R Bassoli, F. H. P. Fitzek, “6G Function Modularity: Benefits, Challenges, and Options”, 2024 IEEE Wireless Communications and Networking Conference (WCNC), 2024.

[AZD+16] J. G. Andrews, X. Zhang, G. D. Durgin, and A. K. Gupta, “Are we approaching the fundamental limits of wireless network densification?” IEEE Communications Magazine, vol. 54, no. 10, pp. 184–190, 2016.

[BCD+14] S. Branzei, Y. Chen, X. Deng, A. Filos-Ratsikas, S. Frederiksen, and J. Zhang, “The fisher market game: Equilibrium and welfare,” in Proc. AAAI Conf. Artif. Intell., vol. 28, no. 1, 2014.

[BD04] P. Bolton, and M. Dewatripont, “Contract theory”, MIT press, 2004.

[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, 2022, p. 477–490.

[BV04] S.P. Boyd, and L. Vandenberghe, “Convex Optimization”, Cambridge University press, 2004.

[BVB+22] J. Baranda, L. Vettori, B. Bakhsh and J. Mangues. (2022, Sept. 20). ZSM-based Orchestration for Inter-Administrative Domain Cross-Border Vehicular Scenarios. IEEE International Conference on Sensing, Communication, and Networking. Zenodo. https://doi.org/10.5281/zenodo.7341444. Online: https://zenodo.org/records/7341444.

[CB17] H. Corrigan-Gibbs and D. Boneh Prio, “Private, Robust, and Scalable Computation of Aggregate Statistics,” Stanford University, March 14, 2017.

[CB24] A. Clark, C. Baker, "Rural mobile coverage in the UK: Not-spots and partial not-spots", Commons Library Research Briefing, Number CBP 7069, 1 March 2024. https://commonslibrary.parliament.uk/research-briefings/sn07069/ latest access 3 March 2024.

[CBC+21] D. Casini, A. Biondi, G. Cicero, G. Buttazzo, “Latency Analysis of I/O Virtualization Techniques in Hypervisor-Based Real-Time Systems,” 2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS), Nashville, TN, USA, 2021, pp. 306-319, doi: 10.1109/RTAS52030.2021.00032, 2021. [Online]. Available: https://retis.sssup.it/~a.biondi/papers/RTAS21-IOvirt.pdf

[CCC+23] M. Chatzidakis, J. Chen, O. Chick, E. Circlaeays, S. Gopalan, Y. Goren, K. Guo, M. Hesse, O. Javidbakht, V. Jina, K. Kalu, A. Katti, A. Liu, R. Low, A. McMillan, J. Meyer, S. Myers, A. Palmer, D. Park, G. Parsa, P. Pelzl, R .Rishi, M. Scaria, C. Sumanth, K. Talwar, K. Tarbe, S. Wang, and M. Yadav, “Learning Iconic Scenes with Differential Privacy”, Apple Machine Learning Research, July 2023, Online: https://machinelearning.apple.com/research/scenes-differential-privacy

[CDP23] P. Charatsaris, M. Diamanti and S. Papavassiliou, "On the Accuracy-Energy Tradeoff for Hierarchical Federated Learning via Satisfaction Equilibrium," 2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), Pafos, Cyprus, 2023, pp. 422-428, doi: 10.1109/DCOSS-IoT58021.2023.00073.

[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.

[DFM+21] M. Ding, Z. Feng, D. Marpaung, X. Zhang, M. Komanec, D. Suslov, D. Dousek, S. Zvánovec, E. R. Numkam Fokoua, T. D. Bradley, F. Poletti, D. J. Richardson, and R. Slavík, “Optical Fiber Delay Lines in Microwave Photonics: Sensitivity to Temperature and Means to Reduce it,” JOURNAL OF LIGHTWAVE TECHNOLOGY, VOL. 39, NO. 8, APRIL 15, 2021, https://ris.utwente.nl/ws/portalfiles/portal/275570122/Ding2021optical.pdf

[DZF+20] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar and A. Y. Zomaya, “Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence,” in IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7457-7469, Aug. 2020, doi: 10.1109/JIOT.2020.2984887.

[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

[EGG20] O. Esrafilian, R. Gangula and D. Gesbert, "Autonomous UAV-aided Mesh Wireless Networks," *IEEE INFOCOM 2020 – IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, 2020, pp. 634-640, doi: 10.1109/INFOCOMWKSHPS50562.2020.9162753.

[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

[Fou16] M. P. Fourman, "Measuring the Broadband Access Divide", Proceedings of the Eighth International Conference on Information and Communication Technologies and Development (ICTD'16), Article No.: 33, June 2016, https://doi.org/10.1145/2909609.2909616

[Free5GC] "Free5GC." Accessed: Jul. 2022. [Online]. Available: https://free5gc.org/

[FTM+16] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE P802.11-REVmc/D8.0", August 2016, pages 1–3774, 2016.

[GCF+22] M. Gramaglia, M. Camelo, L. Fuentes, J. Ballesteros, G. Baldoni, L. Cominardi, A. Garcia-Saavedra, and M. Fiore, "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

[GPR+23] T. Geoghegan, C. Patton, E. Rescorla, and C. A. Wood "Distributed Aggregation Protocol for Privacy Preserving Measurement," IETF. July 10, 2023.

[GSH+22] 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, doi: 10.1109/TNSM.2022.3230206.

[HAR22] Hasanin 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

[HEX223-D12] Hexa-X-II Deliverable D1.2 "6G Use Cases and Requirements," Hexa-X-II project, 2023, [Online]. Available: https ://hexa-x-ii.eu/wp-content/uploads/2024/01/Hexa-X-II_D1.2.pdfAvailable: https ://hexa-x-ii.eu/wp-content/uploads/2024/01/Hexa-X-II_D1.2.pdf

[HEX223-D21] Hexa-X-II Deliverable D2.1 "Draft foundation for 6G system design," Hexa-X-II project, 2023, [Online]. Available: https://hexa-x-ii.eu/wp-content/uploads/2023/07/Hexa-X-II_D2.1_web.pdf

[HEX223-D22] Hexa-X-II Deliverable D2.2 "Foundation of overall 6G system design and preliminary evaluation results," Hexa-X-II project, 2023, [Online]. Available : https ://hexa-x-ii.eu/wp-content/uploads/2024/01/Hexa-X-II_D2.2_FINAL.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

[HEX223-D42] Hexa-X-II Deliverable D4.2 "Radio Design and Spectrum Access requirements and key enablers for 6G Evolution," Hexa-X-II project, 2023, [Online]. Available : Hexa-X-II_D4_2_final.pdf

[HEX224-D53] Hexa-X-II Deliverable D5.3 "Initial design and validation of technologies and architecture of 6G devices and infrastructure," Hexa-X-II project, 2024, [Online]. Available: https://hexa-x-ii.eu/wp-content/uploads/2024/02/Hexa-X-II_D5.3_v1.0.pdf

[HEX22-D13] Hexa-X Deliverable D1.3 "Targets and requirements for 6G – initial E2E architecture," Hexa-X project, 2022, [Online]. Available : https://hexa-x.eu/wp-content/uploads/2022/03/Hexa-X_D1.3.pdf

[HEX22-D52] Hexa-X Deliverable D5.2, "Analysis of 6G architectural enablers' applicability and initial technological solutions", ", Hexa-X project, Oct. 2022, [Online]: https://hexa-x.eu/wp-content/uploads/2022/10/Hexa-X_D5.2_v1.0.pdf

[HEX22-D62] Hexa-X Deliverable D6.2 "Design of service management and orchestration functionalities", Hexa-X project, 2022, [Online]. Available: https://hexa-x.eu/wp-content/uploads/2022/05/Hexa-X_D6.2_V1.1.pdf

[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/2023/08/Hexa-X_D5.3_v1.1.pdf

[HEX23-D63] Hexa-X Deliverable D6.3, "Final evaluation of service management and orchestration mechanisms", April 30, 2023, Online: https://hexa-x.eu/wp-content/uploads/2023/05/Hexa-X_D6.3_v.1.1.pdf

[IJR+22] M. Iovene, L. Jonsson, D. Roeland, M. D'Angelo, G. Hall, M. Erol-Kantarci, and J. Manocha, "Defining AI native: A key enabler for advanced intelligent telecom networks", Ericsson Whitepaper, 2022. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/white-papers/ai-native.

[INF001] ETSI GS NFV-INF 001, Network Functions Virtualisation (NFV); Infrastructure Overview, v1.1.1, ETSI, Sophia Antipolis, Jan 2015.

[ITU23] ITU, "World Radiocommunication Conference revises the ITU Radio Regulations to support spectrum sharing and technological innovation ", Press Release, December 2023, available at: https://www.itu.int/en/mediacentre/Pages/PR-2023-12-15-WRC23-closing-ceremony.aspx

[K3S] K3S Lightweight Kubernetes [Online]. Available at: https://k3s.io (Accessed: 4 Dec. 2023).

[K8S] Kubernetes [Online]. Available at: https://kubernetes.io (Accessed: 4 Dec. 2023).

[KKB+14] P. Komar, E. M. Kessler, M. Bishof, L. Jiang, A. S. Sorensen, J. Ye, and M. D. Lukin, "A quantum network of clocks," Nature Physics, vol. 10, no. 8, pp. 582–587, 2014.

[KKT+21] S. Kuklinski, R. Kołakowski, L. Tomaszewski, L. Sanabria-Russo, C. Verikoukis, C.-T. Phan, L. Zanzi, F. Devoti, A. Ksentini, C. Tselios, G. Tsolis and H. Chergui, "MonB5G: AI/ML-Capable Distributed Orchestration and Management Framework for Network Slices," 2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom), Athens, September 2021.

[KNE16] K. Katsalis, N. Nikaein and A. Edmonds, "Multi-Domain Orchestration for NFV: Challenges and Research Directions," 2016 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS), Granada, Spain, 2016, pp. 189-195, doi: 10.1109/IUCC-CSS.2016.034.

[KT18] S. Kuklinski, L. Tomaszewski, "DASMO: A scalable approach to network´ slices management and orchestration," NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium, pp. 1-6, 2018.

[KTK+21] S. Kuklinski, L. Tomaszewski, R. Kołakowski and P. Chemouil, "6G-´ LEGO: A framework for 6G network slices" in Journal of Communications and Networks, vol. 23, no. 6, pp. 442-453, 12/2021, doi: 10.23919/JCN.2021.000025.

[KTO+18] S. Kuklinski, L. Tomaszewski, T. Osinski, A. Ksentini, P. Frangoudis, E. Cau and M. Corici, A reference architecture for network slicing, 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), pp. 217-221, 2018.

[LBS+22] M. Liyanage, A. Braeken, S. Shahabuddin, and P. Ranaweera, "Open RAN Security: Challenges and Opportunities," arXiv:2212.01510 [cs.CR]

[LC15] Y. Lee and D. Ceccarelli, "Abstraction and Control of transport Networks," in Handbook of Research on Redesigning the Future of Internet Architectures, pp. 346-363, IGI Global, 2015, doi: 10.4018/978-1-4666-8371-6.ch015

[LCO+23] D. Larrabeiti, L. M. Contreras, G. Otero, J. A. Hernández, J. P. Fernandez-Palacios, "Toward end-to-end latency management of 5G network slicing and fronthaul traffic (Invited paper),"

Optical fiber technology, vol. 76, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1068520022004059

[LCW+20] S. Luo, X. Chen, Q. Wu, Z. Zhou and S. Yu, "HFEL: Joint Edge Association and Resource Allocation for Cost-Efficient Hierarchical Federated Edge Learning," in IEEE Transactions on Wireless Communications, vol. 19, no. 10, pp. 6535-6548, Oct. 2020, doi: 10.1109/TWC.2020.3003744.

[LMB+15] T.Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common Objects in Context", arXiv, 2015. [Online]. Available: https://arxiv.org/abs/1405.0312

[LYC+22] S. Liu, G. Yu, X. Chen and M. Bennis, "Joint User Association and Resource Allocation for Wireless Hierarchical Federated Learning With IID and Non-IID Data," in IEEE Transactions on Wireless Communications, vol. 21, no. 10, pp. 7852-7866, Oct. 2022, doi: 10.1109/TWC.2022.3162595.

[LZM+22] Y. Liu, S. Zhang, X. Mu, Z. Ding, R. Schober, N. Al-Dhahir, E. Hossain, X. Shen, ., "Evolution of NOMA Toward Next Generation Multiple Access (NGMA) for 6G," in IEEE Journal on Selected Areas in Communications, vol. 40, no. 4, pp. 1037-1071, April 2022, doi: 10.1109/JSAC.2022.3145234.

[MAM+22] N. Mhaisen, A. A. Abdellatif, A. Mohamed, A. Erbad and M. Guizani, "Optimal User-Edge Assignment in Hierarchical Federated Learning Based on Statistical Properties and Network Topology Constraints," in IEEE Transactions on Network Science and Engineering, vol. 9, no. 1, pp. 55-66, 1 Jan.-Feb. 2022, doi: 10.1109/TNSE.2021.3053588.

[MEC003] ETSI GS MEC 003: Multi-Access Edge Computing (MEC); Framework and Reference Architecture, ETSI Std. v2.2.1, 2020.

[MEC012] ETSI GS MEC 012, Multi-access Edge Computing (MEC); Radio Network Information API, v2.1.1, ETSI, Sophia Antipolis, December 2019.

[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.

[MGB+23] Picazo Martinez, P., Groshev, M., Blanco, A., Fiandrino, C., de la Oliva, A., & Widmer, J. (2023, November 21). waveSLAM: Empowering Accurate Indoor Mapping Using Off-the-Shelf Millimeter-wave Self-sensing. IEEE Vehicular Technology Conference, Hong Kong, China. https://doi.org/10.5281/zenodo.10171206

[MHH+22] J. Meng, J. Huang, Y.C. Hu, Y. Koral, X. Lin, M. Shahbaz, A. Sharma, Characterizing and Modeling Control-Plane Traffic for Mobile Core Network, arXiv - CS - Networking and Internet Architecture, 2022-12-26 , DOI:arxiv-2212.13248

[MYL+23] B. Maa, C. Yanga, A. Lia, Y. Chia, L. Chen, "A Faster DBSCAN Algorithm Based on Self-Adaptive Determination of Parameters," 10th International Conference on Information Technology and Quantitative Management, 2023, https://www.sciencedirect.com/science/article/pii/S1877050923007135

[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.

[NCV+23] J. Nasreddine, E. Carmona-Cejudo, R. Vilalta, R. Parada, P. Veyssiere, A. Antonopoulos, J. López Luque, J. Bastida, R. González, G. N. Triantafyllou, and F. Vázquez-Gallego, "5GMED Architecture for Automotive and Railway Communication Services in Cross-Border Scenarios,". IEEE Future Networks World Forum (FNWF), Canada, 2023. https://doi.org/10.1109/FNWF55208.2022.00015. [Online]. Available: https://zenodo.org/records/7544378#.ZGIEe5MzYfF.

[NFV002] ETSI GS NFV 002, Network Functions Virtualisation (NFV); Architectural Framework, v1.2.1, ETSI, Sophia Antipolis, December 2012.

[NOMAD] Nomad [Online]. Available at: https://www.nomadproject.io (Accessed: 4 Dec. 2023).

[NPS+23] S. Nande, M. Paul, S. Senk, M. Ulbricht, R. Bassoli, F. H.P. Fitzek, and H. Boche, "Quantum enhanced time synchronisation for communication network," Computer Networks, vol. 229, pp. 109772, 2023.

