Call: H2020-ICT-2020-2

Project reference: 101015956

Project Name:

A flagship for B5G/6G vision and intelligent fabric of technology enablers connecting human, physical, and digital worlds

Hexa-X

Deliverable D5.3
Final 6G architectural enablers and technological solutions

Date of delivery:	30/04/2023	Version:	1.0
Start date of project:	01/01/2021	Duration:	30 months

Document properties:

Document Number:	D5.3
Document Title:	Final 6G architectural enablers and technological solutions
Editor(s):	Mårten Ericson (EAB), Hannu Flinck (NOF), Panagiotis Vlacheas (WINGS), Stefan Wänstedt (EAB)
Authors:	Hannu Flinck (NOF), Hasanin Harkous (NOG), Bahare Masood Khorsandi (NOG), Petteri Pöyhönen (NOF), Janne Tuononen (NOF), Mårten Ericson (EAB), Stefan Wänstedt (EAB), Merve Saimler (EBY), Mehdi Abad (EAB), Riccardo Bassoli (TUD), Frank H.P. Fitzek (TUD), Panagiotis Vlacheas (WINGS), Giuseppe Avino (TIM), Markus Dominik Mueck (INT), Thomas Luetzenkirchen (INT), Dario Sabella (INT), Giacomo Bernini (NXW), Giovanni Nardini (UPI), Giovanni Stea (UPI), Slawomir Kuklinski (ORA), Ricardo Marco (ATO), Adrian Gallego (ATO)
Contractual Date of Delivery:	30/04/2023
Dissemination level:	PU[1]/
Status:	Final
Version:	1.0
File Name:	Hexa-X_D5.3_v1.0

Revision History

Revision	Date	Issued by	Description
0.1	2022-12-01	Hexa-X WP5	Draft
0.2	2023-01-31	Hexa-X WP5	For cross-WP review
0.3	2023-02-31	Hexa-X WP5	For external and PMT review
0.4	2023-03-30	Hexa-X WP5	For GA review
1.0	2023-04-27	Hexa-X WP5	Final version

[1] CO = Confidential, only members of the consortium (including the Commission Services)

PU = Public

Abstract

This document is the third and last deliverable of WP5, D5.3. The first deliverable, D5.1 [HEX-D51], included a gap analysis of existing architectures and proposed eight architecture principles for the 6G architecture based on the gap analysis. Out from these principles, [HEX-D51] proposed several enablers for intelligent distributed networks, enablers for new network topologies, and enablers for cost-efficient deployment of 6G networks. In [HEX-D52] these enablers were developed, together with several so called frameworks, such as AIaaS, FLaaS, analytics, programmability, CaaS and mesh networks management. These frameworks had in common that they all needed to access cross-domain data (analytics) and spanned over several network domains (such as the devices, RAN, core network functions, etc). In this deliverable, the evaluation of the enablers and the corresponding frameworks continues.

Keywords

6G architecture, Intelligent network, AI, AI as a service, Programmability, Network automation, Flexible network, Mesh networks, NTN, Efficient network, RAN cloudification, Service-based architecture, Compute as a service.

Disclaimer

The information and views set out in this deliverable are those of the author(s) and do not necessarily reflect views of the whole Hexa-X Consortium, nor the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101015956.

Executive Summary

This document is the third and last deliverable of WP5, D5.3. The first deliverable, D5.1 [HEX-D51], included a gap analysis of existing architectures and proposed eight architecture principles for the 6G architecture based on the gap analysis. Out from these principles, [HEX-D51] proposed several enablers for intelligent distributed networks, enablers for new network topologies, and enablers for efficient network. In [HEX-D52], these enablers were developed, together with several so-called frameworks, such as AI as a Service (AIaaS), Federated Learning as a Service (FLaaS), analytics, programmability, Compute as a Service (CaaS) and mesh networks management. These frameworks had in common that they all needed to access cross-domain data (analytics) and spanned over several network domains (such as the devices, RAN, core network functions, etc). In this deliverable, the evaluation of the enablers and the corresponding frameworks continues.

To enable an **Intelligent network,** frameworks are developed for AIaaS and programmability. Common services and functions for consumption of an in-network AI are needed to enable 6G intelligent networks and have a unified exposure (through a Common API Framework) approach to facilitate AI services consumption.

Flexible network aim to enable extreme performance, scalability, and global service coverage that can be extended from core, edge and far edge. To this means, a new framework for mesh ad hoc device networks to enable an increased coverage and capacity on a demand basis are presented in this deliverable. Global service coverage" is shown feasible assuming a realistic satellite network constellation that allows efficient inter-satellite-link hops.

The 6G architecture should enable **Efficient network**. A possible 6G service-based architecture is developed with fewer interfaces and processing points. Another important aspect of efficient network is the total cost of ownership (TCO). In this deliverable, a method is developed on how to perform a qualitative TCO analysis.

The "Network evolution and expansion towards 6G" quantified targets are evaluated. One of the targets is to achieve Simultaneous high data rate (0.1 Tbps) and low End-to-End (E2E) latency (less than 1 ms E2E). The E2E latency is estimated for a combined cloud RAN and Core network scenario with a lower layer split for the radio unit. Assuming fibre and a server not further away than 50 km, it is possible to achieve a user plane latency lower than 1 ms for high data rates. For the full (100%) global service coverage, the conclusion is that it is feasible to support global coverage with a realistic Low Earth Orbit (LEO) constellation that allows efficient inter-satellite-link hops (in order to achieve coverage over ocean areas). Further, for the (99%) of global population reached with (>1 Mbps), the investigation in this document shows that it is possible to serve very low population density areas (where terrestrial networks are not the main viable solution) with 1 Mbps/user with at least 14,000 satellites in orbit. The underlaying assumption here is that there is a terrestrial network for areas with higher population density. Note that the results depend to a large extent on the simulation parameters used, such as the antenna gain, transmit power, bandwidths, etc.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

3D	Three-dimensional
3GPP	3rd Generation Partnership Project
4G	4th Generation mobile wireless communication system
5G	5th Generation mobile wireless communication system
5GC	5G Core network
5GS	5G System
AF	Application Function
AGV	Automated Guided Vehicles
AI	Artificial Intelligence
AIaaS	AI-as-a-Service
AI/ML	Artificial Intelligence / Machine Learning
AMF	Access and Mobility management Function
API	Application Programming Interface
AS	Access Stratum
ATSSS	Access Traffic Steering, Switch and Splitting
B5G	Beyond 5G
BBU	BaseBand Unit
BS	Base Station
CA	Carrier Aggregation
CaaS	Compute-as-a-Service
CapEx	Capital Expenditures
CAPIF	Common API Framework
CHO	Conditional Handover
CL	Control Loop
CLI	Command Line Interface
CN	Core Network
CC	Confidential Computing
CP	Control Plane
CPU	Central Processing Unit
C-RAN	Centralized RAN
CU	Central Unit
D2D	Device-to-Device
DC	Dual Connectivity
DE	Decision Element
DFP	Dynamic Function Placement
DL	Downlink

D-MIMO	Distributed MIMO
DU	Distributed Unit
DP	Differential Privacy
E2E	End-to-End
EAS	Edge application server
ECF	Exposure and Coordination Framework
EES	Edge Enabler Server
eMBB	Enhanced Mobile Broadband
EN-DC	E-UTRA-NR Dual Connectivity
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
EU	European Union
E-UTRA	Evolved Universal Terrestrial Radio Access
FCAPS	Fault, Configuration, Accounting, Performance, Security
FD	Functional Domain
FED-XAI	FEDerated eXplainable AI
FL	Federated Learning
FLaaS	Federated Learning as-a-service
FLEX-TOP	FLEXible TOPologies
FLM	FL Local Manager
FoReCo	Forecast-based recovery in Real-time remote Control of robotics
FPC	FL Process Controller
FPCE	FL Process Computation Engine
FPGA	Field Programmable Gate Array
FR1	Frequency Range 1
FSP	FL Service Provider
GEO	Geostationary Equatorial Orbit
gMURI	generalised Multiradio Interface
GPRS	General Packet Radio Service
GSMA	Global System for Mobile Communications Association
GTP	GPRS Tunnelling Protocol
H2020	Horizon 2020
HAPS	High-Altitude Platform Station
HE	Homomorphic Encryption
HO	Handover
HTTP	Hyper Text Transfer Protocol
ICT	Information and Communication Technology
IEEE	Institute of Electrical and Electronics Engineers

IoT	Internet of Things
IP	Internet Protocol
ISD	Inter-Site Distance
ISL	Inter-Satellite Link
ITU	International Telecommunication Union
KPI	Key Performance Indicator
KVI	Key Value Indicator
LCM	Life-Cycle Management
LEO	Low Earth Orbit
LTE	Long Term Evolution
M&O	Management and Orchestration
MA	Moving Average solution
MA PDU	Multi-Access PDU
MAC	Medium Access Control
MANET	Mobile Ad hoc NETwork
MAPE	Monitoring-Analysis-Planning-Execution
MC	Multi-Connectivity
MDAS	Management Data Analytics Service
MDT	Minimization of Drive Test
MEA	Minimum Elevation Angle
MEC	Multi-access Edge Computing
MEP	Multi-access Edge Platform
MIMO	Multiple-Input Multiple-Output
ML	Machine Learning
MLOps	Machine Learning Operations
MM	Mobility Management
MNO	Mobile Network Operator
MO	Managed Object
MR	Mixed Reality
MTC	Machine Type Communications
MU-MIMO	Multi User MIMO
NAS	Non-Access Stratum
NEF	Network Exposure Function
NF	Network Function
NFV	Network Function Virtualization
NFVI	Network Functions Virtualization Infrastructure
NGAP	NG Application Protocol
NG-RAN	Next Generation RAN