[NSZ+22] J. Niemöller, R. Szabó, A. Zahemszky and D. Roeland, "Creating autonomous networks with intent-based closed loops," in Ericsson Technology Review, vol. 2022, no. 4, pp. 2-11, April 2022.

[OAD24] O-RAN Work Group 1 (Use Cases and Overall Architecture), "O-RAN Architecture Description", V11.00, February 2024

[OBT+16] B. Oljira, A. Brunstrom, J. Taheri and K. -J. Grinnemo, "Analysis of Network Latency in Virtualized Environments," IEEE Global Communications Conference (GLOBECOM 2016), 2016, pp. 1-6, doi: 10.1109/GLOCOM.2016.7841603.

[OSA24] O-RAN Work Group 1 (Use Cases and Overall Architecture), "Slicing Architecture", V12.00, February 2024

[P4] [Online]: https://p4.org/specs/

[PBD+23] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges," IEEE Communications Surveys & Tutorials, vol. 25, no. 2, pp. 1376–1411, 2023.

[PFH+20] Q. -V. Pham, F. Fang, V. N. Ha, Md. J. Piran, M. Le, L. B. Le, W. -J. Hwang and Z. Ding, "A Survey of Multi-Access Edge Computing in 5G and Beyond: Fundamentals, Technology Integration, and State-of-the-Art," in IEEE Access, vol. 8, pp. 116974-117017, 2020, doi: 10.1109/ACCESS.2020.3001277.

[PTL+10] S. M. Perlaza, H. Tembine, S. Lasaulce and M. Debbah, "Satisfaction Equilibrium: A General Framework for QoS Provisioning in Self-Configuring Networks," 2010 IEEE Global Telecommunications Conference GLOBECOM 2010, Miami, FL, USA, 2010, pp. 1-5, doi: 10.1109/GLOCOM.2010.5685235.

[RFC8402] C. Filsfils, Ed. S. Previdi, Ed. L. Ginsberg, B. Decraene, S. Litkowski, R. Shakir, "Request for Comments: 8402, Segment Routing Architecture," Internet Engineering Task Force (IETF), July 2018.

[RGO+23] 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.

[Sab21] D. Sabella, Multi-access Edge Computing: Software Development at the Network Edge," Springer Cham, 2021.

[SBA] [Online]: https://www.confluent.io/blog/kafka-fastest-messaging-system/

[SDR+23] M. Saimler, M. D'Angelo, D. Roeland, A. Ahmed, and A. Kattepur, "AI as a service: How AI applications can benefit from the network", Ericsson Blogpost, 2023. [Online]. Available: https://www.ericsson.com/en/blog/2023/12/ai-as-a-service

[SRK+22] M. Schulz, M. Ruefenacht, D. Kranzlmüller and L. B. Schulz, "Accelerating HPC With Quantum Computing: It Is a Software Challenge Too," in Computing in Science & Engineering, vol. 24, no. 4, pp. 60-64, 1 July-Aug. 2022, doi: 10.1109/MCSE.2022.3221845.

[STK+23] E. U. Soykan, E. Tomur, F. Karakoc. Hexa-X and data protection evolution in 6G. October 05, 2023, Ericsson Blogpost. [Online]. Available: https://www.ericsson.com/en/blog/2023/10/hexa-x-and-data-protection-evolution-in-6g

[SWARM] Docker Swarm [Online]. Available at: https://docs.docker.com/engine/swarm (Accessed: 4 Dec. 2023).

[TP19] T. X. Tran and D. Pompili, "Joint Task Offloading and Resource Allocation for Multi-Server Mobile-Edge Computing Networks," in IEEE Transactions on Vehicular Technology, vol. 68, no. 1, pp. 856-868, Jan. 2019, doi: 10.1109/TVT.2018.2881191.

[TWM+23] K. Talwar, S. Wang, A. McMillan, V. Jina, V. Feldman, B. Basile, A. Cahill, Y.S. Chan, M. Chatzidakis, J. Chen, O. Chick, M. Chitnis, S. Ganta, Y. Goren, F. Granqvist, K. Guo, F. Jacobs, O. Javidbakht, A. Liu, R. Low, D. Mascenik, S. Myers, D. Park, W. Park, G. Parsa, T. Pauly, C. Priebe, R. Rishi, G. Rothblum, M. Scaria, L. Song, C. Song, K. Tarbe, S. Vogt, L. Winstrom, S. Zhou, "Samplable Anonymous Aggregation for Private Federated Data Analysis", Apple, July 2023

[TZL+22] X. Tu, K. Zhu, N. C. Luong, D. Niyato, Y. Zhang, and J. Li, "Incentive mechanisms for federated learning: From economic and game theoretic perspective," IEEE Trans. On Cogn. Commun. Netw., vol. 8, no. 3, pp. 1566–1593, 2022.

[UKK+23] A. Ullah, T. Kiss, J. Kovács, F. Tusa, J. Deslauriers, H. Dagdeviren, R. Arjun, and H. Hanzeh, "Orchestration in the Cloud-to-Things Compute Continuum: Taxonomy, Survey and Future Directions," Journal of cloud computing, vol. 12, pp 135, 2023.

[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

[VCL24] L. Vinkhuijzen, T. Coopmans, A. Laarman, "A Knowledge Compilation Map for Quantum Information," arXiv:2401.01322v1 [quant-ph] 2 Jan 2024.

[YCS+21] Z. Yang, M. Chen, W. Saad and M. Shikh-Bahaei, "Optimization of Rate Allocation and Power Control for Rate Splitting Multiple Access (RSMA)," in IEEE Transactions on Communications, vol. 69, no. 9, pp. 5988-6002, Sept. 2021, doi: 10.1109/TCOMM.2021.3091133.

[YOL+21] Ge, Zheng and Liu, Songtao and Wang, Feng and Li, Zeming and Sun, Jian, "Yolox: Exceeding yolo series in 2021", arXiv preprint arXiv:2107.08430

[ZLH23] T. Zhao, F. Li, and L. He, "DRL-Based Joint Resource Allocation and Device Orchestration for Hierarchical Federated Learning in NOMA-Enabled Industrial IoT," in IEEE Transactions on Industrial Informatics, vol. 19, no. 6, pp. 7468-7479, June 2023, doi: 10.1109/TII.2022.3170900.

[ZSM012] ETSI ZSM, "GS ZSM 012: Zero-touch network and Service Management (ZSM); Enablers for Artificial Intelligence-based Network and Service Automation," ETSI Std. v1.1.1, 2022.

[NMS+22] N. P. Kuruvatti, M. A. Habibi, S. Partani, B. Han, A. Fellan and H. D. Schotten, "Empowering 6G Communication Systems With Digital Twin Technology: A Comprehensive Survey," in IEEE Access, vol. 10, pp. 112158-112186, 2022, doi: 10.1109/ACCESS.2022.3215493.

[RLR+23] I. Rahman, O. Liberg, S. M. Razavi, C. Hoymann, S. Parkvall, G. Rune, R. Keller, P. Persson, A. Grövlen, D. C. Larsson, "5G Advanced: Evolution towards 6G", Ericsson Whitepaper, 2023. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/white-papers/5g-advanced-evolution-towards-6g

11 Annex A: Further details of the studies

11.1 Detailed Versions of Studies in MLOPs

11.1.1 Federated learning approach between different city verticals

In the current paradigm of the Internet of Things (IoT), there are more and more devices around us with computational capabilities; said devices are also gathering relevant data from which we can extract knowledge by processing it. Interested companies usually do this process with local learning models; however, this venture has some drawbacks, as sending data from all of these devices to a central server weighs on bandwidth costs and reliable latency requirements. To solve these issues, there needs to be a shift from a centralised server network paradigm to an edge server network paradigm.

Edge computing can be defined as a "distributed computing framework that brings enterprise applications closer to data sources"; the interaction between these devices closer to data sources and a central server gives the network potential to process workloads far more significant than what a single machine could handle, by distributing said workload among the devices in this edge computing network, and effectively splitting the computation cost. However, it raises issues regarding the proper splitting of workloads, privacy-preserving data usage of models by edge devices and the high resource costs of managing all network devices.

Federated Learning is the process by which we decentralise the network model for ML, in which, instead of running an ML model on our server with the aggregated data from all of our edge devices, the central server "uploads" the training model to the edge devices for them to process and train said model on the collected raw data and sending the model's respective gradient to the central server for aggregation. With various optimisations, this aggregation produces a result that converges to the result that a global model would produce on the whole of the data, with very high levels of accuracy. This framework solves some of the previously raised problems:

- Splitting the workload is streamlined by model uploading and model aggregation;
- Data privacy is preserved as raw data never leaves the edge devices, only the gradient vector of the model weights.

Smart cities use information and communication technologies to improve operational efficiency, share information with the public and provide a better quality of government service and citizen welfare. The success of an intelligent city hinges upon the reliability of its services and the capacity to improve and adapt to the growing and changing demands while also increasing the quality of life for its population; to achieve this, data collection and analysis, communication and action are critical steps for this framework to succeed.

Federated Learning on Edge computing is very relevant for smart cities, as it is a framework that complements well to enhance its functionalities and responds well to an intelligent city's goals. While addressing some of the faults that Edge Computing has, there are better solutions than implementing Federated Learning on the network.

Communication has to be minimised and optimised, as model updates from the central server and trained model uploads from edge devices still have a bandwidth cost; solutions to this problem are data transfer optimisation algorithms (minimising the number of updates necessary through a distance-aware algorithm, predicting data flow and using data caching in edge devices), Asynchronous Federated Learning (that can aggregate models asynchronously in the central server or along the edge), Federated Clustering (by aggregating trusted edge devices to enhance global model accuracy), Customized Partial Client Participation (with a flexible aggregation policy and restraining the upload times), among others.

The computational needs of the network are also dependent more on edge devices. As we decentralise the responsibilities of a central server, more pressure is put on the edge, and issues of heterogeneity of edge device training performance and computational capabilities are raised; some solutions to these problems are optimising participant selection (minimising service costs by using an auction-based device selection that picks the best users for the workload, classifying devices into suitable adaptive clusters according to their local data distribution), Task offloading policy (that reduces task processing delay and computing energy consumption), among others.

Energy consumption is also a factor in play, as the demands of intelligent cities require more processes to happen on the edge network, which in turn requires more devices and more computation on said devices, which raises not only the cost of implementation of services but the cost of running these services. As such, optimising the functionality of the network is a critical role in improving the capabilities of our framework; to tackle these necessities, some proposals are to improve the decision-making of the network (edge caching popular contents based on the global model, a device-to-device strategy that uses the energy of neighbouring devices for local model uploading), optimising resource allocation (selecting best edge devices for the workload according to their capabilities), energy procurement prediction (dealing with uncertain energy requirements between devices), among others.

While Federated Learning addresses some issues regarding data privacy by removing the need to transfer raw data to the central server, it does not make it impervious to attacks on the privacy or security of data. Malicious devices on the edge network can not only intercept the model uploaded by the central server and use that information to infer raw data on other devices' uploaded models, but they can also upload data poisonous to the global model aggregation. Some solutions to this are vulnerability detection of edge devices (using Generative Adversarial Networks to protect model parameters, weight-based anomaly detection against poisoning attacks and privacy-preserving data, data filtering in dealing with poisoning attacks), creating a trust system to label the edge devices on their performance, protecting against backdoor attacks introduced by data points with unusual features, among others.

Finally, the heterogeneity of edge devices also poses a problem to the network, as different devices have different computational and communication capabilities, and local training does not guarantee devices are using non-IID raw data, which may lead to unbalanced local data samples and impact the global model after uploading for aggregation. Some solutions to this are edge device selection (through algorithms with a reduction heuristic or greedy-decay heuristic), dealing with potential non-IID (Independent Identically Distributed) data on the clients (using a pre-trained model for data augmentation, implementing a Federated Clustering algorithm among different end devices), using a client scheduling algorithm to reduce statistical heterogeneity and redundancy of data caused by multi-task learning, among others.

Under federated learning, multiple cities remotely share their data to collaboratively train a single deep learning model, improving on it iteratively, like a team presentation or report. Each party updates the model from an ML repository in the cloud, usually a pre-trained foundation model. They train it on their private data, then summarise and encrypt the model's new configuration. The model updates are returned to the cloud, decrypted, averaged, and integrated into the centralised model. Iteration after iteration, the collaborative training continues until the model is fully trained.

In vertical federated learning, which is our case, the data are complementary; parking and traffic data, for example, are combined to predict someone's route preferences. Finally, in federated transfer learning, a pre-trained foundation model designed to perform one task, like detecting parking availability, is trained on another dataset to do something else, like identify routes with minor traffic.

This model, using the edge nodes for local learning, will address the issues of data privacy and resilience. All data concerning city A will remain locally. What is updated on the central node are the algorithms used without all the private information, and therefore without the concerns of what information types and privacy, and without all the amount of data locally present on each edge. These will be managed by a centralised ML Catalogue, shared by all, and all these communications will be encrypted in transit.

Regarding network requirements needed to ensure communications, Heterogeneous communication systems that can support various innovative wireless technologies, services and applications are required. To do so, the 6G Radio Access Network is deployed standalone and connected to a so-called E-5GC. 6G intra-RAT CA and 6G-6G DC combine capacity and coverage bands dynamically shared with 5G via MRSS. This solution will ensure shared communication between the nodes, enabling the exchange of information in a secure and reliable medium.

All these services must be integrated and standardised to achieve a global service, with model aggregation, directly fed from all the nodes after all the local modelling and training. The solution will present AIaaS, due to this architecture that will provide an edge to the cloud continuum with high availability and resilience.

The introduction of 5G Advanced [RLR+23] is characterized by the widespread adoption of AI. AI is currently used within the network to support complex tasks in various domains, With the journey towards 6G, we envision the network evolving into an AI-native platform that not only utilizes AI internally but also exposes its AI capabilities to support a wide range of applications. We refer to this concept as AIaaS. The idea is that the network, functioning as an AIaaS provider, can offer pre-built AI models, datasets, algorithms, and tools that applications can readily access through APIs. The primary advantage of AIaaS lies in enabling applications to leverage AI functionalities without having to construct and manage their own AI infrastructure. Consequently, the network transforms into a platform that fosters innovation for a variety of use cases.

Generally, support for MLOps is provided through a set of MLOps tools that can be implemented within the CSP domain (refer to Figure 11-1). The MLOps toolset(s) can be internally exposed to any of the network domains within the CSP domain. Arrow (1) in Figure 11-1 is an example of exposure to the RAN to support the training of AI-enabled network functions. The toolset(s) can also be made available to applications running on top of the CSP network. We refer to the latter as MLOps-aaS.

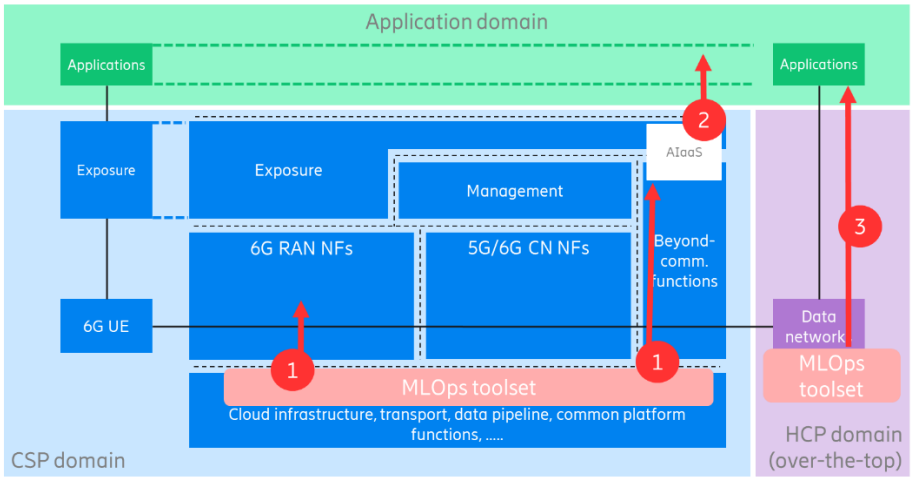


Figure 11-1 The FNP architecture with internal exposure of the MLOps toolset(s) to any of the network domains in arrow (1) and external exposure of AIaaS to applications in arrow (2)

AIaaS APIs are realized by API orchestration of the MLOps toolset (1) and of network functionalities in network domains such as RAN, CN, and management. The AIaaS box in Figure 11-1 includes API orchestration but also other functionalities such as authentication and access management. By implementing these functions, higher-level AI services can be realized and exposed to the application domain (2). We emphasize that an application can utilize MLOps-aaS also directly from an HCP — (3) in Figure 11-1.

AI plays a crucial role in enhancing the capabilities and efficiency of Robots-becoming-cobots use case. AI algorithms can be employed to enable robots to perceive and comprehend their environment, enhancing their adaptability and responsiveness to human interaction. Machine learning techniques can empower robots to learn from human feedback and optimize their performance over time. Additionally, AI can facilitate predictive maintenance, wherein algorithms analyse sensor data to identify potential issues and proactively schedule maintenance.

While the mentioned functionalities can be implemented and embedded as stand-alone capabilities within a robot application, a question arises: How can such robots leverage AI services provided and exposed by the network?

1. The AIaaS APIs might receive requirements that cannot be addressed by existing HCP offerings or implemented by the cobot itself. For example, the cobot application might request the creation and execution of a custom AI/ML model tailored specifically for its use case and consume the inference via the exposed API. Such a model may be designed to predict the cobots' locations, which may rely on a combination of data sources, including location sensor information from the cobots' application (e.g., GPS); cobot UE mobility events; or sensing data from within the CSP domain, e.g., by using network's antennas for sensing, see Figure 11-2.
2. The AIaaS APIs may receive the application's requirements and fulfil them by invoking relevant functions within the CSP domain. The specific functions invoked vary based on the application's

requirements but may encompass computational offload, sensing capabilities, or activate an orchestrator to provision additional resources or reconfigure specific network functions. For example, QoS guarantees on bandwidth and latency may trigger the setup of new bearers, or the allocation of RAN and CN resources deployed at the edge of the network to enable low-latency connections from cobot's UEs using the application.

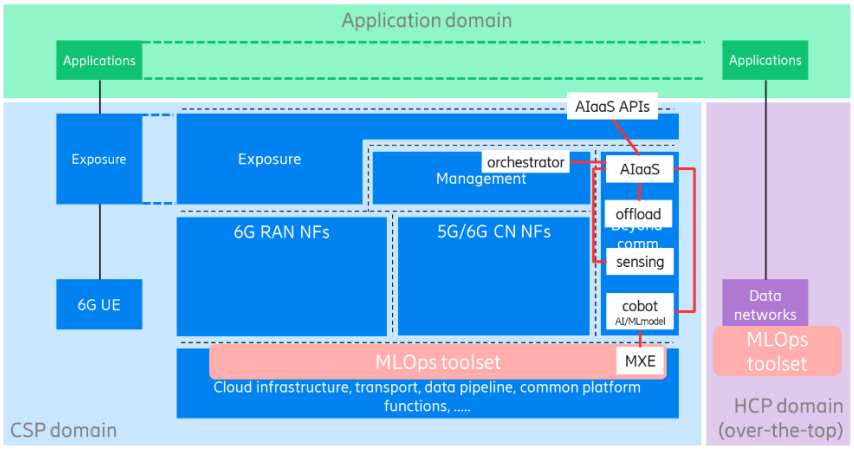


Figure 11-2 AIaaS for the cobots use case

Generalizing from the robot-to-cobots use case we envision a list of API families characterizing AIaaS in Figure 11-3.

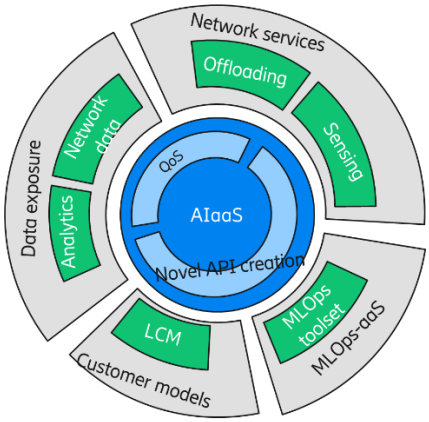


Figure 11-3 API families for AIaaS

The **MLOps-aaS API family** offers a set of tools and services for managing the complete life cycle of machine learning models. For example, it supplies the necessary tools and environment for training, deployment, and monitoring of models.

The **Exposure of network services API family** offers, for example: sensing and compute offload functionality.

The **Exposure of network data API family** provides applications with diverse network-related data and analytics, such as mobility events describing movement patterns, network load levels aiding adaptation to changing conditions, performance, and insights into energy consumption.

The **Life-cycle management of customer models API family** oversees the full lifecycle of customer-specific models. For example, it includes training of application models with a combination of network and application data and exposing APIs for model inferencing.

The **QoS APIs** may be used in combination with the other API families. For example, the application has an AI/ML model that requires a certain maximum latency during inference. In such a scenario, the application may package such a model within a container, request compute capability (from the network service API family) to run the container on, and further request a maximum latency through the QoS API.

11.1.2 Strategies and mechanisms for distributed AI and AIaaS functions management

The proposed study focuses on identifying and defining common services and functions for AI and AIaaS lifecycle management that enable support heterogeneous cloud-native deployments across the continuum while enabling different decentralized and cooperative AI functions. Specifically, as described in D3.2, the study aims at evolving the AIaaS solution initially proposed in Hexa-X, evaluating if the AI functions defined there are enough to support different deployment models required by distributed and federated AI/ML solutions.

Starting from this, the existing AIaaS approach and initial functional split has been revisited to consider AIaaS as a comprehensive framework exposing unified APIs for AI services management and consumption. With the aim of achieving a stand-alone and self-consisting AIaaS framework, dedicated AI (and APIs) management and coordination functions are required to properly handle different AI lifecycle aspects, including training, deployment, inference, etc. Indeed, training and runtime phase have normally different requirements (in terms of data, performance, execution environments, lifecycle etc.), and may be invoked and executed at different timescales and according to different application-level constraints.

Therefore, as depicted in Figure 11-4Figure , the AIaaS framework functional split is evolved to accommodate two different categories of functions: (logically) centralized management functions, and on-demand per AI service (distributed) functions. The (logically) centralized management functions (in blue in the Figure 11-4) aim at covering the AI service and functions lifecycle management gap, as well as providing coordinated execution of training and runtime AI services, taking care of exposing AI services through dedicated APIs accessible to external consumers. On the other hand, the on-demand per AI service (distributed) functions are the actual AI functions implementing AI services, and are different for the runtime (i.e., intended for serving/inference capabilities) and training phases. These, in line with what already defined in Hexa-X, include at least ML model serving and ML model monitoring for the runtime functions, and ML model training and ML model validation for the training functions. Additionally, local ML model repository/storage and local runtime/training data stores may be included according to the specific ML technique to be supported. These on-demand functions are specifically conceived to support distributed and cooperative deployments and operation (i.e., in support of federated, cooperative, distributed, parallel and split ML techniques), and to be executed as cloud-native virtual functions across the cloud continuum. Specifically, this study aims at providing an AIaaS framework that allows to deploy and execute on-demand runtime and training functions in cloud-native Kubernetes infrastructures.

In particular, the AI/ML Service Manager wraps and embeds the logics for managing and coordinating the deployment, configuration and execution of on-demand AI training and AI runtime services. The Training Manager is responsible to orchestrate the execution of the various pipelines on top of the cloud continuum, deploying and configuring the on-demand training functions, and managing re-trainings based on the feedback collected from monitoring and validation of ML models performances (from both training and runtime related functions). The Runtime Manager orchestrates the deployment and configuration of the on-demand runtime functions, taking care of their constraints in terms of execution (e.g., for distributed/cooperative ML techniques). The AI/ML Service Manager, beyond integrating the runtime and training coordination mechanisms, can also provide additional data distribution and management functionalities, leveraging when available, on DataOps tools facilitating the collection, maintenance and transfer of training and runtime data across the cloud continuum. Moreover, it also supports the Runtime and Training Managers in the on-demand functions placement (satisfying performance and deployment requirements). An ML model storage is kept as part of the (logically) centralized management functions to collect, maintain and track the whole set of ML models trained and available within the AIaaS framework. It is populated by the Training Manager and accessed by the Runtime Manager. In addition, with the aim exposing AI services towards external consumers, the AI/ML Catalogue encapsulate the available ML models with the required metadata for their description (in terms of capabilities, type of model, description of output, etc.), execution and consumption (e.g., with metadata for model invocation and inference execution). Specifically, the AI/ML Catalogue maintains both the AI services available in the framework (i.e., those related to ML models trained but not yet deployed and in execution), as well as those already running and ready to be consumed (also with the aim of enabling share and re-use of models). At the top of this evolved AIaaS framework, the AI/ML API Orchestrator represents

the main entry point, responsible to glue and hide all the internal logics. Its main aim is to expose towards external consumers a set of APIs which allow to: query available AI services (either to be provisioned or already provisioned and ready to be consumed), provision AI services (i.e., trigger the on-demand deployment and execution of either AI training or AI runtime services). These exposed APIs are intended to provide access to the various AI services available in the framework, and which depend on the given use cases to be supported, with their constraints in terms of performance, execution and deployment, and models that have been trained and maintained in the AI/ML Catalogue and ML model storage. Dedicated APIs are also supported by the AI/ML API orchestrator to onboard new AI services in the framework, e.g., for new training pipelines and services that when executed can generate new models and AI runtime services. Therefore, the AI/ML API Orchestrator wraps the access to the AI/ML Catalogue and serves as the coordinator of the AI/ML service manager internal workflows and logics. In addition, it can embed discovery functionalities for smart selection of available models and services, for both runtime and training AI services, targeting (where and when applicable) share and re-use of models across different consumers.

An initial analysis on how to implement the various functions and building blocks of this evolved AIaaS framework has been carried out, with the plan to consolidate and finalize it for D3.4. Specifically, at the moment the analysis considers the integration of existing open source tools, and has identified the following solutions as suitable for the implementation of the AIaaS functionalities on top of distributed Kubernetes execution environments: i) Prefect for the Training Manager, a workflow orchestration tool that allows to automate the execution, scheduling and observability of distributed pipelines with a unified approach; Seldon for the Runtime Manager, a tool that allows to deploy ML models on Kubernetes at scale, packaging them as production REST/GRPC microservices; MLFlow for the AI/ML Catalogue, a tool that supports extensive training tracking and model accessibility through customizable metadata definition; Minio for ML model storage, the high Performance Kubernetes native object storage.

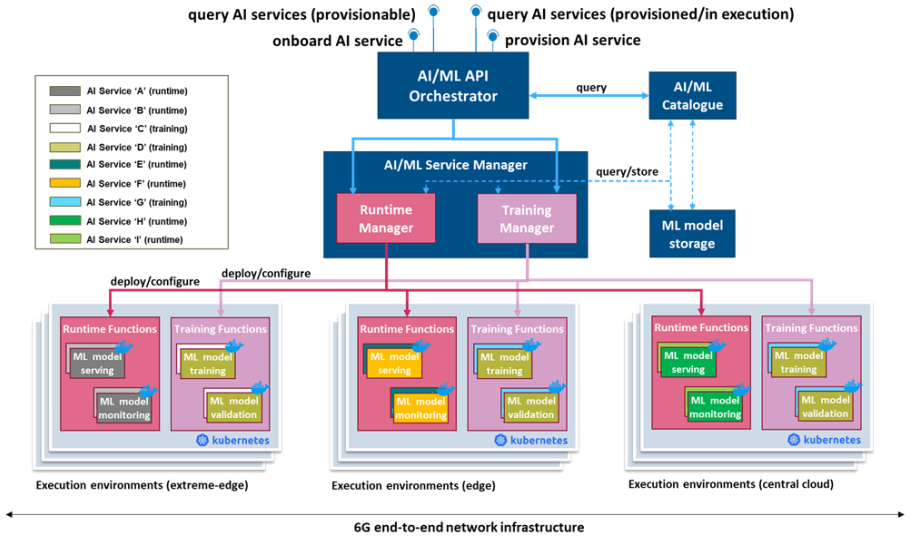


Figure 11-4: AIaaS functional split for distributed AI services

11.1.3 Intent Based Management

Table 11-1 summarizes the intent based management enabler.

Table 11-1 Intent based management enabler summary

In the context of AI enablers, intent-based management enhances the orchestration of various AI components within a data-driven framework. It enables the dynamic allocation of resources, adapts to changing workloads, and optimizes the overall performance of AI-driven processes based on the intended objectives. By leveraging intent-based management, the agility, responsiveness, and intelligence of the data-driven architecture can be enhanced, ultimately leading to more effective and adaptive AI implementations.

KPI improvement	**Measurable KPIs:** automation rate, response time, accuracy of intent interpretation, resource utilization, adaptability and scalability, error rate, cost efficiency, and security and compliance metrics. **Non-measurable KPIs:** innovation and creativity, collaboration and communication, agility and flexibility, user experience, continuous improvement, and environmental sustainability
Design principle improvement	**P1 - Exposure of Capabilities**: Expose advanced data analytics capabilities for informed decision-making, enhancing network features and optimizing 6G services. **P2 - Designed for (Closed Loop) Automation:** Integrate advanced data analytics and ML for intelligent analysis, prediction, and autonomous optimization of network and service operations. **P3 - Flexibility to Different Topologies:** Dynamically adapt to diverse network topologies using data-driven insights, optimizing processes for high performance without manual intervention. **P4 - Scalability:** Implement scalable data processing and analytics to dynamically scale network resources, ensuring efficient utilization across varied deployment sizes. **P5 - Resilience and Availability:** Enhance resilience and availability through predictive maintenance with advanced data analytics, proactively identifying and optimizing resources for increased reliability. **P6 - Exposed Interfaces are Service-Based:** Enhance service-based interfaces with robust data exchange mechanisms, integrating data analytics services for real-time insights and adaptive decision-making. **P7 - Separation of Concerns of NFs:** Emphasize data-driven modularization, enabling seamless integration of network functions with data analytics for independent development and replacement. **P8 - Network Simplification:** Leverage intent-based management principles and ML to automate configurations, streamlining the 6G network for adaptive, efficient design, deployment, and maintenance.
Dependencies / Basis for another enabler	MLOps enhances intent-based management by optimizing the end-to-end ML lifecycle, ensuring seamless integration and deployment. Architectural means and protocols contribute to the interoperability and standardized communication necessary for effective intent interpretation and execution. AIaaS provides a scalable and accessible infrastructure for AI components, aligning with intent-based management 's goal of automated management. DataOps, with its focus on collaborative data management practices, complements intent-based management by ensuring that data-driven insights are efficiently integrated into the decision-making processes governed by intent-based management.
Requirements	A data-driven architecture within an intent-based management perspective requires seamless integration of advanced analytics, providing real-time insights for dynamic decision-making. It should feature scalable data processing, an automation framework for efficient management, and a modular design allowing independent development. The system must be adaptable to diverse network topologies, enhance resilience through predictive maintenance, and employ service-based interfaces for flexibility and reuse.