NGEN-DC	NG-RAN EUTRA-NR Dual Connectivity
NIC	Network Interface Card
NN	Neural Network
NP	Nondeterministic Polynomial
NPN	Non-Public Networks
NR	New Radio
NS	Network Service
NTN	Non-Terrestrial Network
NWDAF	Network Data Analytics Function
OpEx	Operating Expenditures
OS	Operating System
P4	Programming Protocol-independent Packet Processors
PDCP	Packet Data Convergence Protocol
PDU	Protocol Data Unit
PFCP	Packet Forwarding Control Protocol
PHY	PHYsical layer
PNI-NPN	Public Network Integrated NPN
PoC	Proof-of-Concept
PoP	Point of presence
PSA	PDU Session Anchor
QAM	Quadrature Amplitude Modulation
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RAT	Radio Access Technology
REST	REpresentational State Transfer
RF	Radio Frequency
RIC	RAN Intelligent Controller
RIM	Remote Interference Management
RLC	Radio Link Control
RLF	Radio Link Failure
RMSE	Root-Mean-Square Error
RRC	Radio Resource Control
RRS	Reconfigurable Radio Systems
RS	Resource Scaling
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
RTT	Round-Trip Time

SA	Standalone (NR (5G) network)
SACK	Selective ACKnowledgement
SBA	Service Based Architecture
SBI	Service Based Interface
SBMA	Service Based Management Architecture
SCG	Secondary Cell Group
SCP	Service Communication Proxy
SCTP	Stream Control Transmission Protocol
SFC	Service Function Chaining
SDAP	Service Data Adaption Protocol
SDO	Standard Definition Organization
SDN	Software Defined Networking
SINR	Signal to Interference plus Noise Ratio
SL	Supervised Learning
SLA	Service Level Agreement
SMC	Secure Multi-party computation
SMF	Session Management Function
SMO	Service Management and Orchestration
SNPN	Standalone NPN
SOA	Service Oriented Architecture
SpL	Split Learning
SON	Self-Optimized Networks
TCO	Total Cost of Ownership
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TM	TeleManagement (TM Forum)
TN	Terrestrial Network
TSN	Time Sensitive Networking
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UE	User Equipment
UL	Uplink
UM	Unacknowledged Mode
UP	User Plane
UPA	User Plane Adapter
UPF	User Plane Function
URLLC	Ultra-Reliable Low-Latency Communication
VAR	Vector AutoRegression

VM	Virtual Machine
VNF	Virtualised Network Function
WANET	Wireless Ad hoc NETwork
WP	Work Package
XAI	eXplainable AI
ZSM	Zero-Touch Service Management

1 Introduction

As 5G is currently being made available to more and more users globally, some of the benefits of it are starting to be realized, e.g., the increased capacity, evolution of download speeds and the new use cases that 5G brought compared to previous cellular generations (e.g., Ultra-Reliable Low-Latency Communication (URLLC), private networks, etc). There is a constant change and increased demand for new types of services and this change is one of the reasons why the industry has to start preparing for the next generation of mobile systems. Another reason is the changes to society. Society in general, with large global variations, is becoming increasingly digital. Many services that some years ago or just recently required face-to-face interactions are now provided through a mobile equipment. The changes have been made possible by the evolution of the cellular industry, by the evolution of industries providing services and the evolution of user behaviour, whether it is a behaviour forced upon users, e.g., due to new business models, or something that users have asked for. What is certain is that the change continues.

Technological breakthroughs in other areas will also affect users' lives and the development of cellular systems. Some of them are mentioned briefly here since they set the direction for the studies. For instance, AI has been discussed for a long time but the effects on ordinary users have been minimal up to this point. However, recently several interesting AI applications have been made available to the public; applications that render images or music from a short description as well as applications that can take part in discussions, answer questions and write computer code. The AI applications will for sure affect how users interact with cellular systems but also how the systems are managed.

Another area that increasingly affects society is sustainability in a general sense. Recent happenings have reminded society that energy is not an unlimited resource. Use of energy affects cellular systems in many ways. A direct effect is the energy needed to run the networks and this is an important task of the project, i.e., to look at solutions that help reduce the energy footprint of the networks. It is worth emphasizing that at the same time there are expectations on the network performance and what the networks should do. A secondary effect on energy usage is, e.g., an application in a device or in the car, which provides a route that minimizes power consumption of the car.

The Hexa-X project is a flagship initiative in which to bring together key industry stakeholders in Europe, the full value chain of future connectivity solutions and major research institutes to work together, providing research and development towards B5G/6G. The work towards 6G will consider the abovementioned changes in the society, technological developments and other important inputs. The project comprises several work packages that study different areas. In this report results from work in WP5 are presented.

The overarching objective of WP5 is to develop architectural components that support a new flexible network design, full AI integration and network programmability and, at the same time, streamline and redesign the architecture for a network of networks. WP5 will develop disruptive technology components aiming to realise a fundamental impact on existing network and device architectures.

This document is the third and last deliverable of WP5, D5.3. The first deliverable, D5.1 [HEX-D51], included a gap analysis of existing architectures and proposed eight architecture principles for the 6G architecture based on the gap analysis. Out from these principles, [HEX-D51] proposed several enablers for intelligent distributed networks, enablers for new network topologies, and enablers for cost-efficient deployment of 6G networks. In [HEX-D52], these enablers were developed, together with several so-called frameworks, such as AI-as-a-Service (AIaaS), Federated Learning as-a-Service (FLaaS), analytics, programmability, CaaS, and mesh networks management. These frameworks have in common that they all need to access cross-domain data (analytics) and span over several network domains (such as the devices, RAN, core network functions, etc). In this deliverable, the evaluation of the enablers and the corresponding frameworks continues.

1.1 Hexa-x objectives on network evolution and expansion towards 6G

1.1.1 WP5 objectives

The main objective of this document is to address the objectives of WP5 as defined by the "Network evolution and expansion towards 6G" [HEXA], see Table 1-1. As can be seen, the progress of the first objective, WPO5.1, is considered fully addressed in [HEX-D51], while the other objectives are fulfilled by this deliverable.

Table 1-1 WP5 objectives and the progress.

Objective	Objective description	Progress
WPO5.1	Identify technology trends, use cases and requirements for architecture transformation.	Fully addressed in [HEX-D51].
WPO5.2	Develop technical enablers for **Intelligent network** capable of full AI integration and network programmability to boost connected intelligence. Distributed AI agents, running in both network functions and wireless devices, will be supported to provide increased network performance, while preserving the privacy of the users.	AIaaS framework with required services and functions are developed, together with the analytics framework needed (Section 3.1.2). A complete programmability framework is also developed (see Section 3.1.4).
WPO5.3	Enable extreme performance and global service coverage within **Flexible network**. Vertical requirements will be addressed such as ultra-low latency via local ad hoc networks, cost-efficient global service coverage, and functionalities for securely managing local ad hoc networks in coordination with the infrastructure.	A concepts for ad hoc mesh network controller with management solution are developed and evaluated (see Section 4.2). The global service coverage are addressed with analysis of different TN architecture, including evaluations (see Section 4.3).
WPO5.4	The **Efficient network** will extend the existing Service Based Architecture for the Core Network to the Radio Access Network and wireless devices, streamlining and redesigning the functional architecture, merging or removing redundant functionalities and defining a clear functional split to reduce the Total Cost of Ownership related to network integration and implementation and improve network energy efficiency.	A concept for more self-sustained network functions assuming a cloud native Core Network (CN) and RAN extending current SBA concept (see Section 5.1). A method on how to perform a qualitative TCO analysis for some Hexa-X use cases are also developed (see Section 5.4).

In [HEX-D51] we initiated and started a discussion about the architectural enablers and in [HEX-D52] the enablers were conceptualized and to some extent analysed. In this document, we aim to also give a better view on how the different enablers integrate with each other in a flexible, efficient, and secure manner as well as continue with analysing the enablers and the concepts.

1.1.2 Measurable results

The solutions outlined in this document addresses the main project-level objective **Network evolution and expansion towards 6G [HEXA]**, which is shared by WP5 and WP7 [HEX-D73]. The following measurable results for the objective are completed in the deliverable:

Architecture components enabling integrated and distributed AI and programmability.

We have developed several components for distributed AI and also developed more generic frameworks (a collection of components and functions). To enable a distributed AI, an AIaaS framework with required services and functions are developed, together with the analytics (data collection) framework needed (Section 3.1.2). A complete programmability framework has been developed. The framework enables the network to reprogram certain functionality over all nodes and functions in the network (UE, RAN CN etc), controlled via the management and orchestration (see Section 3.1.4).

Design of a flexible network enabling global service coverage and extreme use cases, integrating heterogenous accesses (e.g., satellites, High Altitude Platforms (HAPs), multi-hop, device to device (D2D)).

To address the global service coverage, a realistic global satellite network is evaluated. The deliverable shows that it is possible to achieve global service coverage for 100% of the area assuming inter-satellite-link hops are possible (see Section 4.3.1). Another architecture solution for HAPS or Satellites to split the RAN between UAVs (on low altitude) and HAPS/satellites (with low orbit). This solution takes advantage of the benefits given by the functional split of the softwarized baseband unit and may provide a more stable connection for the devices since it can connect via almost stationary HAPS unit (see Section 4.3.2). Further on, we have developed a concept on how to optimize placement of NFs for latency for a Non-Terrestrial Network (NTN) scenario. Latency aware NF function placement can reduce the control plane latency introduced in satellite backhaul and fronthaul scenarios for Edge computing (see Section 5.2)

Architectural solutions enabling connectivity of a wide range of devices and sub-networks, for a wide range of use cases and scenarios.

A new framework for mesh ad hoc (sub-network) device networks, to enable increased coverage and capacity on a demand basis, is developed (see Section 4.2). The ad hoc network is based on a mesh D2D technology and is controlled by a management network that gives a detailed control of the mesh network. The ad hoc mesh network is evaluated to illustrate the ability of the ad hoc network to optimize cost, throughput, energy etc. A possibly new solution to enable a reliable connectivity and utilize the available 6G spectrum is a new 6G multi-connectivity solution for 6G, which combines the best features from Carrier Aggregation (CA) and Dual Connectivity (DC) [HEX-D52] (see Section 4.1). These measurable results are shared with WP7 [HEX-D73].

Architecture for a service-based network (CN, RAN and devices).

A possible 6G service-based architecture with fewer interfaces and processing points are outlined in Section 5.1. Part of the concept are evaluated in terms of latency for a handover procedure in Section 5.3. Further on, in [HEX-D52] we introduce the concept of 6G-RAN-CN function elasticity, which is achieved by co-locating some of the common 6G-CN NFs with the 6G RAN-CP in the cloud environment. Co-locating critical signalling processing together with 6G-RAN-CP in the regional edge cloud, signalling performance is improved thus reducing latency. Based on the cloud native RAN and CN approach, a Compute as a Service (CaaS) framework is proposed. It allows delegating/offloading generic application-related workloads. (see Section 5.6). An investigation of Joint Communication and Sensing (JCAS) shows that it can potentially improve the 6G mobility for some special scenarios (see Section 5.7.2).

1.1.3 Quantified target results

The document also finalizes the analysis of the so-called “quantified targets” for the “Network evolution and expansion towards 6G” objective. The quantified targets are also shared with WP7 [HEX-D73]. The quantified targets are defined in the Hexa-X project proposal [HEXA] and part of the overall project objectives.

Access links supporting simultaneous high rate and low E2E latency (>0.1 Tbps @ <1 ms E2E).

A scenario where the cloud RAN and CN is co-located with a lower layer split for the radio unit is used. The latency is thereafter estimated for all nodes and protocol stack. The physical layer results are taken from [HEX-D23]. Assuming fibre and a server not further apart than 50 km, it is possible to achieve a user plane latency lower than 1 ms for data rates higher than 0.1 Tbps (see [HEX-D23] for the data rate results). This target is addressed in Section 6.1.

Supporting (>100 bn) connected devices in the network.

To evaluate the ability to support the target of 100 billion connections, we use two cities with high population density, in this case Paris and Athens. To get the corresponding city target connection density, we scale the city population with the Earth's population (8 billion) and multiply this with 100 billion connections. The findings are that the maximum number of connections achieved in [37.910] for NR exceeds the target connections with 4-5 times. See Section 6.2 for more details.

(>99%) of global population reached with (>1 Mbps) data rates at sustainable cost levels; Full coverage (100%) of world area.

For the "full (100%) global service coverage" target, the conclusion is that is feasible to support assuming a Low Earth Orbit (LEO) constellation that allows efficient inter-satellite-link hops (in order to achieve coverage over ocean areas) with at least 600 satellites. Further, for the target of "99% of global population reached with more than 1 Mbps", the investigation in this document shows that it is possible to serve very low population density areas (where terrestrial networks are not the main viable solution) with 1 Mbps/user assuming at least 14000 satellites in orbit. The underlaying assumption here is that there is a terrestrial network for areas with higher population density. Note that the results depend to a large extent on the simulation parameters used, such as the antenna gain, transmit power, bandwidths, etc. See Section 6.3 and 6.4 for more details.

1.2 Structure of the document

The document is structured as follows: Chapter 2 gives a short overview of the overall architecture and the enablers and introduces a common integration between the different concepts (frameworks) presented here. Chapter 3 includes the enablers for Intelligent network, Chapter 4 presents the Flexible network enablers, Chapter 5 presents the Efficient network. Chapter 6 shows the final results regarding the Hexa-X quantified targets, and finally, Chapter 7 is the conclusions. References can be found in Chapter 8 and there is also a list with common terminologies used in the documents in Annex A.1.

2 Architecture overview

This section gives a brief overview of the architecture enablers we develop in this document, see Figure 2-1. The architecture enablers are divided into three parts: the Intelligent network (blue boxes in Figure 2-1), the Flexible network (green boxes) and the Efficient network (orange boxes).

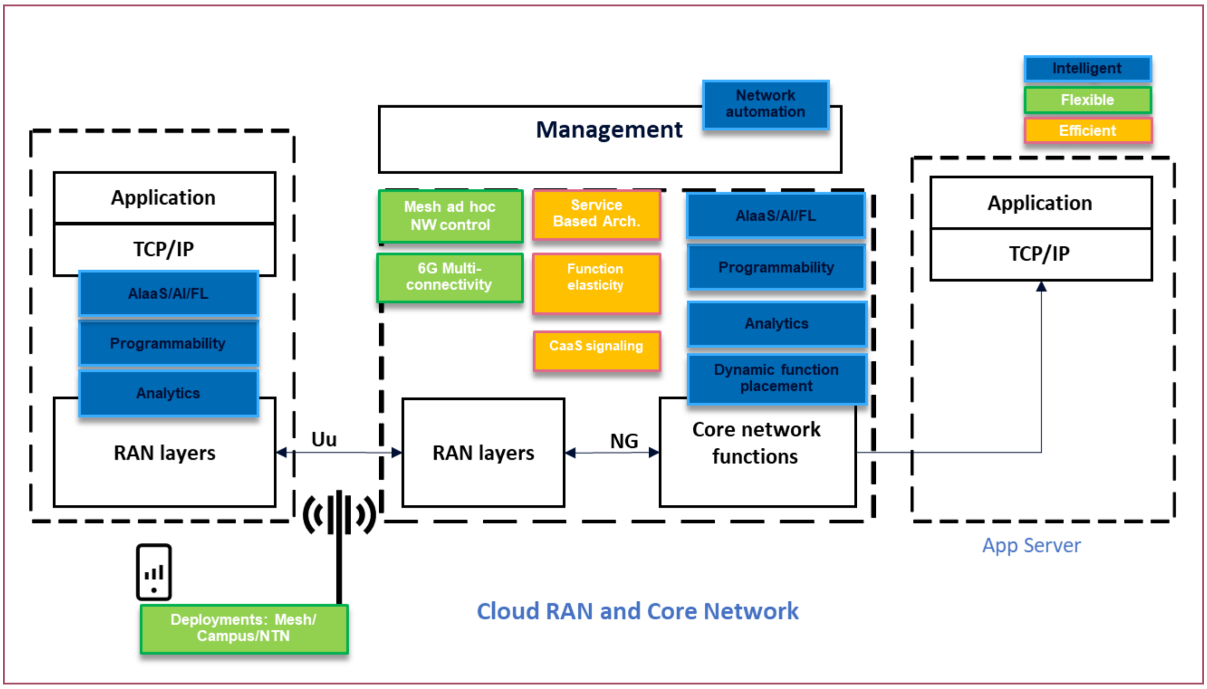


Figure 2-1 Overview of the 6G architecture enablers introduced in [HEX-D52].

The Intelligent network enablers (blue boxes in Figure 2-1) deal with enablers for developing a fully integrated AI and programmable networks. The enablers that are developed are grouped into independent frameworks that cover specific functionality and supporting mechanisms to implement for an enabler. A framework exposes services to a consumer of a framework. AI-as-a-Service (AIaaS) provides the necessary machinery for managing, distributing and training AI models to AI-agents. Federated Learning as-a-Service (FLaaS) defines means to discover and join federation of UEs, in order to exploit federated AI models and, possibly, collectively participate in model training. Analytics framework provides basic means for data collection, storing and analytic functions over cross domain settings. The network automation and orchestration are an integrated part of this intelligent network and are using the AI and analytics to run the network in a fully automated manner. We assume that intelligent network enablers exist in the device as well as in the network. The frameworks can interact and use each other's service as we explain in the next section.

The Flexible network enablers (green boxes in Figure 2-1) consist of a mix of enablers for radio resource management and for supporting deployments such as mesh networks and Non-Terrestrial Networks (NTNs). The mesh ad hoc network control enables to quickly set up new networks on a demand basis using Device-to-Device (D2D) communication by introducing new enablers to control the involved nodes. The 6G Multi-Connectivity (MC) concept is an effort to enhance 5G features to support the new 6G requirements such as sub-THz frequencies and even higher flexibility. The campus and satellite networks (NTNs) should be an integral part of 6G, in order to give full global coverage.

The Efficient network enablers (orange boxes in Figure 2-1) are a collection of new ways to streamline the Radio Access Network (RAN) and Core Network (CN) architecture, minimize the signalling needs and make the architecture more flexible (function elasticity). The notable enablers here include methods to extend the Service Based Architecture (SBA) also to the RAN, new design of the network functions in order to make them more self-sustained as well as the possibility to deploy Network Functions (NFs) in different cloud environments (network refactoring). This also includes a concept for Compute as a service (CaaS).

2.1 Exposure and Coordination Framework (ECF)

As can be seen from Figure 2-1, [HEX-D52] introduced a number of different enablers and enabler frameworks (e.g., AIaaS, FLaaS, analytics, programmability, CaaS and mesh networks management). These frameworks can leverage each other's services, even though we do not mandate that all of them must be deployed in a given network configuration. These individual frameworks of Hexa-X are considered to be autonomous domains with their dedicated internal functionality and business logic. These frameworks provide on-demand access to their services, information, and resources. Access to the provided services and other resources are controlled by policies provided by the management and orchestration system. In this section, we describe how these frameworks can interact to form a full system. This leads to notion of cross framework interactions, that we propose to be coordinated by an Exposure and Coordination Framework (ECF). The ECF should meet the following requirements:

- It should provide frameworks discovery mechanism of the given deployment of Hexa-X architecture that consists of multiple frameworks. It should further provide a discovery mechanism for the APIs that a given framework is willing to expose to others depending on its policies. It is noted that the instantiation of a framework depends on the deployment scenario and may be different in different use case configurations (principles 3 and 7 [HEX-D51]),
- It should manage the cross-framework connections and interactions according to given policies,
- It should share data and information between the different frameworks of Hexa-X architecture
- It should manage potential conflicts and provide closed-loop control across the frameworks (principle 2 [HEX-D51]).
- It should be extensible to include new and yet unspecified frameworks (principle 4 [HEX-D51]).

The approach to cross-framework interactions depends on how tightly or loosely the defined frameworks are to be integrated. Assuming each framework is realized by a set of interconnected functions (as explained in Chapter 3 where the various frameworks are introduced), in case of tight integration, the functions of different frameworks are directly interconnected into a single flat SBA domain. This approach means that all the functions must be under the same trust domain based on SBA internal mechanisms. The "loosely-coupled" approach keeps the frameworks separated and logically isolated, even at the level that different service providers could operate them. Loose integration of the frameworks can be achieved by applying the cross-layer API manager approach introduced in [HEX-D62] that can be implemented through the Common API Framework (CAPIF) of 3GPP [23.222].