11.2 Network modularisation

11.2.1 5G Service Based Architecture

From 3GPP Rel-15, 5G networks introduced a key disruption into its architectural design, the Service Based Architecture (SBA), which departed from point-to-point interactions between 3GPP Network Functions (NFs) to embrace standard service Application Programming Interfaces (APIs). This concept allows modular and cloud-native approaches to be adopted into the core network where network functionalities are presented as network services exposed by NFs, with a service framework to support registration, discovery and

authorization of network services, cf. Figure 11-5. 3GPP defined a common control protocol (i.e., HTTP) to implement two communication models for the interactions among NFs through the SBA, namely *(i)* request-response; or *(ii)* subscribe-notify. Communication between NFs can be either direct, using the NRF for discovery purposes, or, starting from Release 16, mediated by the Service Communication Proxy (SCP), which enables centralized signalling monitoring and decouples the discovery and selection procedures.

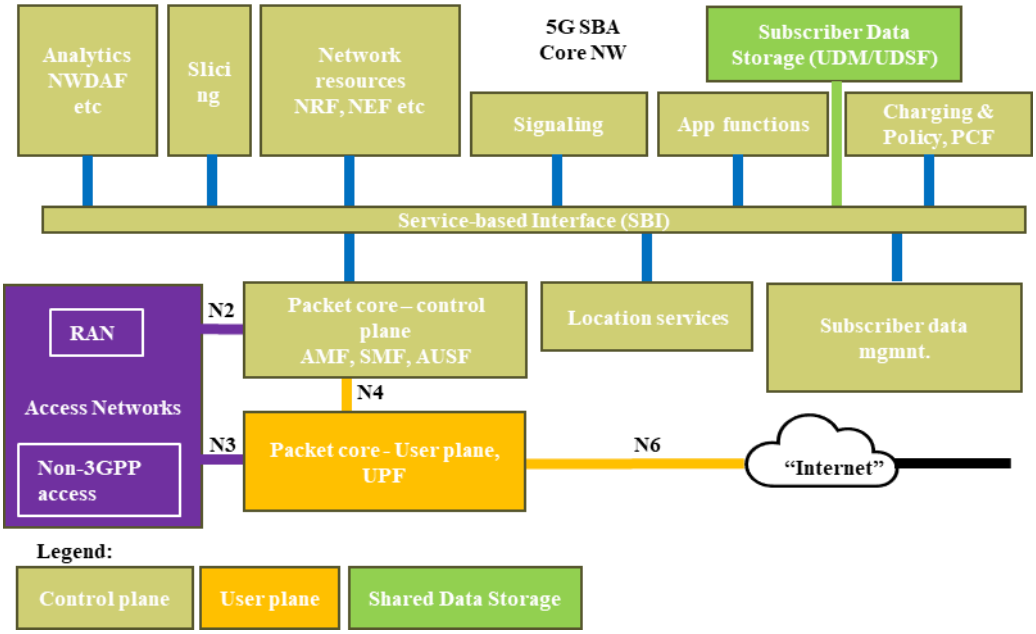


Figure 11-5 5G Architecture for RAN and CN

Although the SBA introduced a giant leap in terms of extensibility and scalability of the network, it remains a host-centric approach, which is not suited for distributed and decentralized NF communication due to the need for mapping function addresses to service names. Improvements in this area, introduced by e.g., SCP, have not demonstrated clear benefits. Moreover, 6G research is also considering the optimization of the SBA model to extend E2E orchestration to the Far-Edge domain, including the adoption of serverless computing. The SBA will then become the communication kernel (also commonly referred to as data fabric) of any E2E 6G system, easing the implementation of self-sustained functions, network scalability, efficient exposure of network capabilities, automation, flexibility to new deployments, and simplification of the architecture. Nevertheless, a streamlined and seamless integration of capabilities and services anywhere in the Cloud-to-Far-Edge continuum (hereinafter, referred solely as continuum) is required to abstract the inherent complexity of a heterogeneous and distributed infrastructure.

To efficiently tackle such an approach in the 6G architecture, the foundation of the underlying SBA must be rethought from scratch without ties to any particular communication architecture.

The SBA of the 5G CN includes usage of a so called SBI between the network functions. With SBA, an authorized NF (within the CN) can access services exposed by another (authorized) NF, as depicted in Figure 11-6. This access is enabled by the Registration, Discovery (and service authorization) functionality of SBA to request a network service. The example in Figure 11-6 considers NRF, AMF and an arbitrary network function NF1.This means that there is no specific signalling like "NG Setup" between any CN NFs in order to establish an interface, they all use the same SBA framework for registration and discovery. Each NF still needs to handle the services it provides (handle incoming request and generate a response) to another NF, and a consumer NF needs to handle request/response of a service it wants to access. Note that the current SBA concept for 5GC relies on text-based messages, requiring parsing to obtain a specific information element.

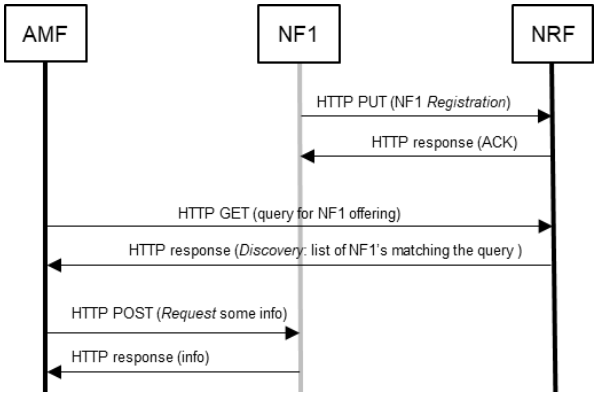


Figure 11-6 Basic SBA functionality Registration, Discovery, and service Request. The current SBA concept for 5GC relies on text-based messages, requiring parsing to obtain a specific information element.

11.2.2 RAN modularity

The following in Table 11-2 are the parameters used to simulate the cell-free architecture and compare it with the different benchmarks in Figure 4-11.

Table 11-2 Cell-free parameters

Parameter	Value	Parameter	Value
Bandwidth	200 MHz	Inter-RU distance	50 m
Transmission power	30 dBm	Min. User-RU distance	5 m
Noise power	-174 dBm/Hz	No. of RUs	900
User noise figure	9 dB	No. of DUs	9
Path-loss	$13.54 + 39.08\log_{10} d$ dB	No. of RU antennas	4
Shadow fading	N(0,42) dB	No. of served users	50
Shadow fading correlation	0.5	Total no. of users	5000
Size of area	2.25 km2	Shift directions	4

11.3 Architectural enablers for new access and flexible topologies

11.3.1 Multi-connectivity

11.3.1.1 Fast addition of a PSCell/SCell during transition from Idle mode to Connected mode

Figure 11-7 illustrates an example of a fast addition of a PSCell when transitioning from Idle mode to Connected mode, where the 5G procedures are used as a baseline example and the proposed modified messages are indicated with green colour. Note that this mechanism can also be used for a fast addition of an SCell.

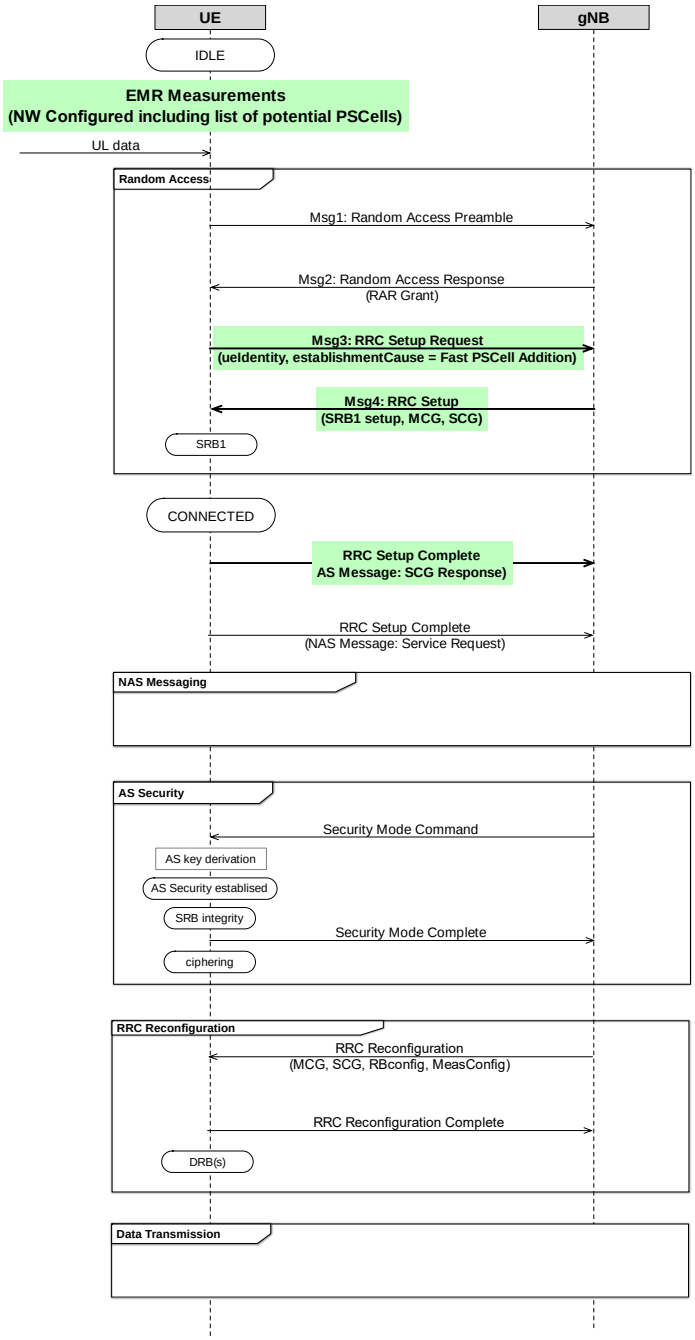


Figure 11-7 Fast PSCell addition from Idle mode to Connected mode

11.3.1.2 Multi-server offloading

As a means of delivering computing and caching services directly from the network edge, MEC has emerged, by the distributed deployment of MEC servers within the RAN [PFH+20]. The MEC servers reside at the BSs and are potentially of different computing capabilities. This reality allows mobile users to utilize the different available RANs nearby and offload computation tasks of their compute-intensive and latency-critical applications to multiple MEC servers simultaneously, as shown in Figure 11-8. In this way, given that the users' computation tasks can be divided into different and independent computation parts, a combination of different MEC servers can be chosen for the processing of each part. Specifically, heavy Machine Learning (ML) tasks, e.g., image processing, can benefit from multi-server MEC offloading. Different video feeds generated from vehicular, healthcare, or security applications - to name a few - can be offloaded to different MEC servers for processing [DZF+20]. In this way, the total task complexity is reduced, and various levels of processing accuracy can be targeted for each part of the task based on each MEC server's computing capability.

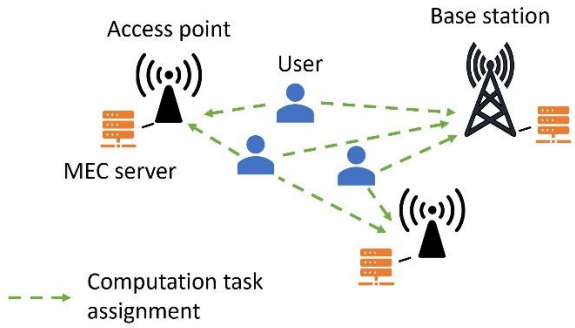


Figure 11-8 High-level overview of multi-server MEC offloading.

11.3.1.3 Dynamic multi-connectivity based on connectivity abstractions

Below, more details about the components of the framework are described.

Functional Domain

An FD, a basic framework block (see Figure 11-9), comprises a set of resources and/or functions that realize a specific goal. A set of UP and CP services is defined for each FD, and such services will have counterparts in other FDs of the same type. The federation of FDs is dynamic, and as a result, broader coverage can be achieved (in case of chaining), a new UP can be added, or MP functionalities can be enhanced.

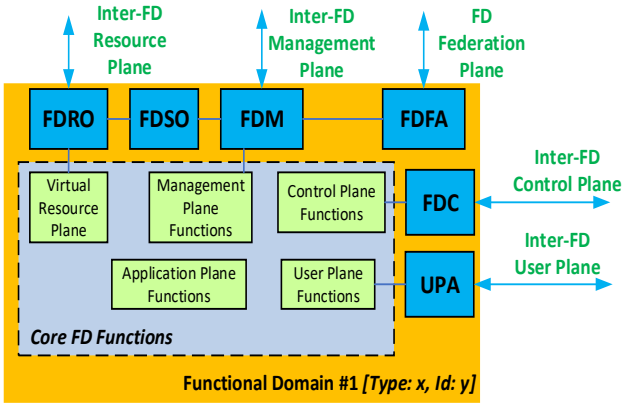


Figure 11-9 A generic Functional Domain structure

The FDs have the following features:

- each FD implements a specific solution that can be described in a well-known, abstracted way (transport, 5G Core, DC, etc.). The FDs can be of the same or different types (access, transport, etc.), of the same or different technology (4G-RAN, NR, WiFi), and have a definition of their serving area. Please note that in the case of NTN, the service area and FD can move. FDs are self-described; however, some FD types can be predefined.
- the identification of services enables an end-to-end federation of several domains on a service level. The CP services, for example, can deal with resource allocation, mobility management, security, etc. The RP resources can be exploited by other system planes. The RP typically consists of virtual resources (connectivity, computing, storage) allocated to virtual functions via orchestration.
- each FD is highly autonomous; the intra-FD operations are AI-driven. The embedded M&O of FD reduces interactions with external M&O platforms. It is a similar approach to [KT18], [KKT+21]. MP is programmable and has an intent-based interface. The FD service orchestrator can be used for intra-FD and inter-FD operations, allowing abstracted orchestration of different technological domains. It has been decomposed into resource and service orchestrators to allow resource aggregation of FD federation.

- external FD interfaces, use abstractions (intents and KPIs) to hide technological differences between FDs and allow uniform operations for different FD federations. Each FD has translators to translate high-level, inter-FD protocol primitives (i.e., intents) into domain-specific atomic protocol operations;
- FD M&O components are involved in the FD federation process; however, it is governed by a federation handling entity called FD Federation Agent. Each FD includes Core Functions, i.e., functions that realize domain-specific functions (cf. LTE, 5GC, set of 802.11 APs, etc.) and functions that support FD integration. The latter functions are the following (cf. Figure 11-10):
- *Functional Domain Controller* (FDC) is mainly used for CP interactions between FDs. It can be seen as a signalling gateway. Its primary role is to translate high-level messages obtained from other FDs into local, FD-specific actions and expose its domain's abstract view and status to other FDs. As mentioned, splitting FDC functionalities according to widely accepted CP services is proposed in the concept. The FDCs of different FDs cooperate to achieve their goal per service level. For example, mobility management entities can be deployed in multiple chained FDs and may cooperate. Mobility management of one FD can also offer its services to other FDs. The functionalities of FDC are programmable using orchestration.
- *Functional Domain Manager* (FDM) is responsible for managing its FD - it plays a role of OSS/BSS of FD. The management is automated and exhibits a high-level management interface. The management services include FCAPS (Fault, Configuration, Accounting, Performance and Security). FDM is implemented as a set of VNFs; therefore, it can be orchestrated. FDM also triggers orchestration requests concerning its domain CP and UP functions. It interacts with the Service and Resource Orchestrators (see description below) for that purpose. It exposes abstracted management information and the management policies/reconfiguration in the form of intents. To achieve its goals efficiently, the FDM should be AI-driven and implement control-loop-based real-time management. FDM functions include self-configuration, self-optimization and self-healing. The usage of FDM in FD integration will be described in the following subsection.
- *User Plane Adapter (UPA)* is an optionally orchestrated entity that can be seen as a UP gateway. Its primary role is UP data adaptation between FDs if necessary.
- *Functional Domain Resource Orchestrator* (FDRO) is an optional entity responsible for the service-agnostic allocation of resources from the FD resource pool. FDRO resources are mainly used for intra-domain operations. Still, they are visible at the federation level and can be used by the federation orchestrator (FDFRO) to handle inter-FD issues.
- *Functional Domain Service Orchestrator* (FDSO) is an optional entity responsible for FD services orchestration. FDM drives the process and interacts with FDRO, which allocates resources for FDSO. The combined functionality of FDM/FDSO and FDRO is similar to the combination of OSS/BSS and the ETSI MANO orchestrator. For some FDs (cf. radio), FD-specific orchestration has to be used. In the case of NTN FDs, the FD-embedded orchestration reduces the amount of data transmitted via a radio link (computing/storage vs. transmission), speeds up orchestration operations and increases orchestration reliability.
- *Functional Domain Federation Agent* (FDFA) is an FD entity that provides the interface to the FP. It is involved in the process of Functional Domains federation. The process is called FDF lifecycle management, and its primary operations are *FDF Creation*, *FDF Update* and *FDF Termination* (see below for more details).

Please note that some FDs, especially the physical infrastructure-based ones, can be permanent; some (NTNs) can move, whereas others can be created (orchestrated) on-demand and added to the existing or new FDF.

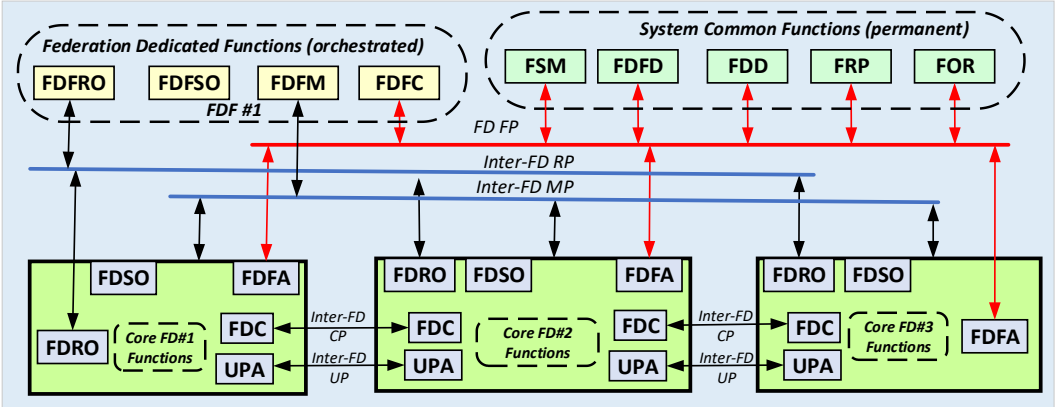


Figure 11-10 Overall framework in case of three FD chained

Federation Dedicated Functions

An FDF needs a set of functions called Federation Dedicated Functions to support inter-FD operations. It is proposed to orchestrate such functions for each FDF. The functions are the following (see Figure 11-10):

- FDF Manager (FDFM) is an entity responsible for managing an FDF. For that purpose, it interacts with FDMs of the respective domain using the Inter-MP message bus and exposes a management interface to the FDF operator. It assumed that FDFM does not provide deep integration of FDMs but interacts with them via intents and KPIs. A deep integration makes FDFM complicated and more troublesome in the case of federation changes – adding or removing FD would involve significant changes in FDFM. For a static FDF, however, the FDFM can be a dedicated, deeply integrated solution. So, the FDM integration level depends on FDF stability.
- FDF Service Orchestrator (FDFSO) – a service orchestrator, which, in cooperation with FDFM and FDFRO (see description below), performs FDF orchestration during the FDF lifetime (please note that the FDFSO can also be orchestrated). The FDFSO is responsible for 'cross FD' orchestration, supports the creation of FDF and may act as an FD orchestrator if a specific FD per se has no orchestration capabilities.
- FDF Resource Orchestrator (FDFRO) is a resource orchestrator attached to FDFSO. It may use resources from all FDs of FDF. It may also enable FDSO of one domain to use FDFRO and resources of other FD. This mechanism supports an immersive federation of FDs.
- FDF Controller (FDFC) is an entity that interacts with FD System Common Functions and FDFAs of FD that form the federation. It uses a Federation Plane (FD FP) for interaction with the mentioned entities. It controls FDF operations, mainly related to adding or removing FDs.

C. FD System Common Functions

- The system's permanent entities are external to FDs and FDFs and play a key role in FDF processes. The functions are responsible for the orchestration of Federation Dedicated Functions for each FDF and providing a mapping between abstracted FDF requests and their implementation. The list of the entities is the following:
- Federation Request Portal (FRP) – is a system that customers use to send high-level requests concerning the FDF lifecycle. The requests concern FDF Creation/Update/Termination operations. The FDF Creation may involve negotiations based on allowed FDF creation options with related costs. In the case of mobile FDs, the decision about their attachment to FDF or removal is taken by FOR.
- Federation Orchestrator (FOR) – is an entity orchestrating FDFs. It has very complex functionalities. First, it responds to FRP requests concerning the FDF lifecycle. Creating an FDF it firstly orchestrates Federation Dedicated Functions, which in turn orchestrates FDs belonging to the federation. During the FDF lifetime, it checks FDD and, if necessary, sends FDFC requests concerning FDF updates, i.e., adding or removing FDs.
- Federated System Manager (FSM) manages the whole system. It interacts with other entities of the Federation Plane to perform system FCAPS.

- FD database is a logically centralized database comprising information about all FDs and their status. Please note that some FDs, especially NTN FDs, may change their service area, and the FD ”mobility pattern” can also be included in the FD database. Each FD registered in FD database obtains a unique ID.
- FDF Database (FDFD) – a database that keeps the information about all existing FDFs and is updated when FDF changes configuration. FDF description includes a list of involved FDs and the interconnection topology. It also may consist of a high-level description of FDF, i.e., the functionality to be provided and the service area, without a specific list of FDs that form FDF.

Outline of framework procedures

This section provides an overview of the framework’s essential procedures related to FDF creation and adding an FD to FDF during FDF runtime.

FDF creation

A generic process of FDF creation can be decomposed into the following steps:

1. each new FD that registers in FD database obtains a unique identifier. The FD database records include synthetic information about FD, type, technology, Core Functions lists, supported CP and UP services, coverage area, mobility pattern (if FD is mobile), already-orchestrated federation framework functions (some of them or all of them can be already added, if the FD in the past was a part of an FDF or the FD was prepared for the framework);
2. the FDF requester (network operator, 3rd party) sends the FDF Creation request in the form of intent (high-level description of ’service needs’) to FRP. The FOR is trying to make a mapping between the request and the hypothetical FDF that can handle the request. If needed, FOR creates (orchestrates) an FD that is necessary for the FDF. It may also add to the existing FDs’ needed functions;
3. FOR creates Federation-Dedicated Functions (FDFRO, FDFSO, FDFM and FDFC) and provides the data needed for FDs interconnection and configuration;
4. FDFM configures FDF, triggers the initial data exchange procedure, and handshakes between FDs are initiated;
5. 5FDFC informs FDFM that a new FDF has been created successfully, and the FDF is registered in FDFD. The FDF enters the running state.

Adding FD to running FDF

The running FDF can be modified by adding or removing one or more FDs. Adding an FD can be triggered by the FDF operator or automatically by FOR if certain conditions are met (a new FD in the FDF service area is discovered, etc.). The process should keep FDF operations continuity and is composed of the following steps:

1. the FDF requester (network operator, 3rd party) sends FDF Update request to FRP or FOR generate such request according to the defined policy;
2. after successful negotiations or FOR decision about adding an FD to a specific FDF, the FOR checks if all functions needed to integrate FDs are part of the FDF and FD. If not, such functions are added by FOR.
3. FOR sends instructions to FDFC, which in turn starts interactions with FDFAs to specify the integration process;
4. duplicated functions of FDF, and new FD are removed with eventual integration of their data;
5. FDFM is updated to interact with the FDM of added FD; item all inter-FD planes interactions are updated to interact with the new FD;
6. FDFC informs FDFM that an FDF has been updated successfully, the updated FDF configuration is registered in FDFD and updated FDs in FD database.

The presented concept does not claim to be a complete solution; it is instead a starting point for discussing the vision of the architecture of future heterogeneous networking solutions with dynamic topologies. Future research should provide more details of inter-FD interactions, FD information models and verification of the concept in a simple use case.

11.3.2 E2E context awareness management

11.3.2.1 Transport network abstraction

This section provides details on the architectural implications of transport network abstraction analysed and discussed in Section 5.3.2.1 of the main document. Specifically, details on the abstraction creation and update procedure are outlined in the following, along with a practical illustrative example to enhance understanding.

First, an initial, global, static abstraction of transport resources is performed at the beginning of network operations. The global periodic abstraction allows for handling cases with limited knowledge about the expected traffic, resulting in an initial abstraction based on a restricted understanding of the future traffic to be served. Nonetheless, even in cases where there is a complete mismatch between the initial abstraction and the actual traffic behaviour due to the unknown distribution of traffic in time and space, it is important to note that the information regarding the characteristics of traffic (e.g., bandwidth, latency, etc.) is still known, particularly for critical traffic with characteristics stipulated in specific Service Level Agreements (SLA). This knowledge of traffic characteristics is sufficient to create suitable initial baskets that can be updated on an ongoing basis through the global periodic abstraction process.

When a vertical client requests a specific service, such as a URLLC link for robots' connectivity, the E2E Orchestrator maps the required service onto a slice that is suited for the request. The orchestrator then selects the appropriate Physical Network Functions (PNF), Virtual Network Functions (VNF), and Cloud Native Network Functions (CNF), along with their corresponding transport connections in the abstract view to ensure the QoS of the slice. Subsequently, it configures the PNF and places the VNF/CNF with the related transport connections.

The basket of abstracted resources is then dynamically updated based on the current availability of resources. According to specific rules or policies, the basket update can trigger a modification of the abstraction view. Asynchronously with respect to traffic requests and basket updates, a parallel process runs within the transport network to determine if a basket update is necessary. For instance, baskets containing more paths than needed can be partially emptied or removed to create new baskets between source-destination transport nodes that better align with the actual traffic distribution.

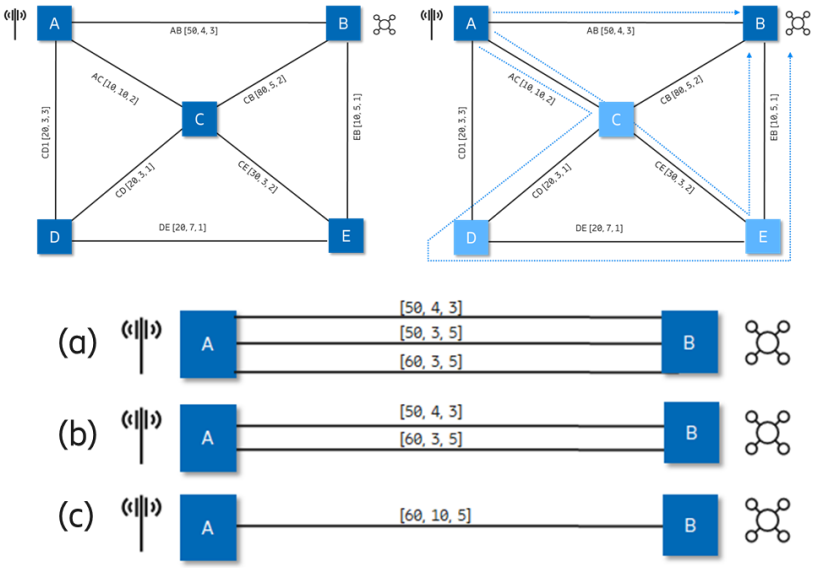


Figure 11-11 Abstraction of a five nodes transport network

To clarify the abstraction of transport resources, a simple example is illustrated in Figure 11-11 and described below. This network consists of five nodes. Each link extending from node X to node Y is characterized by three parameters: a cost parameter (C), the link's supported bandwidth (throughput, B), and the maximum latency (L). The definition of "cost" is beyond the scope of this example. The cost (C) and latency (L) parameters are considered cumulative across links, while the band width (B) of a sequence of links is equal to the minimum bandwidth among those links. Let us examine the connectivity from nodes A to B, which represent edge nodes of the physical mesh.

A path computation engine determines K physical paths between points A and B. Assuming K = 3, the paths are as follows: (1) AB [50, 4, 3] directly via AB, (2) AB [50, 3, 5] via AC-CE-EB, and (3) AB [60, 3, 5] via AC-CD-DE-EB. These three physical paths can be advertised to the higher layers as one or more groups, called baskets, of possible virtual links. Each group is characterized by specific costs, bandwidths, and latencies. There are several alternatives for aggregating the three physical paths into virtual links, as illustrated in Figure 11-11:

(a) The abstract view discloses all the paths.
(b) Paths (1) and (2), which share the same bandwidth and latency, are revealed as a single path, while path (3) remains explicitly exposed.
(c) The abstract view presents a single path with bandwidth equal to the sum of the bandwidths for each physical path and the highest possible latency value.

This third case of abstraction is applicable only in certain transport technologies that allow for bandwidth link aggregation.

The previous example can also be applied when the transport is a composition of multi-domain networks, as illustrated in Figure 11-12. In principle, it is possible to replicate the same abstraction examined in each domain and expose a combination of abstract views for each of them. In such case, if the multi-domain network involves different network operators, it is important to establish common rules that each operator adopts for their respective domain.

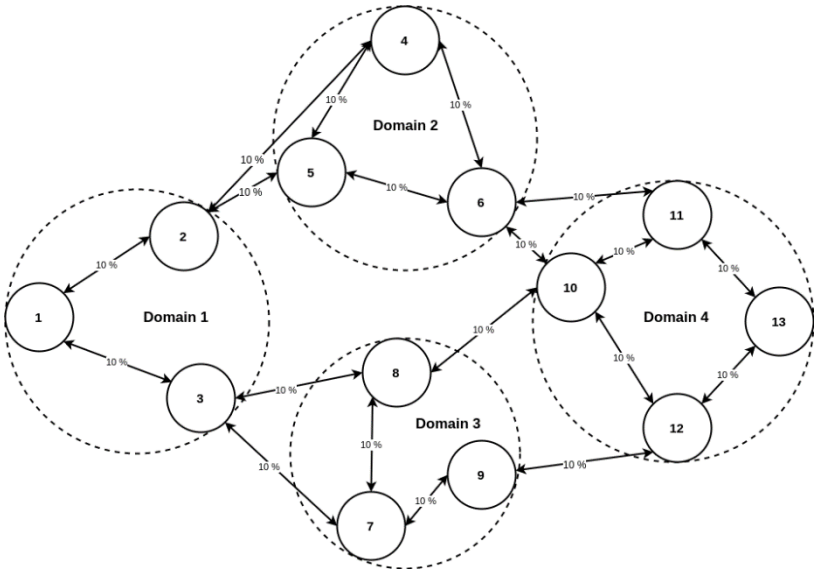


Figure 11-12 Example graph showing inter-domain and intra-domain connections

Regardless of whether the network is single or multi-domain, a major open issue is the standardization of the API that allows correct interworking between the transport and upper layers, such as RAN and Core network Domains.