In architecture overview of [HEXD62], a cross-layer API management exposure block was defined to communicate between the different frameworks in the different network layers (see [HEX62]). The cross-layer API management follows the behaviour of the Zero-Touch Network and Service Management (ZSM) cross-domain integration fabric [ZSM-002], enabling the capabilities exposure [5GVIN-D31] of the network functionality in the various architectural layers. It makes possible communicating the various Management and Orchestration (M&O) resources within and between administrative domains, although it could be applied more broadly to represent potential federated interactions.

The cross-layer API manager keeps the different frameworks separate and loosely integrated while offering means to manage API exposure with different trust and security levels. CAPIF provides needed functionalities to REpresentational State Transfer (REST) based API authentication, invocation, discovery and usage including charging. However, most of the frameworks rely on the use and processing of data, e.g., for AI/ML model training, inference and actions by the AI agents, collecting and monitoring of telemetry, etc. Therefore, the frameworks need access to various types of data sets and information (e.g., processed labelled data) that could be shared and reused between other frameworks. For the data sharing between the frameworks, we need to complement CAPIF with efficient data sharing for which purpose we recommend using Data Mesh technology [Deh20] that takes care of streaming and synchronizing data between authorized frameworks. Data Mesh supports domain-oriented decentralized data ownership, and it can accommodate various technologies for data pipelining and storage to ensure interoperability. To provide and consume data over the Data Mesh, a Mesh Node [Deh20] is needed in all frameworks of Intelligent network that want to access or share their streaming data. The Mesh Node, see Figure 2-2, implements the control of data sharing and privacy policies by filtering the relevant exposed data based on the policies provided by the Data Mesh management that oversees the cross-framework data sharing. The Mesh Node interconnections with the Data Mesh management would go over API invoker interfaces, see Figure 2-2. Framework internal connections and communications are framework-dependant (i.e., they depend on the specific capabilities and characteristics of the functions building the framework) and are discussed in the subsequent section of Chapter 3. Only those frameworks that would benefit from data sharing implements data mesh components and join it. Data Mesh support is optional.

According to [29.222], the frameworks supporting API exposure can be designed following different deployment models (centralized, distributed, peer-to-peer (multiple CAPIF core function – CCFs) and hierarchical) depending on how the API management functionality (e.g., CAPIF functionality) is split between the domains, i.e., the frameworks. In the centralized deployment model, the CAPIF core function and the API provider domain functions are co-located. The API invoker can interact independently with the CAPIF core function and the API exposing function including the service APIs. The application of this deployment model for ECF is depicted in Figure 2-2 as an example. In Figure 2-2, the Exposure and Coordination Framework is hosting the API Management core functions, management of Data Mesh nodes, cross-framework conflict management and closed-loop governance.

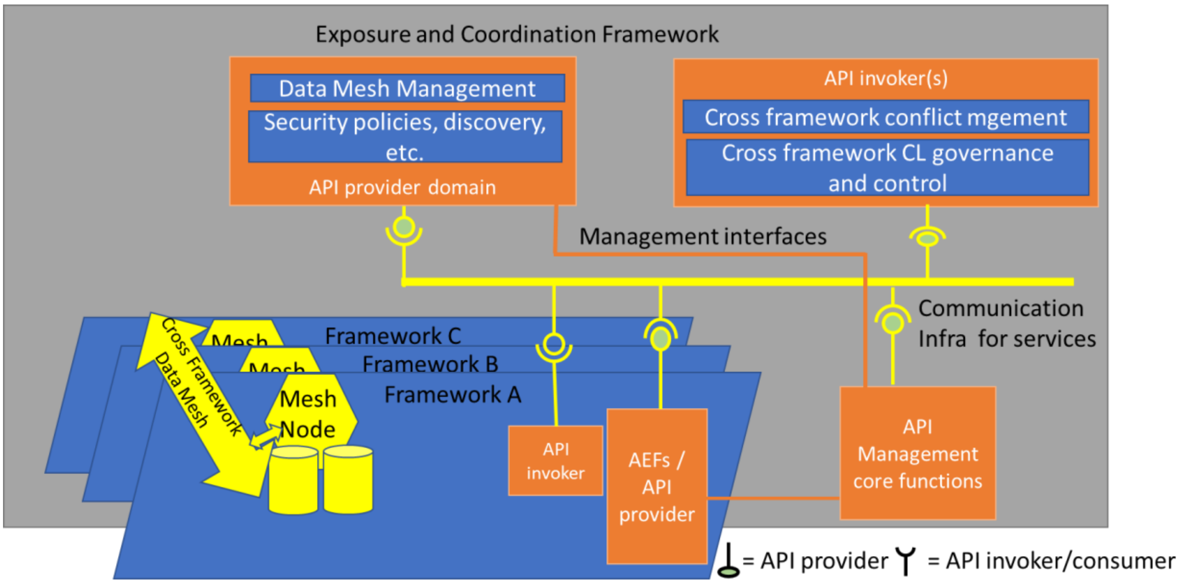

Exposure and Coordination Framework
Data Mesh Management
Security policies, discovery, etc.
API provider domain
API invoker(s)
Cross framework conflict mgement
Cross framework CL governance and control
Management interfaces
Communication Infra for services
Framework C
Framework B
Framework A
Mesh
Mesh
Mesh Node
Cross Framework Data Mesh
API invoker
AEFs / API provider
API Management core functions
= API provider
= API invoker/consumer

Figure 2-2 Proposed Exposure and Coordination Framework based Data Mesh and API manager of [HEX-D62] applying CAPIF Central mode.

Yellow lines represent the service consumption communication infrastructure (e.g., CAPIF1(e)/2(e) reference points within or outside the trust domain tagged by "e", [29.222]), orange lines CAPIF management connections (i.e., CAPIF3/4/5). Other deployment models of API management exposure where the API management functions are distributed among the participating frameworks are also possible.

To summarize, the ECF provides REST-based APIs for transaction type interactions based on cross-layer API management exposure of [HEXD6.2] and a Data Mesh interface for streaming data. It handles management of policies (for accessing APIs, data sharing, security, privacy, etc.) as well as conflict management in case of contradicting operations between the connected frameworks for the case individual frameworks are separate domains.

3 Intelligent network

This section is divided into two major parts. In Section 3.1, a summary and update of Intelligent network frameworks [HEX-D52] is provided with the focus on the services provided and consumed by a given framework across the ECF. Cross-network domain trust and regulatory aspects are discussed under AIaaS in Section 3.1.2. . Programmable management and Programmability framework are discussed in Sections 3.1.3 and 3.1.4 respectively. Applications and evaluations of the Intelligent network are covered in Section 3.2.

3.1 Intelligent network frameworks

AI and ML are beginning to be key valuable tools in the context of mobile networks, as the complexity of the network grows. Since 5G, mobile networks have become much more heterogeneous and complex and, therefore, the number of parameters to be configured over the whole network, to achieve optimal services, has increased exponentially. Consequently, data-based approaches have been raised as the next-generation shift for legacy model-based approaches [WRS+20]. The main advantage of using AI techniques in these kinds of networks is the proven capability that they have to face humongous volumes of data and extract precise, meaningful actions/conclusions from them [GSR+21]. Therefore, in future 6G mobile networks AI is expected to be in charge or support a large set of operations over the whole network (i.e., predictive orchestration [HEX-D43], aid security functions, optimize placement, QoS/QoE monitoring and configuration, etc.). In summary, the role of AI and ML in future 6G mobile networks will be to aid those tasks where legacy techniques are not able to cope with the new conditions and requirements related to those networks e.g., high device heterogeneity, automation, multi-domain and multi-stakeholder environments, wide range of services, etc. Defining and embedding AI/ML functionality as an integral part of Hexa-X 6G architecture has led us to develop a set of enablers, capabilities and services that are grouped into a set of frameworks (e.g., analytics, AIaaS, FLaaS, Programmability, etc.) that collectively form Hexa-X “Intelligent network” [HEX-D51], [HEX-D52]. These frameworks can be deployed independently even though they benefit from the services they provide to each other. In this chapter we look deeper into the services exposed by each of the frameworks and their interactions with each other. The Analytics framework is in charge of collecting, maintaining and storing data from network functions across distributed multi-cloud environment for the purpose of developing analytics. It optionally contains ML model training, inference and repository functionalities that build on top of 5G Network Data Analytics Function (NWDAF) functionality [29.520]. AIaaS framework provides full machinery for supporting and managing AI-agents across cloud continuum. It provides AI services with tailored inference capabilities depending on the specific consumer and automation goal request. AIaaS framework provides services to support closed-loop network and service automation. AI based automation handles the complexity in terms of technology and services to meet various requirements, such as quality, security, and resilience requirements. Programmability framework provides the means to adjust and customize the underlying infrastructure and user devices capabilities under the control of programmability management that can interact with AIaaS and other developed frameworks. The proposed programmability management is hierarchical to ensure end-to-end consistency using local programmability managers for the different domains: User (UE), access network, transport network, and core network.

3.1.1 Analytics framework functional description

The analytics framework [HEXD52] envisioned for the next generation of mobile networks (6G) should be self-contained to provide basic analytics services of mobile networks and to interact with legacy 5G systems that support 5G NWDAF [23.288], [29.520] based approaches. Additionally following the architecture design principle #7 (separation of concerns of network functions), the analytics framework, based on a particular deployment and needed configuration can use the services provided by the other frameworks e.g., AI/ML functionalities from AIaaS framework [HEX-D51], [HEX-D52].

For example, the limited interactions of RAN-CN of 5G system in data collection and the lack of AI agent in 5G RAN do not enable AIaaS natively. The analytics framework can be useful not only for seamless transfer of analytics across domains/planes but can also pave the way to a new AI-enabled architecture that supports distributed AI agents which are providing services such as analytics, prediction, classification, etc. This means, with the help of AI agents, the analytics services envisioned for 6G can analyse data and uncover hidden trends, patterns, and insights in a more automated fashion. In order to implement the analytics framework, the following entities are required. Some of these entities providing analytics services e.g., analytics function, analytics repository, etc. and some other entities are responsible for providing necessary AI/ML functionality for analytics service e.g., ML model training function, ML model repository, etc. As explained above, the 6G analytics framework designed in a way that based on the implementation requirement can be able to host and provide the AI/ML related services locally (local to the analytics framework) and independently from other frameworks in the system (as shown in Figure 3-1) as well as delegate the AI/ML related services to other frameworks e.g., AIaaS and access them through CAPIF. In that case, additional inter-framework communication is required. In the other words this framework can be used as a transitional tool from 5G analytics to 6G analytics.