11.3.2.2 Task allocation in Semantic RAN

In this section, the Semantic RAN concept initially introduced in Section 5.3.2.2 is further presented under the ETSI MEC architecture. Also, a special case devoted to the harmonization of the ETSI MEC architecture with the 3GPP-5G architecture is further enclosed.

First, the ETSI MEC Standard Architecture is studied, and the different components integrated within this architecture are presented in Figure 11-13. Specifically, ETSI MEC enables to integrate the SDLA and SESM as new functional blocks inside the MEO. With the integration of the SDLA, the MEO can obtain the task requirements and calculate the task latency and accuracy functions. The SESM, with this calculated latency and accuracy functions, processes the requirements of the task and calculates the optimal offloading policy, the compression level, and the computation and radio slicing.

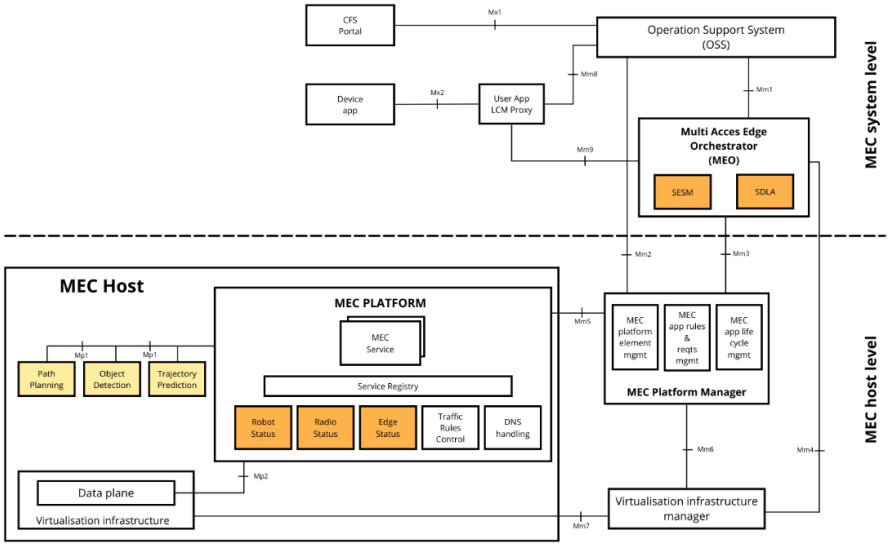


Figure 11-13 Integration of the different modules inside the ETSI MEC Architecture

Specifically, the following operations take place in ETSI MEC:

1. **Petition for an instantiation of a robot task**. The corresponding petition comprises a task descriptor of the robot task, including the deep learning model to be executed and its execution requirements, such as latency and accuracy. This petition runs from the CFS Portal to the OSS through Mx1 interface, and then to the MEO using the Mm1 interface.
2. **Semantic Analysis of the task**. The SDLA computes the latency and accuracy functions with the information gathered from the radio status services and the task descriptor. These functions are then shared with the SESM.
3. **Semantic Edge Slicing of the task**. The SESM determines how to accomplish the robot task by choosing between all available policies and identifying the necessary slicing requirements for radio and compute resources. For this purpose, the SESM uses as inputs the latency, accuracy, the requirements of the tasks, and the metrics gathered from the Radio, Edge, and Robot status services.
4. **Task instantiation as MEC Apps**: The SESM communicates the necessary slicing requirements to the MEO. The MEO starts the process of instantiation of MEC Apps. This module communicates with the Virtualisation infrastructure manager, through Mm4, which is responsible for preparing the virtualisation infrastructure to run a software image.
5. **Resources and Slices allocation**. With the corresponding resource slicing computed in the SESM, the instantiation of the MEC Apps can be fulfilled. For the radio slicing part, the interested reader can refer to [ETSI36].

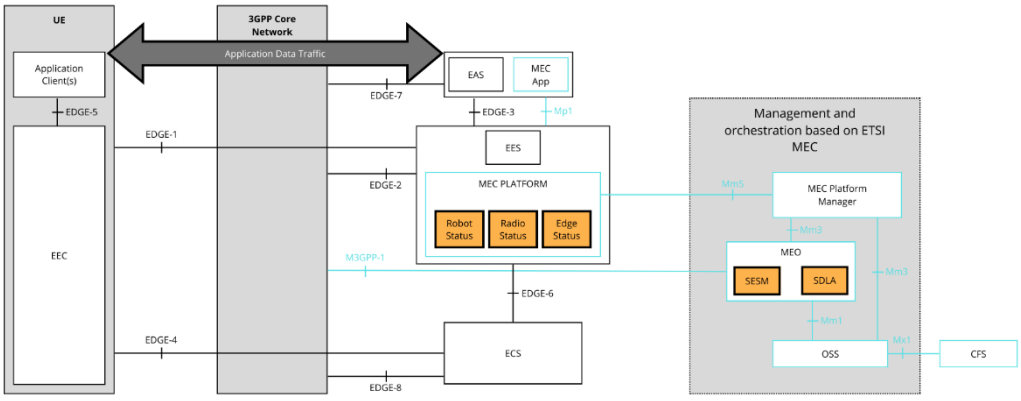


Figure 11-14 Harmonizing 3GPP with the new modules inside ETSI MEC Architecture

Next, the harmonization of ETSI MEC and 3GPP architectures is discussed, as shown in Figure 11-14. In this case, the SDLA is necessary to interact with the Control Plane of the 3GPP Architecture. The interoperability

with the Control Plane allows a close interaction with the AMF and the SMF. The SEMS focuses on managing network slicing at the edge within the 3GPP architecture. The interaction of the SEMS is needed at the User Plane, working towards with the UPF. The integration enables a semantic understanding to manage in an efficient way the edge resources, ensuring that the slicing at the edge is in line with the semantic needs of the mobile robot's applications.

Although there is an interface that connects directly the MEO with the 3GPP Core Network (M3GPP-1), this interface covers only the Service Capability Exposure Function (NEF). Extra information about how this process could be made can be found at ETSI GS MEC 012 [MEC012]. The RNIS offers data pertaining to radio networks to both MEC applications and the platforms they operate on. The service consumers communicate with the Radio Network Information Service over RNI API to get contextual information from the radio access network. With this proposal, radio status context can be added to the MEO.

11.3.2.3 Multi-domain SDN scalability evaluation

This section provides details on the operation complexity of several path computation algorithms that is related to the evaluation of multi-domain SDN scalability in Section 5.3.3.1 of the main document. The respective algorithmic complexities are presented in Table 11-3 and Figure 11-15.

Table 11-3 Computational complexity of different path computation algorithms

Path Creation Algorithm	Algorithmic delay
Dijkstra	$O(V^2)$
Dijkstra (binary heap)	$O((E+V)*log(V))$
Parallel Dijkstra	$O\left(\frac{V^2}{P}+V\log V\right)$
multi-SDN (simple case, two-level Djikstra)	$O\left(x\frac{V^2}{N}+(x+1)^2N^2\right)$

where: E – edges, V – vertices, P – number of processors, x – number of GCL requests to establish a path in a domain, N – number of domains

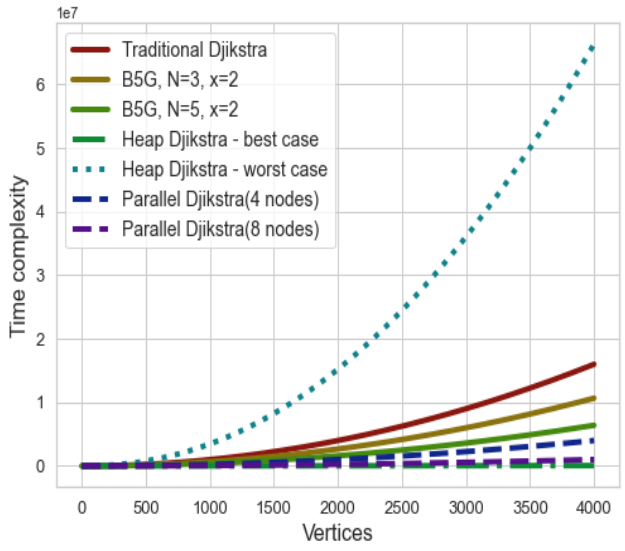


Figure 11-15 Algorithmic (time) complexity dependence on the number of network vertices.

11.3.2.4 Autonomous Robots

In this section, the technical details of the experimental validation of the autonomous robots use-case scenario in Section 5.3.3.2 are listed in the following two tables. Considering the two models, the COCO dataset [LMB+15] was used for the detection of the classes.

Table 11-4 Object detection models details

Model	Size	mAPval (0.5:0.95)	Params (M)	FLOPs (G)
YOLOX-s	640x640	40.5	9.0	26.8
YOLOX-tiny	416x416	32.8	5.06	6.45

Table 11-5 Hardware setup details

Computer	Dedicated GPU	CPU	RAM	Role
Alienware m16 R1	Nvidia GeForce RTX 4060	AMD Ryzen 7 7745HX	32 GB	Server
MacBook Pro-2017	-	Intel Core i7-5557U 3.10GHz x4	16 GB	Robot

11.4 Decentralised compute-continuum smart management

As already introduced in the previous Deliverable D3.2 [HEX223-D32] the decentralised compute-continuum smart management concept targets the new challenge regarding the integration of the compute-continuum infrastructure resources for the network services M&O, and with focus on integrating the so-called extreme-edge domain [HEX22-D62] due to its key challenging features: the aggregation of devices beyond the MNO own premises, the diversity of stakeholders in this domain, the high heterogeneity of devices, the potential high volatility of those devices (which could unexpectedly move or even be disconnected), and the size of this domain, that can be potentially huge.

The decentralised compute-continuum management concept proposes an alternative to the regular MNO-centric M&O approach that was already considered for the 5G networks [KNE16], and also for some initial PoCs targeting the future 6G networks [HEX23-D63], consisting in delegating the complete M&O functionalities to a hierarchy of orchestrators in a multi-domain approach with several specific orchestrators for the different domains (e.g., cloud, edge and extreme-edge domains), which in turn are orchestrated by a common E2E centralised orchestrator, and towards 6G, considering the extreme-edge just as an additional network domain.

This decentralised compute-continuum management concept considers that this legacy 5G-like approach could pose limitations to integrate the extreme-edge domain in full-scale deployments, e.g.:

- Regarding the scalability of the solution. The extreme-edge specific features can lead to a high level of complexity in the administration and configuration of a large number of resources and services, which could not be easily scalable using the legacy 5G MNO-centric approach. Just as an example, the gathering and processing of monitoring and diagnostics data from the huge amount and diversity of devices in the extreme edge could be a significant challenge.

- Regarding the capacity planning. It has to be considered that the extreme-edge refers those resources beyond the MNO own resources, i.e., it would be necessary to orchestrate the network services on a hybrid resource pool, in which there would be resources belonging to the operator, but also, other resources belonging to other stakeholders (e.g., devices in vertical industries, smart cities, hyperscalers or even end-users).
- The communication among orchestrators from different stakeholders may be not a suitable option. I.e., in some complex use cases of multi-domain service orchestration based on the 5G technology it is assumed that there could be a straight-forward communication among the different orchestrators of the different operators in charge to orchestrate common services for all of them [NCV+23][BVB+22] (this is the concept known as "federated orchestration"). However, this is a problem that is not easy to be solved in practice, since the different operators could have very different technological solutions for orchestrating their own services, and that communication between orchestrators could not be so straight-forward or effective enough for the specific common services to be deployed (and this beyond the business implications that a scenario like this could pose).
- Orchestrating the orchestrators in the different domains may add a new level of complexity. Just adding a top-level centralised E2E orchestrator as described above may not be suitable enough to perform an effective E2E orchestration integrating an environment as large, diverse, and volatile as the extreme edge. E.g., just propagating towards the higher level of the orchestration hierarchy the low-level infrastructure details, which for certain services might be useful or even necessary, could be not feasible or straight forward.
- Regarding the orchestration resources optimization. The network services to be orchestrated may be very different in scale and complexity, so not always requiring the same kind of orchestration resources. A common MNO-centric orchestration framework would provide the same orchestration resources to all network services, which could mean an oversizing of the orchestration resources over those that would really be necessary.
- A common MNO-centric M&O framework could discourage external parties (e.g., vertical industries or SW vendors) relying on different technological solutions to integrate their solutions in the network. I.e., beyond the business aspects, that common M&O framework could represent a technological barrier for external parties that probably are used or prefer to work with different technological approaches.
- As in all the centralised approaches, the centralized M&O solution could be a single point of failure.

Considering this, and as already anticipated in [HEX223-D32], the decentralised compute-continuum smart management concept proposes a different approach towards the M&O of services in the future 6G Networks, which is considered to be more flexible and practical in accordance with that high heterogeneity, size, and dynamicity of the extreme-edge domain. As a whole, this new approach consists in delegating the service assurance M&O mechanisms on the network services themselves and relying on a set of decentralised network elements the services provisioning operations, thus providing a more autonomic and decentralised approach. This approach also relies on designing network services in a cloud native way, i.e., integrating service components in the form of micro-services, which, in turn, can include specific micro-services to implement the specific service assurance M&O mechanisms that would be required for each service. It is considered that this approach provides the following broad benefits:

- There would be a better orchestration resources optimization, which would be tailored to each network service specific needs. Each service would be designed containing only its own specific M&O resources depending on its specific context. It is well-known that certain services may not require the same level of complexity in the orchestration than others (e.g., in what regards data monitoring, use of AI/ML techniques, etc.), being the case, that certain services may require even very simple orchestration primitives.
- A better scalability. Most of the services M&O mechanisms are envisaged to be distributed, per-service, so inherently scalable. The system is designed to support a large number of resources and services.
- The approach allows integrating non-owned resources in a multi-stakeholder environment. The network services would be in fact an aggregation of multiple micro-services, which would be provided by different stakeholders, and that would interact through their exposed APIs in the regular cloud-native way.

- The approach does not require overall technical or business agreements to interconnect complex MNO-centric orchestration frameworks. In "federated-like" scenarios the required communication would be simplified, being addressed only at service level, and only for specific service components.
- There would be a lower technological barrier to external parties (e.g., vertical industries or SW vendors) to integrate and share their specific technological solutions in the network.
- The whole system would be much more fault tolerant (single points of failure are minimised). Being decentralised this approach can be more resilient to failures, as a problem in one specific service would not necessarily affect other services.
- The need to align services with the specific requirements and functionalities of a common MNO-centric orchestrator (which may become obsolete) is minor, since the decentralised approach might be more effective in fostering ongoing innovations, as concentrating control within a single entity could hinder new concepts and developments.
- Higher adaptability: decentralised orchestration can more easily adapt to changes in network topology or unforeseen conditions, which is especially relevant regarding the extreme-edge integration, as services can make autonomous orchestration decisions in real time without relying heavily on a central point.
- All the previous could lead to less operational costs: for the MNO, externalizing orchestration resources to the network services reduces the complexity in its own infrastructure. Network services could be managed by different stakeholders, and certain network services could be managed even without the need of the MNOs to be involved.