- An entity which is responsible for ML models e.g., *ML model Training Function*. This entity is able to train the ML models based on the designated data set. The ML model Training Function can produce new ML models or retrain ML models found in the ML model repository with a different data set. This function can fetch the required data set by using the Data Mesh system either from the local (local to the analytics framework) repository or from repositories from the other frameworks e.g., AI framework. All the trained models will be stored on the databases equipped with version controlling features.
- *Analytics Function* is the entity which performs and supports the analytics, inference and prediction services. In order for the Analytics Function to perform the abovementioned services, it has to collect both data and, in some cases, trained ML models. Same as the ML model Training Function, data collection can be facilitated by a data mesh system either from the local repository from the analytics framework or through the data mesh node(s) of other frameworks. In case of trained ML model collection, Analytics Function can invoke the discovery service from the dedicated repository in the analytics framework for the trained models or use the CAPIF API [23.222] to invoke the training service for ML model in AI framework. The Analytics function registers the list of analytics services in the repository for other network functions (local to the analytics framework or from other frameworks) to discover and invoke them, see Figure 3-1.
- *Analytics service repository* is a dedicated repository for the Analytics Function to store the possible analytics, inference and prediction services.
- *ML model repository* is a repository for the trained ML models. This repository needs to support the version control features.

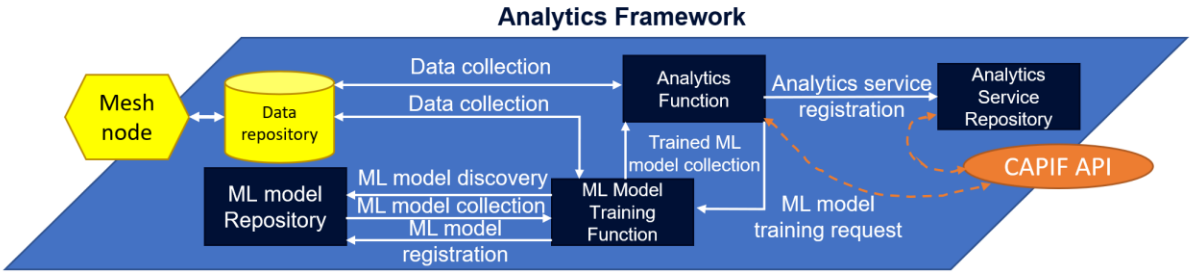


Figure 3-1 Analytics framework and how it connects to other frameworks.

The entities envisioned for the analytics framework providing and consuming services are listed in Table 3-1.

Table 3-1 Analytics framework required and provided services.

Analytics framework entities	Required services	Provided services
ML model training function	ML model discovery	ML model training service
	ML model collection	
	ML model registration	
	Data (set) collection	
Analytics Function	Analytics service registration	Perform Analytics, inference and Prediction
	Data collection	
	ML model training	
	Trained ML model collection	
	ML model discovery	
Analytics service repository		Analytics discovery
		Analytics update
ML model repository		ML model discovery
		ML model update

3.1.2 AIaaS framework and AI functions functional description

The AI as a Service (AIaaS) concept proposed by Hexa-X is built by the combination of several functions offering AI capabilities to a wide set of consumers [HEX-D52], including management and orchestration, other network functions (e.g., belonging to different domains, including RAN and core), application functions, as well as third parties. In practice, the AIaaS is an entire framework which offers a set of AI services and tailored inference capabilities depending on the specific consumer and automation goal request to the AI service itself. Beyond the pure prediction, classification, etc., ML capabilities (which can per-se be consumed in support of full automation at the 6G network and service layers), the AIaaS framework provides additional AI capabilities and services (including training, monitoring, evaluation) to support closed loop network and service automation, targeting their implementation and deployment as cloud-native in-network virtualized functions. Similar to the analytics framework described in the previous section, this AIaaS framework is designed as an independent set of functions implementing and exposing specific AI services, but still capable to consume services from other frameworks. In particular, for some specific use case cases and AI/ML model implementation requirements, the analytics services (and related data produced) offered by the analytics framework may be consumed by the AIaaS framework. In such a case, the CAPIF-like APIs can be leveraged to discovery the required analytics capabilities, while the cross-framework data mesh used to actually access and consume the analytics data,

As described in [HEX-D52], four main AI functions are building the AIaaS framework: AI model repository function, AI training function, AI monitoring function and AI agent. Figure 3-2 shows this AIaaS framework functional decomposition, specifically mapped to the ECF approach described in Section 3.1. While the functional description of each of these AI functions is provided in [HEX-D52], Figure 3-2 aims at highlighting the high-level operational interactions among the functions, including the exposure of the related AI services and functions (aligned with the CAPIF exposure presented in Section 3.1). As shown in the figure, beyond the four core AI functions, the AIaaS framework includes other critical assets, and specifically two logically centralized data stores to collect and expose (either internally to the framework, but also externally to other frameworks) inference data to feed the runtime operation of AI/ML models, and training data to feed the AI training functionalities. The presence of the two data stores aims at clearly separating the AI/ML model training and runtime operation/inference phases, including the possibility to use different data sets (e.g., either coming from different sources, or from the same source but with different data pre-processing).

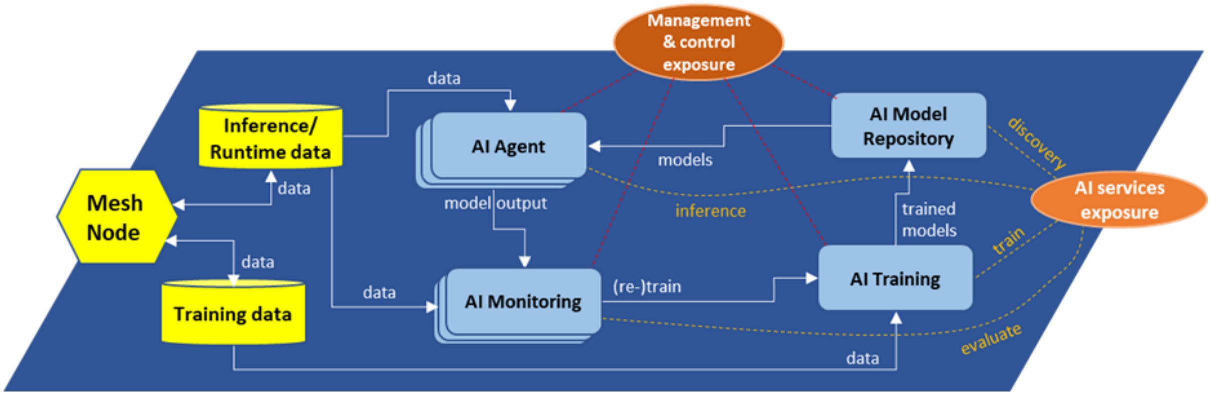


Figure 3-2 AIaaS framework functional decomposition and main interactions.

Three main AI related operational workflows are supported by the proposed AIaaS framework: training, inference, monitoring. In the first workflow, i.e., the training operation, the AI training function can be invoked by an external entity (or possibly internally by the AI monitoring, as described below) through the CAPIF APIs exposing the specific AI training service. Specifically, a new training can be requested by the management and orchestration (e.g., for AI/ML models targeting network automation and closed loops) or even other frameworks. The AI training function executes a training pipeline, using a specific training dataset from the Training Data store. The training pipeline is realized in several steps, which can include data preparation, validation and pre-processing (when needed), the actual training, and a preliminary trained model test and validation. As a result, the AI training stores the new version of the AI/ML model into the AI Model Repository, together with the required metadata and trained model information (e.g., name, version, data requirements, etc.) to facilitate its deployment and use. Different instances of the AI Model Repository may exist within the AIaaS framework, e.g., to enable a clear separation of staging (i.e., for pre-production models testing and verification) and production environments. Moreover, in case of a fully distributed AIaaS framework, spanning across different edge and clouds, dedicated AI Model Repositories per-edge or cloud location could be deployed, to have local and specialized AI/ML models available for specific edge or cloud network automation and optimization goals.

The second operational workflow refers to the AI/ML model inference phase, which is mostly performed within the AI agent. As a pre-requisite of this operation, the AI agent can be first deployed as a cloud-native function by the management and orchestration, and then properly configured to onboard one or more specific AI/ML models and versions, in support of the actual network or service automation and optimization task delegated to the AI agent itself. The AI agent takes care to ingest the inference runtime data into the AI/ML model (or models) and can serve it (or them) following different approaches (e.g., periodic serving or on-demand serving), which mostly depends on the model data requirements (as expressed and stored in the AI Model Repository) and the model objective and scope. The AI agent may also act as a consumer of specific analytics services exposed by the analytics framework, e.g., to access any required analytics output (such as predictions) and complement the given AI/ML model (or models) capabilities. The consumers of the AI agent provided inference service and its outputs can be either the management and orchestration itself (e.g., to implement network services and slices closed loops), or other frameworks belonging to the ECF. In terms of deployment, the AI agent can be dedicated per-service or per-slice (e.g., each using a dedicated service or slice automation AI/ML model), as well as dedicated per edge or cloud location (e.g., using different AI/ML models depending on the inference request).

The third workflow, i.e., the monitoring operation, is enabled by the AI monitoring function. This function is conceived to be deployed by the management and orchestration as a cloud-native function together with the AI agent, thus following the same deployment models described above. Each AI monitoring function is configured at deployment time by the management and orchestration to provide continuous monitoring and evaluation of the AI agent inference results/outputs. To implement its performance evaluation logic (e.g., to measure the accuracy of the model outputs), the AI monitoring function can access the Inference Data store, and perform further data validation and sanity checks (i.e., to identify potential data drifts). Optionally, the AI monitoring function can provide an additional decision logic to trigger automated model re-training to overcome specific model drifts situations. The consumers of the AI monitoring service and its evaluations can be either the management or orchestration, as well as other frameworks belonging to the ECF.