As anticipated in [HEX223-D32], the technological approach to address this decentralised compute-continuum smart management concept consist of four so-called "distributed network stakeholder support services" (DNSSS), namely:

- The Deployment Service (DS).
- The Infrastructure Registry Service (IRS).
- The Services Registry Service (SRS).
- The Infrastructure Status Prediction Service (ISPS).

These four stakeholder support services are responsible for making possible the provisioning of new network services in the network, and would be implemented by four types of specific nodes associated to each service: deployment nodes, Infrastructure Registry Nodes (IRN), Service Registry Nodes (SRN), and Infrastructure Status Prediction Nodes (ISPN), which in turn would be distributed throughout the network hosted by different stakeholders (see Figure 11-16). As it can be seen, these nodes would be allocated in the facilities of different stakeholders with access to the network, and with the resources and capacity to host and manage these network elements. The number of instances deployed for each of these nodes would be variable, depending on the size and requirements of the network.

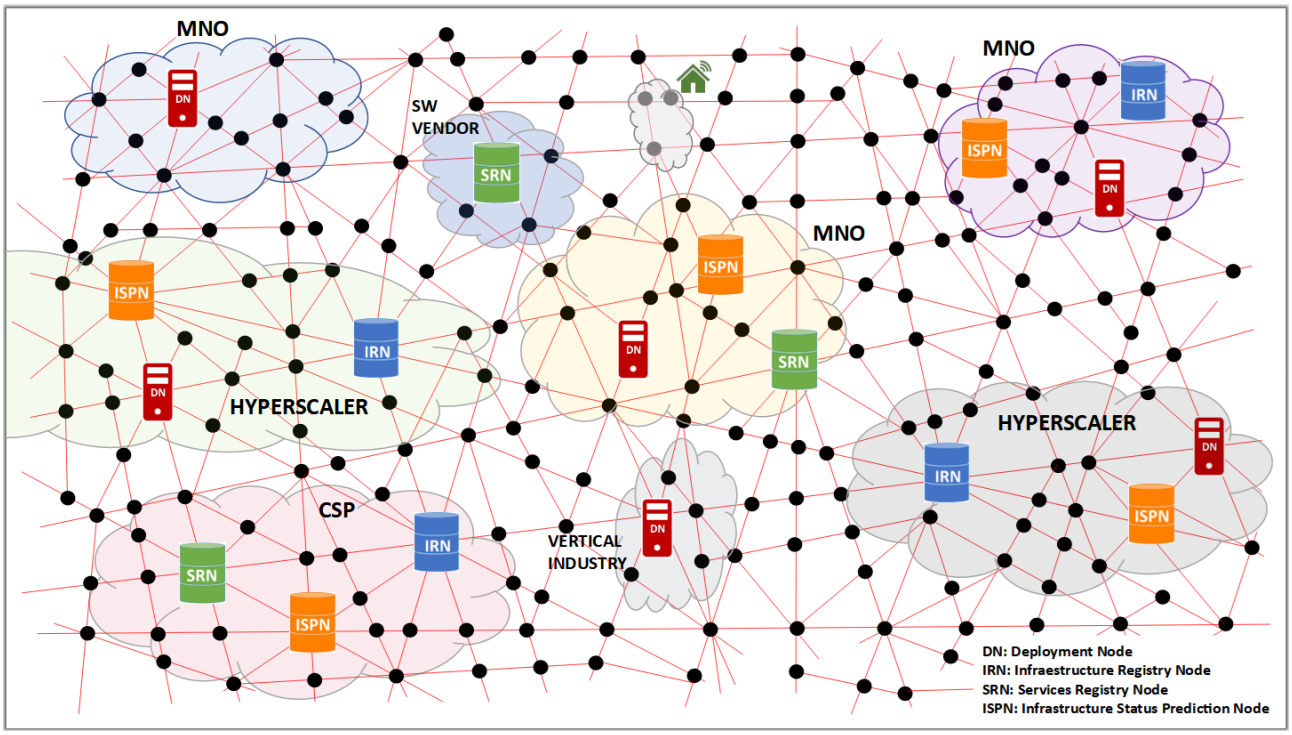


Figure 11-16. Network with different stakeholders (clouds) and the four Stakeholder Support Service Nodes distributed throughout it.

The previous Deliverable D3.2 [HEX223-D32] already introduced the basic interactions among these nodes (see Figure 11-17): as it can be appreciated, the overall process distinguishes two main stages: (i) the network services provisioning stage, which is delegated on these four kind of nodes in Figure 11-16, and finishes by providing a "service handler" to the stakeholder that was requesting the service provisioning, and (ii), the post-provisioning stage, where whatever other subsequent M&O operations are delegated to per-service M&O mechanisms, that would be embedded on the deployed network services themselves, and would be designed tailor-made according to each service specific needs. Different technologies for implementing those per-service M&O mechanisms could be considered, such as state-of-the-art containers M&O solutions (e.g., [K8S], [K3S], [SWARM], [NOMAD]), ad-hoc orchestration systems specifically developed for the services (e.g., in case a certain stakeholder already could provide that), or even through microservices choreographies (i.e., without relying on any specific orchestration component or framework) [CDT18].

Beyond the initial information provided in the previous [HEX223-D32] more details are provided below for each of the above-mentioned distributed stakeholder support services, focusing on more functional aspects.

Deployment Service

The DS works as the entry point to the network for deploying network services, which could be done using a declarative intent-based approach, i.e., by specifying just the desired final result regarding the deployment, and without needing to specify how to achieve that result. For each service to be deployed, the stakeholder (or a group of stakeholders) requiring the deployment would select the most appropriate deployment node to deploy the service, e.g., based on geographical criteria, specific technical requirements, agreements with the deployment node hosting stakeholder, business terms and conditions, etc.

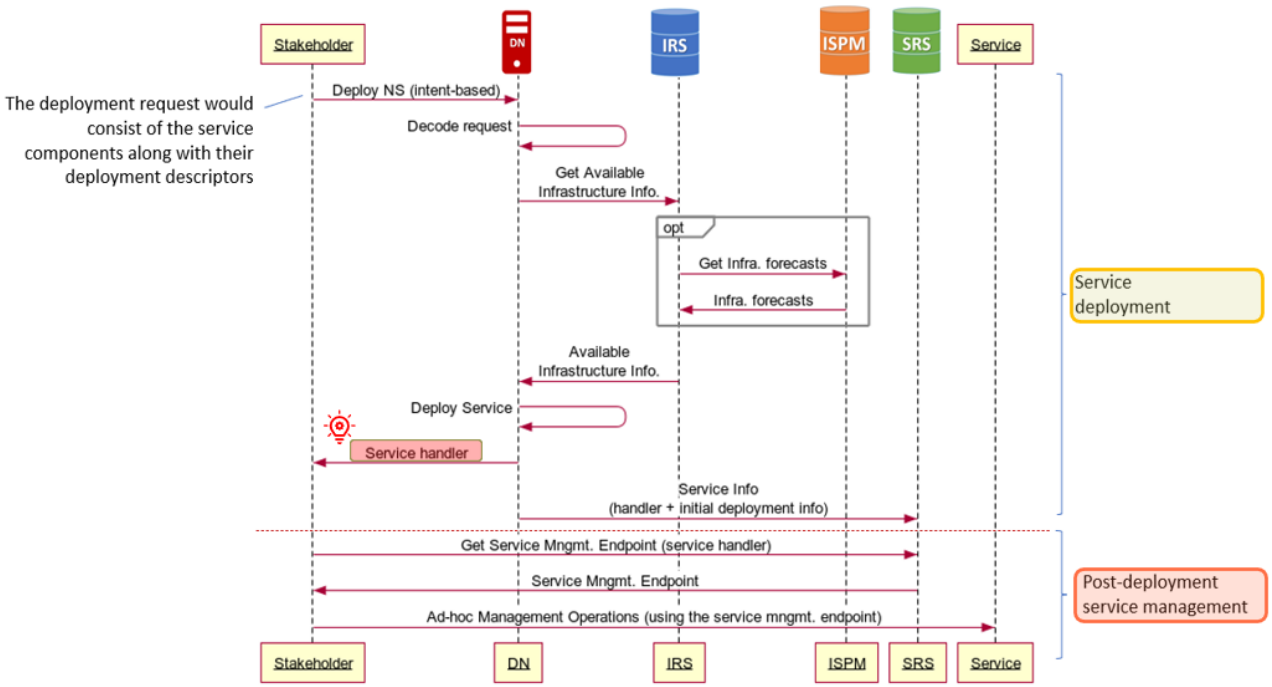


Figure 11-17. Decentralised Orchestration. General operative sequence diagram [HEX223-D32].

As already explained in [HEX223-D32], once deployed, the DN would provide to the stakeholder requesting the deployment with a so-called "service handler", which from that time can be used each time the user needs to access the service to perform any operation on it. This "service handler" must be considered as an "access key" to reach the service once it has been deployed. In case the service could not be deployed for any reason, instead of that "service handler", the user will receive a "deployment error" message explaining the cause of the failure.

In line with the decentralised compute-continuum smart management overall focus, network services to be deployed need to be designed following a cloud-native micro-services approach, i.e., they must consist of one or more small self-contained micro-services able to communicate over well-defined exposed APIs. As explained, the services to be deployed are expected also to include the specific M&O functionalities that could be considered necessary for the service assurance (e.g., to access relevant service metrics, to perform service components scaling or migration actions, etc.), although this is considered optional and at the discretion of the network service designers.

As a high-level approach, Figure 11-18 showcases how the deployment of a network service composed of different microservices (from µS1 to µSn) would be defined. In line with what we have just explained, one or more of these microservices could optionally contain specific resources dedicated to the orchestration of the service as a whole. As it can be seen, the way this service should be deployed on the network can be defined relying in a declarative intent-based approach, using the deployment descriptors of each of the microservices for that, i.e., for each microservice, the deployment descriptor defines the target features of the infrastructure components on which these microservices are required to be deployed, including a list of parameters that define the features of the required infrastructure nodes, such as the number and type of CPUs, available RAM, IP addresses, the networks to which the device should be connected, the stakeholder to which the node belongs (e.g., an MNO, a hyperscaler, an industry, etc.), the network domain to which it is associated (cloud, edge, or extreme-edge), geographical information, etc.

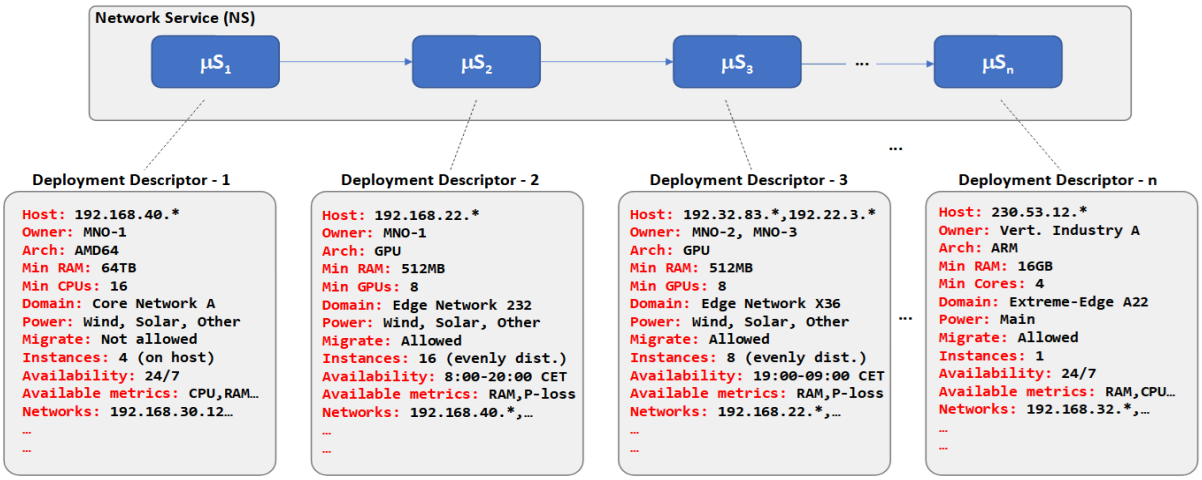


Figure 11-18. Example of intent-driven deployment of a Network Service [HEX223-D22].

To define these parameters the user will have a data model available, which will allow to deploy the different micro-services on different kinds of target nodes in the infrastructure, i.e., with the required computing architectures, network domains, kind of power supplies, etc. The particular data model will be provided to the service designers by the stakeholder hosting the deployment node. Although in general it could be expected that all data models of all deployment nodes in the network will be largely aligned, it could also happen that some stakeholders could offer certain data models with more configuration granularity than others, which should be also a consideration that may condition the decision to use some deployment nodes in favour of others.

Of course, although there would be a mechanism in charge of processing the high-level declarations in the descriptors to find and select the most suitable nodes in the infrastructure for the deployment of each microservice, this would be hidden for the stakeholder deploying the service. This means that, in general, the user would not have to care about defining on which specific nodes of the infrastructure the service components should be deployed, as this will be done automatically by the deployment service. So, the user could focus on defining just certain high-level features that would be considered necessary for each microservice (e.g., preferred geographical area for the microservice to be deployed, preferred computing capabilities, etc.). However, if necessary, the user could also determine some of these features in a very specific and univocal way, as represented in Figure 11-18 (e.g., to deploy certain microservice on a node with a specific IP address).

However, the stakeholder deploying the service must be aware that those automatic processes just mentioned above would work only for the initial deployment of the service. As also mentioned above, after the initial deployment performed from the DS, the specific service assurance M&O functionalities would be delegated on the deployed network service itself. The service designer should take this into account especially when the deployed services need to be total or partially deployed on the extreme-edge domain, since the infrastructure resources at that domain could be highly volatile (the infrastructure resources could unexpectedly connect/disconnect, move, or drastically change their available resources). So, the user should consider the appropriate service orchestration assurance mechanisms as part of the service, which should be continuously executed to keep the deployment consistent with the declarations in the deployment descriptors by, e.g., re-deploying or migrating the affected network service components in case the node that was initially selected for deployment becomes partial or unavailable.

For interacting with the Deployment Service, the deploying stakeholder would be able to upload the service components to be deployed into the network using a specific user interface, provided by each deployment node. This could be a graphical user interface, a commands-line interface, an intent-based interface, or a combination of them. Figure 11-19 shows a schematic representation of a kind of GUI that a deploying stakeholder may encounter in a deployment node. On top it can be seen that, besides aggregating the user's own components, the interface also allows accessing components from other stakeholders (different MNOs, Vertical Industries, etc). The lower part would be used to compose and/or visualize the network service graph as a whole, as well as to inspect and/or edit the deployment descriptors of each component. On the right-hand side we can see also a natural language intent-based interface, which would be used to define the deployment

descriptors or to get information from the available resources, among others. Of course, is not expected that all the deployment nodes would offer the same level of complexity as this representation in Figure 11-19. Anyway, even in the simplest case, the deployment node would always provide the deploying stakeholder with the basic functionalities to make the deployment feasible, and to provide the service handler once the service has been properly deployed.