In summary, the AI functions composing the AIaaS framework provide and consume services as summarized in the Table 3-1 below.

Table 3-1 AIaaS framework required and provided services.

AIaaS framework functions	Required services	Provided services
AI model repository		AI/ML model discovery
		AI/ML model storage
		AI/ML model update
AI training	AI/ML model storage	AI/ML model training
	AI/ML model update	
	Training Data ingestion	
AI agent	AI/ML model discovery	AI/ML model inference
	Inference/Runtime Data ingestion	AI/ML model onboard
AI monitoring	AI/ML model inference (e.g., analytics output, inference result, classification, etc.)	AI/ML model performance evaluation
	Inference Data ingestion	AI/ML model re-training decision logic (policy based)

As depicted in Figure 3-2, in addition to the specific AI services offered, the AIaaS framework exposes (following the CAPIF approach) dedicated APIs and interfaces to regulate (e.g., for each AI function) the management and control of the various AI functions, e.g., in terms of deployment into cloud-native virtualized infrastructures, initial (i.e., day-1) and runtime (i.e., day-2) configurations (e.g., onboarding of training pipelines), lifecycle management, etc.

3.1.2.1 Cross-network domain trust integration with CAPIF

In [HEX-D52] a Network Exposure Function (NEF) [29.522] was extended to expose the services offered by an AI framework (see Section 3.1.2) where the NEF is responsible to protect data crossing the domain boundary (cross- Mobile Network Operator (MNO), cross-geographical region). The data is accessed via service APIs and aimed to be used for either AI/ML-model training/ updating purposes or for inferencing purposes (see 3.1.2.2 for further details). In this section the use the Common API framework (CAPIF) as specified in 3GPP TS 23.222 [23.222] for cross-network domain trust integration is elaborated in detail. CAPIF includes common aspects applicable to any northbound service API enabling internal and external exposure (e.g., to a third-party application outside the MNO trusted domain). Furthermore, [23.222] specifies a functional model to support 3rd party API providers as illustrated in Figure 3-3. For the interconnection of different trust domains (e.g., trust domains of different organizations with business relationship) the different trust domains (each with CAPIF deployed) need to interoperate to allow API invokers in each trust domain to utilize service APIs from the CAPIFs. From each CAPIF provider's perspective the other CAPIF provider is a 3rd party requiring authentication with TLS on CAPIF-1e and CAPIF-2e (e.g., based on TLS Pre-Shared Key (TLS-PSK), Public Key Infrastructure (PKI), or OAuth token). To support CAPIF the services offered by the AI framework (see Section 3.1.2) of a specific domain can be exposed as specified in [23.222]. In this solution the AI requesting AF obtains the service API information exposed by AI-EF (AI exposure function provided by NEF) from the CAPIF core function via CAPIF-1/1e reference point (Availability of service APIs event notification or Service Discover Response as specified in [23.222]) and [23.222]). The AI-EF shall support (i) the API exposing function and related APIs over CAPIF-2/2e and CAPIF-3/3e reference point (, (ii) the API publishing function and related APIs over CAPIF-4/4e reference point, and (iii) the API management function and related APIs over CAPIF-5/5e reference point. In a centralized deployment as defined in [23.222], where the CAPIF core function and API provider domain functions are co-located, the interactions between the CAPIF core function and API provider domain functions may be independent of CAPIF-3/3e, CAPIF-4/4e and CAPIF-5/5e reference points.

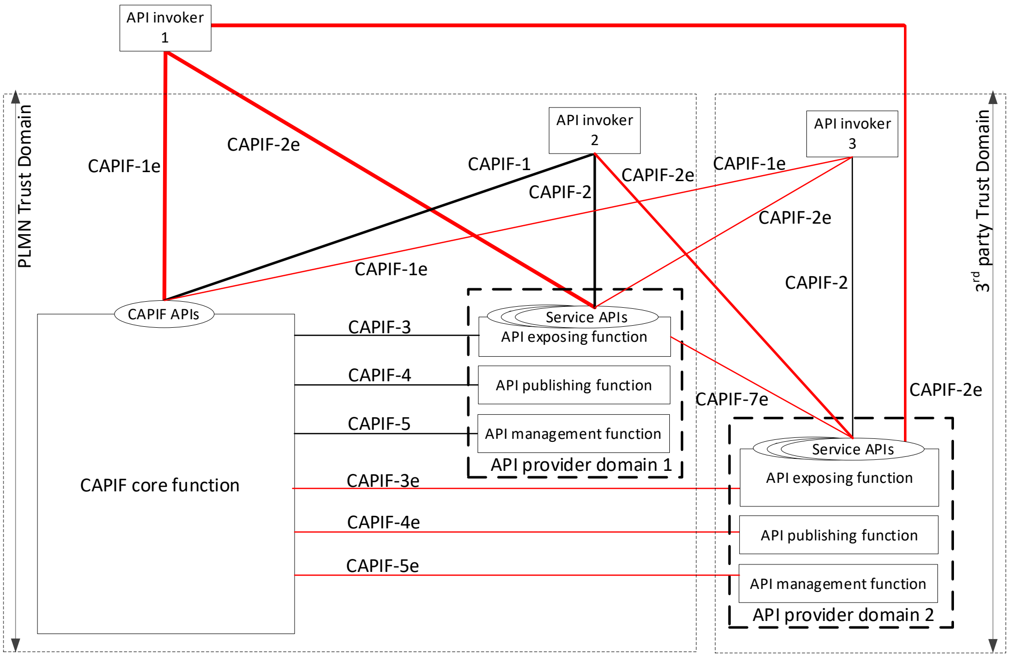


Figure 3-3- Cross-network domain trust integration with CAPIF.

3.1.2.2 Managing cross-network domain trust for in-network learning

While the above section introduces CAPIF to commonly manage AI framework services exposed by a NEF this section addresses possible protection of AI related information when crossing trust domain boundaries.

In Hexa-X delivery D5.2 (see [HEX-D52]) a platform was introduced supporting the management of cross-network domain trust for in-network learning. The proposed platform enables fine-grained, privacy-preserving user (or any other data-contributing entity) data Life-Cycle Management (LCM) across security domains of a network (e.g., cross- Mobile Network Operator (MNO), cross-geographical region), where the data is aimed to be used for either AI/ML-model training/ updating purposes or for inferencing purposes. In the following, interactions between AI frameworks of different trust domains are illustrated where the exchange of information (Training data, AI ML models, computation data,..) is protected by a NEF at the boundary of each trust domain:

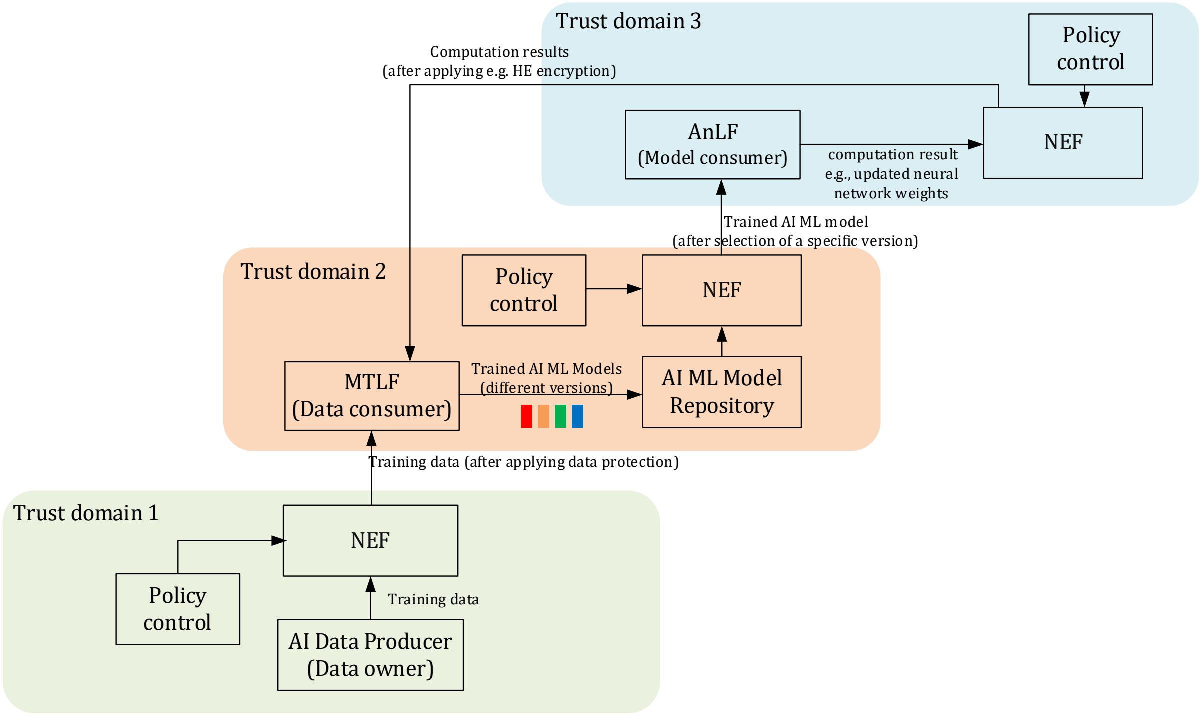

Trust domain 3
Policy control
Computation results (after applying e.g. HE encryption)
AnLF (Model consumer)
computation result e.g., updated neural network weights
NEF
Trained AI ML model (after selection of a specific version)
Trust domain 2
Policy control
NEF
MTLF (Data consumer)
Trained AI ML Models (different versions)
AI ML Model Repository
Training data (after applying data protection)
Trust domain 1
NEF
Policy control
Training data
AI Data Producer (Data owner)

Figure 3-4 Interactions between AI frameworks of different trust domains.

In Figure 3-4 data protection is applied by NEF when data is exposed to a NF of different trust domain than the producing NF according to policy received from policy control. Some of the privacy preserving techniques that can be used are mentioned below:

- training data protection, e.g., using differential privacy (local DP), anonymization, pseudonymization, encryption (e.g., homomorphic encryption (HE))
- AI ML Model protection, e.g., by generating different AI ML model versions using, e.g., Differentially-Private Stochastic Gradient Descent (DP-SGD). DP-SGD can be used to add noise to the clipped gradient during SGD training to prevent training-data privacy breaches [ACG+16].
- Computation result (e.g., updated neural network weights when participating in FL) protection, e.g., by applying HE.

The NEF is responsible for enforcing a protection scheme as provided by policy control which is based on the trust level of the receiving domain and its regulations. For example, jurisdictions may have regulations that govern the use of AI/ML models in certain industries, such as healthcare or finance.

3.1.2.3 In-network AI system architecture addressing requirements of the EU AI regulation

The European Commission is supporting the development of a European Artificial Intelligence Act (AI Act) by the European Parliament and Council. A corresponding draft proposal has been made available in 2021 [AIAct+21]. In [HEX-D52], a proposed system architecture was derived in accordance with the requirements of the draft AI Act. In the present section, we further build on this architecture proposal and derive interactions between the inherent entities with the objective to meet requirements laid out in the draft AI Act. We propose a new functional architecture, in which the entities introduced in [HEX-D52] (Annex A.2, Figure 9-1), are being interconnected through a bus as in Figure 3-5. Those functions can be mapped to the AIaaS provided services as specified in Section 3.1.2 and to the management plane.

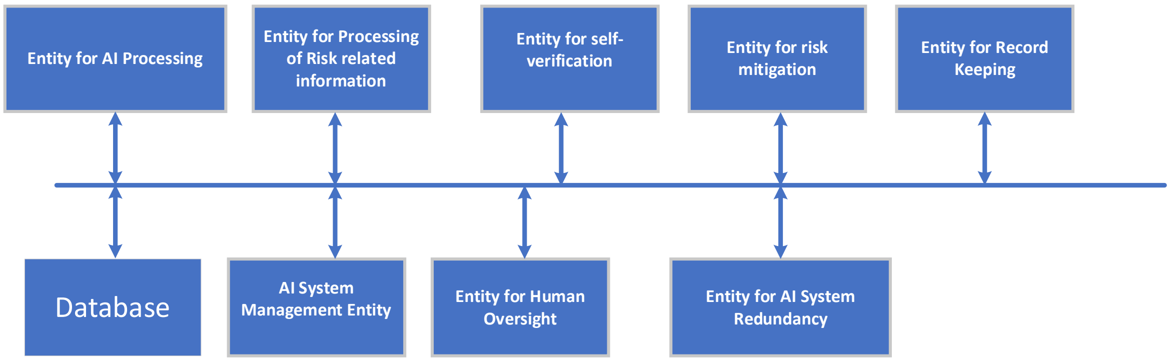

Entity for AI Processing
Entity for Processing of Risk related information
Entity for self-verification
Entity for risk mitigation
Entity for Record Keeping
Database
AI System Management Entity
Entity for Human Oversight
Entity for AI System Redundancy

Figure 3-5 Functional Architecture interconnecting entities of AI System Architecture through a service bus.

The functional entities illustrated in Figure 3-5 have the following functionality:

- **AI Processing** is the core of the AI system, i.e. After proper training some AI decision making process is performed, e.g., Pattern detection based on a Neural Network approach or similar.
- **Processing of Risk related information** presents the information to authorized User(s) for identifying the correct operation of the AI system.
- **Self-verification** is responsible for verifying the operation of the AI system against criteria set out in the AI Regulation, including identification of eventual biases, verification of training data, etc.
- **Risk mitigation** offers trade-offs to the user to choose from, e.g., Functionality/risk trade-off (i.e., offer less (more) functionalities implying less (more) risks, etc.).
- **Record keeping** is logging of the user activity, the behaviour of the AI system, information on re-training of the AI system, etc.
- **Database** is responsible for storing reference training data, logging of user activity, logging of AI system behaviour, etc.
- **AI System Management** orchestrates AI system internal processes, e.g. When a user requests information on AI system behaviour or similar, the information is recovered from database, processed, and presented to User, etc.
- **Human Oversight** identifies information that may be relevant for authorized users to intervene, e.g., Stop the operation of the AI system because biases are observed or similar.
- **AI System Redundancy** responsible to replace critical entities of AI System in case of failures (e.g., stop operating or operate erroneously) by redundant (replacement) entities.

The interactions between the various entities of the AI Architecture depend on the specific use cases. Key examples are given below:

3.1.2.3.1 Interactions of entities of AI Architecture: Avoidance of undesired biases

A key requirement of the draft AI Act relates to the avoidance of undesired biases. The problem statement is obvious in applications such as biometric identification of physical persons: The applicable AI model must rely on training data that is representative and includes sufficient references to physical persons of various characteristics and origins. Specifically, it is not acceptable to base the training of an AI model exclusively on a small number of persons belonging to one ethnic group (or a small number of distinct ethnic groups). Rather, a large representative training data set needs to be applied that appropriately addresses the target audience.

Once the AI system is trained, its database is typically further extended by using ongoing observations. It is obvious that care must be taken to avoid that any undesired biases are being developed. For this purpose, an interaction between the following entities of the AI System is applied as illustrated Figure 3-6. The “Entity for self-verification” is constantly monitoring the AI system and is verifying whether the update of the AI model may lead to undesired biases (for example through applying validation example data sets in order to verify that all ethnic groups are handled equally); the corresponding assessment is communicated by the “Entity for self-verification” to the “Entity for Human Oversight”.

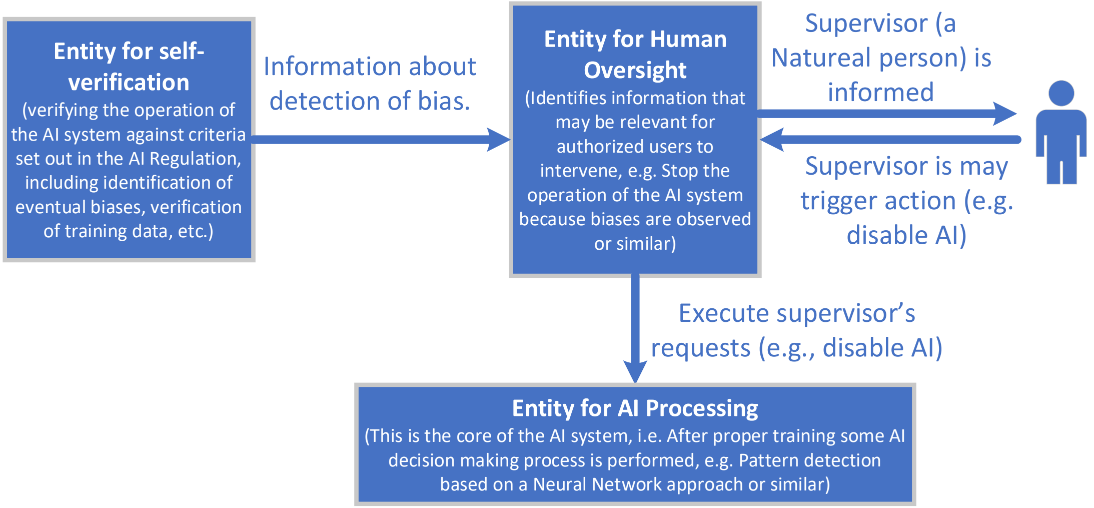

Entity for self-verification
(verifying the operation of the AI system against criteria set out in the AI Regulation, including identification of eventual biases, verification of training data, etc.)
Information about detection of bias.
Entity for Human Oversight
(Identifies information that may be relevant for authorized users to intervene, e.g. Stop the operation of the AI system because biases are observed or similar)
Supervisor (a Natureal person) is informed
Supervisor is may trigger action (e.g. disable AI)
Execute supervisor’s requests (e.g., disable AI)
Entity for AI Processing
(This is the core of the AI system, i.e. After proper training some AI decision making process is performed, e.g. Pattern detection based on a Neural Network approach or similar)

Figure 3-6 Interactions of entities of AI Architecture – Avoidance of undesired biases.

The “Entity for Human Oversight” is extracting the most relevant information and makes it available to a human supervisor (typically through appropriate graphical representation, indication of the most critical numerical metric (for example, the detection probability for various ethnic groups, etc.), etc. The human supervisor will then decide whether any specific action needs to be taken, for example the retraining of the AI System, fallback to an initial AI model, fallback to a previous AI model before any undesired bias was detected, etc. The human supervisor is communicating his decisions and actions to the “Entity for Human Oversight” which is then initiating the corresponding actions through interaction with the relevant entities, typically with the “Entity for AI processing” (and possibly further entities of the AI System).

3.1.2.3.2 Interactions of entities of AI Architecture: Management of AI System Redundancy

Critical components / entities of an AI System require redundancy, i.e., in case that a critical component / entity is failing or misbehaving (i.e., not behaving according to expectation for example leading to physical harm of persons, violating fundamental rights or EU values, etc.), the AI System must be able to replace that failing component / entity.

The basic process is as follows: The “Entity for self-verification” is constantly monitoring the AI system and is verifying whether all components / entities are operating appropriately; specific attention is given to “critical” components / entities (i.e., the verification frequency is higher for critical systems compared to other components / entities, the correct operation is verified in greater depths by observing a larger number of metrics over a longer period of time compared to other components / entities). In case that an issue is identified for a “critical” component / entity, the “Entity for AI System Redundancy” is triggered and a replacement of the failing / misbehaving component / entity is initiated. The “Entity for AI System Redundancy” is interacting with any other required AI System entity in order to reconfigure the system to take the new configuration into account.

Example (i) and (ii) above illustrate how appropriate interactions of AI System entities can address the requirements expressed by the draft AI Act. Future research is required to identify an exhaustive set of such use cases and related interactions between the entities in order to ensure that all essential requirements of the finally adopted AI Act are being met.