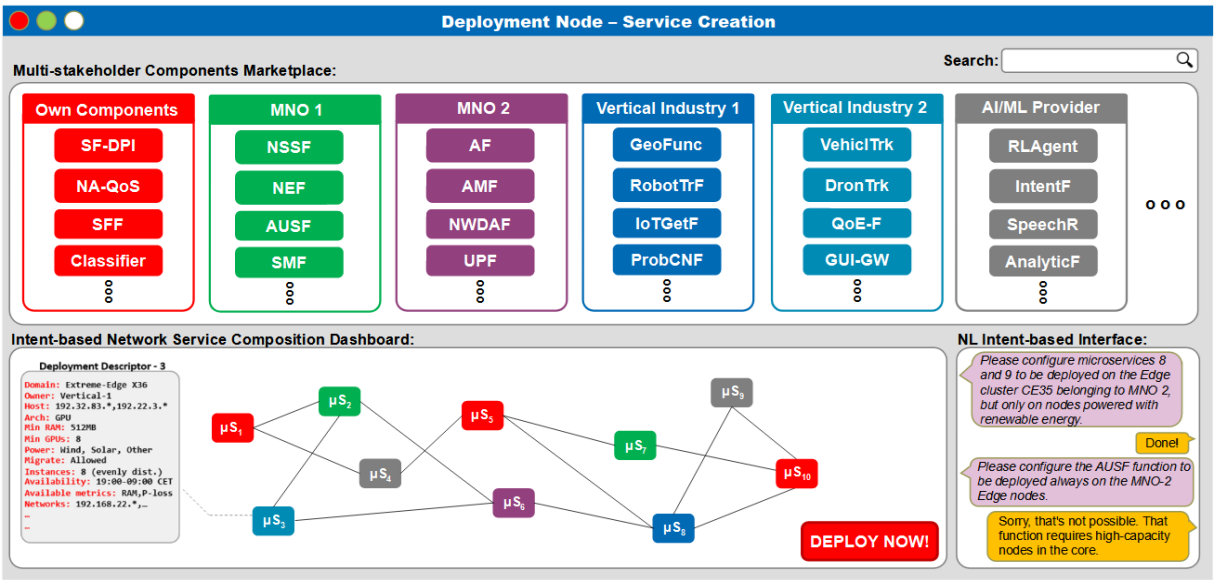


Figure 11-19. Deployment Node GUI example.

Infrastructure Registry Service.

The IRS is the core of the distributed provisioning system. It is intended to store and provide updated information about the available infrastructure devices registered in the network (for both, physical and virtual devices), as well as their most relevant features regarding their capabilities to host the service components (microservices) that could be requested to be deployed from the Deployment Service. Its main role is to reflect as closely as possible all the infrastructure actually available at any given time (it should be borne in mind that at the extreme-edge domain unexpected changes in the infrastructure may occur with some frequency). According to this, the IRS main functions are to register new infrastructure devices (based on a devices discovery mechanism), to update relevant information regarding already onboarded devices, and to delete those devices that for some reason should be taken out of service.

In general, we will assume that the devices to be attached to the network would be enabled with a kind of "hot plugin" mechanism, i.e., devices are assumed to be able to automatically "self-register" by themselves in the IRS when connected, as well as to automatically notify certain updates regarding their status (e.g., regarding their configuration, placement, or availability status). In this case the information of the device would be reachable from the IRS using specific IRS-device interfaces. If that “hot plugin” functionality were not available for certain devices, the infrastructure provider (or the infrastructure integrator) should implement the necessary mechanisms to make the device reachable and/or perform a sort of manual registration of the device in the IRS. Again, the specific IRS-device interfaces would be used for this.

As a utility function, the IRS would provide the stakeholders with an interface to allow verifying those devices registered in the network, as well as to query the relevant device type and parameters. As for the DS, this IRS user interface could vary in complexity and provided functionalities, depending on the technology used for its implementation (e.g., it could be a simple text-based interface, a GUI, or a more complex natural language based interface). The access to this IRS interface could be available through specific IRS end-points, but also, through certain Deployment Nodes, which in this case would act as a “bridge” between the user and the IRS (some deployment nodes could be enabled with this bridge to ease their users with a single administrative access point).

Infrastructure Status Prediction Service

The ISPS complements the IRS generating predictions regarding the infrastructure devices, and specifically, in what regards to their absolute availability (i.e., connection/disconnection status), or also, regarding the availability of certain volatile resources provided from each device (e.g., battery, memory, or computing resources). These predictions are generated using data analysis algorithms (which may also include AI/ML algorithms) applied to the historical devices' behavioural data, which in turn could also be correlated with other data external to the network itself (e.g., data plane information, users' information, etc). So, the ISPS can be seen as an overall network service able to combine information from multiple network devices, different network domains, and even heterogeneous data coming from outside the network, to be able to make accurate predictions correlating a big amount of data. As with the other stakeholder support services, the ISPS may consist of one or more Infrastructure Status Prediction Nodes distributed throughout the network, which could gather and process data from, e.g., specific network domains or different geographical areas.

The main purpose of the ISPS is to help dealing with the extreme-edge dynamicity and volatility: based on their predictions, (i) the DS would more optimally select the infrastructure resources to deploy new network services (e.g., avoiding those nodes envisaged to be disconnected in a short span of time), and (ii), the network services already deployed could also take more proactive orchestration decisions based on the forecasts provided from the ISPM (e.g., a service component could be migrated if the device on which it is deployed is forecasted to have insufficient memory or computational resources).

For the first case (i.e., the deployment of new services) the deployment process would work fully automatic, based on the interactions among DS, IRS and ISPS: the DS will automatically select the most optimal set of resources according to the information received from the IRS and the ISPS. However, for the second case, if the user wanted the deployed service to be enriched with the ISPS forecasts, it would be necessary to subscribe the service to the ISPS forecasting events. In practice, these events could be generated directly from the devices where the service is deployed (e.g., a given device could send an event to the service saying something like "warning: device envisaged to be switched off in 10 minutes"), or also, from specific ISPS dedicated nodes (e.g., "warning: node A envisaged to be available again in half an hour"). For the first option (device notifying) the ISPS would write the corresponding information directly on each device according to specific data models.

In order to receive this information from the devices or the specific ISPS nodes the service development team would need to design the network service including events handlers to receive those ISPS events and to take actions accordingly (e.g., regarding the device connectivity status, memory or CPU levels thresholds, etc.).

Although most of the processing work regarding the ISPS is assumed to be fully automatic (i.e., the gathering of data from the network, the application of the data analytic algorithms, etc.), as with the other stakeholder support services, the different nodes in the ISPS could offer also specific user interfaces, which could also take different forms and shapes (from text-based user interfaces to more complex GUIs). These interfaces would provide information about the supervised devices in each ISPN, the available events per device, etc. They would also provide facilities for querying, and for the manual registration or deregistration of certain infrastructure devices to be monitored (or not) by the ISPS as a whole, or by certain specific ISPNs.

Services Registry Service

Since, beyond the provisioning, the subsequent services M&O operations are delegated in the network services themselves, the network service designers are expected to enable certain access points in the services themselves to make it possible these M&O operations. However, the network must grant these service M&O access points are always available, which, considering the high volatility of the extreme-edge devices, would in principle not be guaranteed (e.g., the service M&O access point could be through a M&O interface that could be running on a volatile device, which could be unexpectedly disabled).

Of course, to overcome this problem, the device deployer could request to the DS to deploy the service M&O artifacts on stable nodes (e.g., on nodes belonging to the MNO core network). However, other possibility would be to delegate over the SRS.

This SRS is a stakeholder support service containing information on the current execution environment for the already deployed services. Its function is precisely to support the managing stakeholders so that they can locate the nodes on which their services are running at all times, thus supporting the possible change of nodes due to the volatility at the extreme edge domain. Based on this, the user would just need to provide the SRS with the "service handler" provided by the DS at deployment time. The SRS is assumed to work automatically, updating

in real time its registers with the information about the infrastructure nodes on which all the services deployed on the network are running, so that it can perform that translation from the “service handler” into the current service M&O access point.

As with the other stakeholder support services mentioned before, the SRS is also envisaged with a user interface to perform the service-handler/access point translation. Also, as with the other stakeholder support services, the implementation could be very different depending on the resources used in the implementation of this interface, which could range from a simple console command accepting the service handler as parameter and returning the access point as result, to a complex GUI able to show the exact location of each service component in real time.

As for the architectural implications of this decentralised approach, these can be approached from two perspectives: topological and functional.

In what regards the topological perspective, the decentralised M&O approach indeed poses relevant implications compared to the MNO-centric approaches. In this case, the M&O system can no longer be understood as contained in a single block with strictly defined boundaries, functions, and components, as it is usually represented in MNO-centric architectural diagrams, because as explained above, the M&O functions in the decentralised approach are defined and deployed ad-hoc for each network service, as an additional set of service components or functions dedicated to the service orchestration tasks, and thus deployed in a decentralized way. Furthermore, even those common architectural blocks dedicated to the initial instantiation of services (deployment node, IRN, etc.) are also deployed in a decentralized manner, distributing multiple instances of them through the whole network. I.e., in this approach the design by itself targets the network as a whole in a decentralized fashion, considering the rich ecosystem of resources and stakeholders envisaged for the future 6G networks as a whole, instead of focusing only on what should be enabled within the MNO boundaries. As shown in Figure 11-16, the MNO is considered as one of the various stakeholders in the ecosystem, a one quite relevant of course, but in close interaction with the other stakeholders as well.

However, even considering the previous, from the functional perspective the decentralized architecture still allows implementing the M&O functionalities that are normally considered in the MNO-centric approaches. This is because these MNO-centric approaches can be understood as "particular cases" within the decentralised domain as a whole. That is, the centralized M&O systems of a particular MNO could be considered as "management services" specific to that operator, which would be part of the whole set of services distributed throughout the network[2]. Considering this, the decentralised approach is not considered to be a major constraint; quite the contrary, since this approach provides great flexibility to each stakeholder (including MNOs) for designing the required M&O resources. The main requirement for this would be to design the MNO network services (which could be also management services) relying on the cloud-native principles (e.g., relying on the SBMA approach), and embedding their own M&O mechanisms as explained before. Based on this, in the same way the MNO is considered "just another stakeholder" in the ecosystem, the specific services of an MNO would also be considered as "just another set of services" in the decentralised approach. Besides, and as described before, those MNO specific management services could rely on specific M&O solutions that could be decided by the MNOs according to their specific resources and requirements. I.e., each M&O solution of each operator would be as a particular solution based on specific technologies and practices, but fully integrated in the ecosystem by relying on the general cloud-native design principles and the new set of the distributed stakeholder support services mentioned above.

An example illustrating one of these particular solutions is provided in Figure 11-20, which aligns the decentralised M&O concept with the E2E System Blueprint that is being considered in Hexa-X-II as the general E2E System Design for a specific operator (see Architecture Overview - Figure 2-1). As it can be seen, to highlight the architectural implications of the decentralized approach, the following modifications have been introduced:

[2] Please, remember that network services can be both: end-user oriented services (application services) and management services (e.g., those services used by operators -or other stakeholders- to manage their own resources or services, or those of third parties).

1. The MNO (delimited by the red line) is not considered an isolated entity, but an integral part of the cloud continuum, which here is represented by the dots and lines mesh background, and which would extend indefinitely beyond the limits of the figure.
2. The dots in this mesh would represent different nodes of the network infrastructure as a whole. As it can be seen, some of these nodes represent different instances of the stakeholder support nodes that we have been mentioning in this section, i.e., deployment node, IRN, ISPM and SRS.
3. Those nodes beyond the boundaries of the MNO would represent the extreme-edge domain for this particular operator (remember that the concept of the extreme-edge is relative: a node belonging to a specific stakeholder can be seen as an extreme-edge node by other stakeholders).
4. On the other hand, the nodes within the scope of the MNO represent the MNO's own infrastructure. Thus, the Infrastructure Layer represented in the original blueprint figure is here represented by the same mesh that represents the network continuum, but within the MNO Scope, to emphasize that the MNO's infrastructure is also part of the continuum. However, this does not really represent a drastic functional change compared to what it was originally represented in the original Figure 2-1: it simply represents the network continuum already as in the original blueprint, but in a more abstract way, using just as a set of nodes and lines graph.
5. On the other hand, in the Application Layer, a deployment node has been highlighted together with the GUI that was suggested before for this kind of nodes. As explained, deployment nodes are those nodes used to define and deploy the network services (based on microservices from different stakeholders), so it is considered that this deployment node can perform some of the envisaged functionalities of the Application Layer considered in the blueprint. However, this is only an abstract representation; of course, the same operator could have multiple deployment nodes, and probably, other application or service deployment-oriented functionalities that are not considered part of the deployment node (e.g., BSS functionalities).
6. Finally, and returning to the Infrastructure Layer, different "clouds" have been depicted, representing different services deployed on the infrastructure (“Service 1” to “Service n”). As it can be seen, these services can expand beyond the operator's boundaries, including extreme-edge resources belonging to other stakeholders. As mentioned, each of these services would include its own M&O mechanisms, and could be made of different network functions, including those defined in the Network Functions Layer, and the Pervasive Functionalities defined in the blueprint. For example, "Service n" depicted in the figure is intended to represent a specific management service including specific functionalities of the Management and Orchestration blue block in the blueprint.

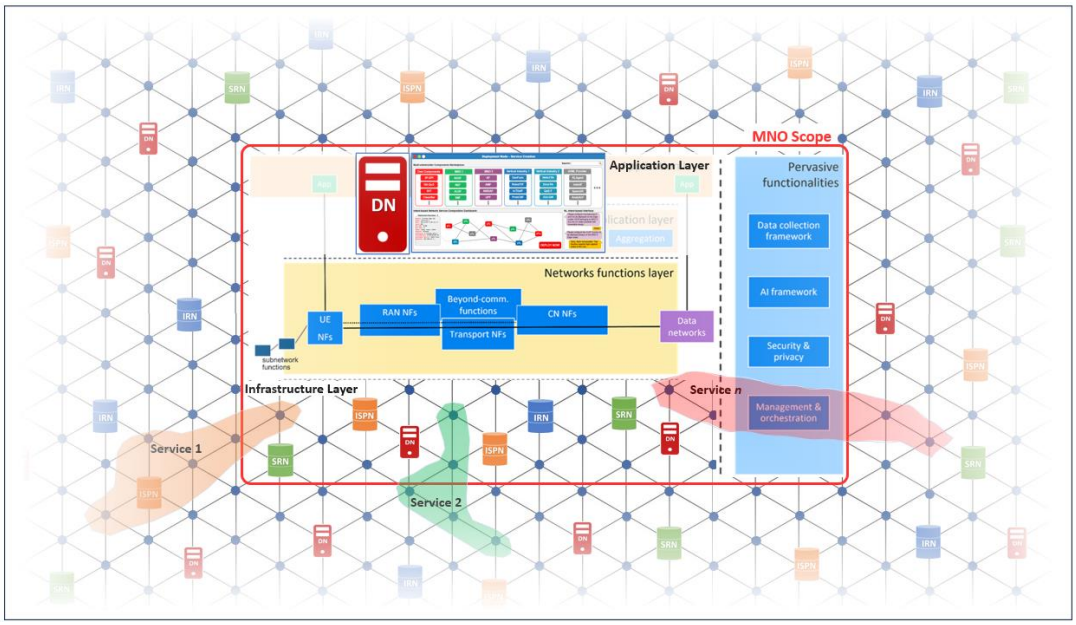


Figure 11-20 Decentralized M&O. Architectural Implications.

In summary, what Figure 11-20 represents is basically the same blueprint as the one in Figure 2-1, but highlighting the distributed nature of the infrastructure (which can go beyond the MNO boundaries) and

representing also the stakeholder support nodes described in this section. In line with what was described in this section these functions (or functionalities) would be implemented cloud-native, based on a micro-services approach, being some of them aggregated to compose network applications oriented to end users, while others could compose management services, i.e., management-oriented network services for the MNO itself or for third parties. However, and as mentioned above, this approach based on this blueprint would be just "one particular approach". A similar exercise to this one done here with respect to this blueprint could also be done for other MNO-centric architectural designs.