3.1.3 Programmable management

The ETSI Zero Touch Network and Service Management (ZSM) Industry Specification Group specifications [ZSM] concerns many aspects of network management evolution. ZSM requirements concern, among others, the capability of automatic installation of management software, mechanisms for detection and conflict resolution between different closed loops inside a domain and in other domains, nesting of closed loops, and the ability of the network owner to disable any automation function in case of malfunction. Some implementation-related issues of Control Loops (CL) based management have already been described in [HEX-D52]. This section will present a framework that enables programmability of management, i.e., adding or removing CL-based management. The proposed idea can be seen as enhancements of concepts presented in deliverables [HEX-D62] and [HEX-D51] of the Hexa-X project as well as a generalisation and extension of a concept described in [Kuk22a]. The mentioned paper concerns distributed management architecture of networks, but the idea is also applicable in a generic case. In almost all CL-based approaches, each management function has a specific goal that is realised independently of other goals. A single CL pipeline is typically composed of the Monitoring Subsystem (MOS), Analytic Engine (AE), Decision Element (DE) and the Reconfiguration Engine (RE). The conflicts lie in the reconfiguration of the same parameters by different CLs or as indirect impact of modified network/service configuration on the overall network Key Performance Indicators (KPIs). The multi-objective optimisation algorithms are not in use due to a lack of scalability. Moreover, such an approach is not applicable in a system in which Monitoring-Analysis-Planning-Execution (MAPE) driven management functions are dynamically added. Coordination of multiple management functions has been proven to be difficult, and only partial solutions based on priorities or time separation between different management functions are used in practice. The problem has been described in [BAK+21]. In the case of Long Term Evolution (LTE) SON, RAN-specific solutions have been proposed. The reconfiguration problem is not the only problem of cooperation of multiple management functions. As it has been outlined in [HEX-D52] and [Kuk22a], the monitoring and network state analysis should be aware of the activity of management functions as the ongoing or completed reconfigurations impact monitoring. For example, the ongoing network reconfiguration process should not be treated as an anomaly by AEs. The presented concept follows [HEX-D52] and [Kuk22a] ideas and assumes that functionality of the Management Platform (MP) of a Management Domain (MD) can be decomposed into a set of Management Services (MS). To that end, following the Fault, Configuration, Accounting, Performance, Security (FCAPS) approach, it is proposed to split the MS set into Fault Management Services (F-MS), Performance Management Services (P-MS) and Security Management Services (S-MS). The following reconfiguration mechanisms are proposed to be enforced by the DEs of an MS:

- **Configuration Parameters Modification (CPM) -** a classical management operation. Changes in parameters can impact, for example, routing, radio coverage or handover execution and do not involve an orchestrator. The MD topology typically is not changed, but users' traffic matrix or assignment to RAN nodes can be impacted. In most cases MOS entities should be aware about reconfigurations.
- **Resource Scaling (RS) -** is an orchestrator-based operation that concerns the modification of resources allocated to software-based functions and links. This mechanism can be implemented in a proactive data-driven way (Proactive RS – PRS) or a reactive (Reactive RS-RRS) one. In the RS-PRS case, the decision on resource allocation is data-driven (based on traffic trends and the number of users' change), i.e., before the change in resource consumption is noticed. RRS is based on resource consumption trends. The RS-RRS procedure does not interfere with other management operations.
- **Topology Modification (TM)** operation is typically driven by an orchestrator and includes a VNF migration, VNF addition, VNF removal and addition or removal of links. The decision about the topology change is typically based on the analysis of the present NS topology, the matrix of traffic flows and the load of each VNF. It can be enforced for performance improvement, fault management or attack mitigation procedures. The topology change requires traffic redirection and impacts monitoring that must adapt accordingly.

The architecture of the proposed solution is presented in Figure 3-7. It is composed of multiple programmable Management Services (MSs) and two components that are common for all MSs, namely the Monitoring Subsystem (MOS) and the Reconfiguration Engine (RE). Both components are involved in the reduction of the mutual impact of MSs. All MP components interact via a message bus.

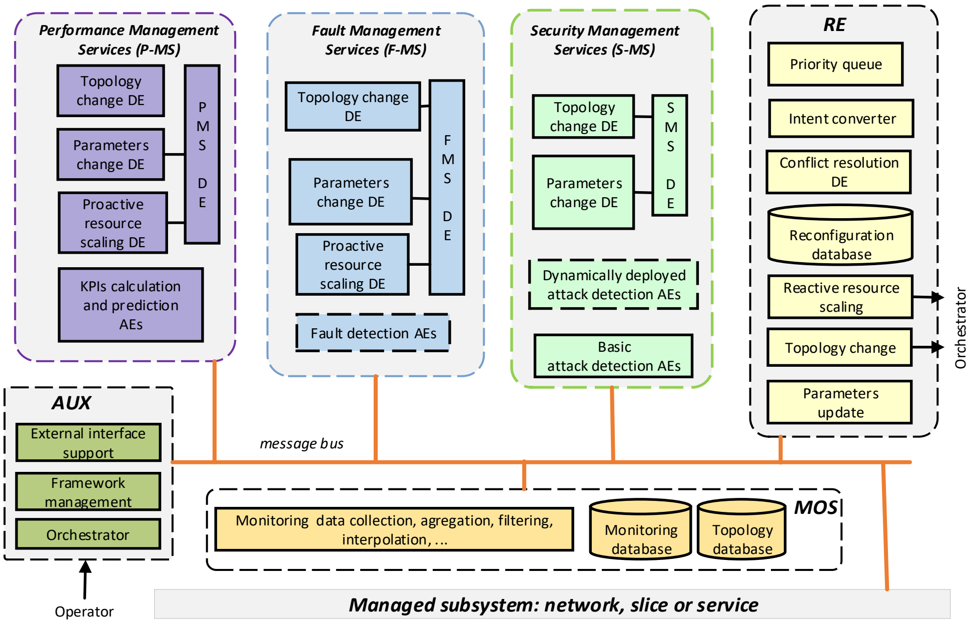

Performance Management Services (P-MS)
Topology change DE
Parameters change DE
Proactive resource scaling DE
KPIs calculation and prediction AEs
P MS DE
Fault Management Services (F-MS)
Topology change DE
Parameters change DE
Proactive resource scaling DE
Fault detection AEs
F MS DE
Security Management Services (S-MS)
Topology change DE
Parameters change DE
S MS DE
Dynamically deployed attack detection AEs
Basic attack detection AEs
RE
Priority queue
Intent converter
Conflict resolution DE
Reconfiguration database
Reactive resource scaling
Topology change
Parameters update
Orchestrator
AUX
External interface support
Framework management
Orchestrator
Operator
message bus
Monitoring data collection, agregation, filtering, interpolation, ...
Monitoring database
Topology database
MOS
Managed subsystem: network, slice or service

Figure 3-7 Architecture of the programmable management framework.

The role of the mentioned components is the following:

- **Monitoring Subsystem (MOS)** provides information about the network/service status and is involved in KPIs and accounting data calculation. It collects information from all sources (VNFs, Infrastructure) and processes the received data. The topology-dependent data processing includes data aggregation, filtering, interpolation, and prediction. The MOS has a Monitoring Database (MDB) and a Topology Database (TDB). Its services, to a certain extent, is customisable (data sample rate, etc.). The MOS data are consumed by MSs but also by REs to verify the results of reconfiguration and to take autonomous decisions about RS-RRS. The MS obtains from RE information concerning reconfigurations (including reconfigurations in progress) to invalidate the monitoring data when the reconfiguration process is in progress or to adapt monitoring to topology changes.
- **Reconfiguration Engine (RE)** is responsible for enforcing reconfigurations triggered by MSs and provides resolution of conflicts. Each of the reconfiguration requests is stored in the priority queue. The RE is responsible for conversion of intents into a set of ordered, atomic commands. The RE is also responsible for reactive resource scaling (RS-RRS). Each old and new reconfiguration is stored in the Reconfiguration Database (RDB) of the RE. During the reconfiguration process, the MOS and all MSs are informed that a reconfiguration is in progress, as it may cause a change in the monitoring values and how they are calculated and interpreted by AEs. One of the critical functionalities of the RE is conflict resolution between MSs which can be handled in several ways, like time scale separation, priorities or by AI. The RE may also use multi-objective optimisation algorithms to solve conflicts between MSs.
- **Auxiliary Functions (AUX)** are functions responsible for framework management and orchestration and provide an interface to the system operator. The description of these functions is out of the scope of this section.

As previously mentioned, all MSs are integrated into F-MS, P-MS or S-MS groups. Each of the groups may have multiple AEs and DEs, but only a single Master DE atop other DEs. The DEs and AEs can be AI-driven. The internal structure of P-MS, F-MS or S-MS is left to implementer as in most cases it has negligible impact on MS coordination issue.

- **Performance Management Services (P-MS):** Performance optimisation goals are expressed in the form of (Key) Performance Indicators (PIs, KPIs). PIs can be not only monitored but also predicted, which gives more time for reconfiguration. The P-MS-triggered reconfigurations are typically done in small steps. The first mechanism triggered by P-MS is typically the RS (RS-PRS or RS-RRS). The CPM is the next reconfiguration option, allowing traffic redirection to achieve load balancing, etc. Finally, if a need for topology change has been discovered, TM can be used.
- **Fault Management Services (F-MS):** Fault management is typically composed of three phases: fault detection, identification of the fault source and fault mitigation. Faults can be directly signalled by alerts or can be detected by AEs. The alerts have to be handled quickly, but the AE-driven fault detection gives more time for handling. Two types of reconfigurations can be used for fault mitigation, CPM and TM. The CPM may include traffic redirection, whereas the TM allows for adding functions or links that replace the faulty ones. Even during a fault, the P-MS tries to optimise the performance; however, it should be notified about faulty resources and functions.
- **Security Management Services (S-MS):** The S-MS detects security threats, finds details of the attack and, finally, provides mitigation. The S-MS behaviour is similar to anomaly-based fault management. The S-MS may use dedicated AEs to detection threats and using TM a dedicated S-MS function (probe, firewall) can be added. After the detection of the attack, the MS entities should be informed to avoid competitive reconfigurations. The attack mitigation may include blocking flows, traffic redirection, isolation of functions or nodes, reduction of used resources, etc. In the case of the detected attack the P-MS should stop allocating more resources to the malicious nodes, and the F-MS should not interpret the blocking of some flows as a fault.

Table 3-2 shows the overall impact of reconfigurations and events (fault, security attacks) on the MOS and X-MSs (P-MS/F-MS/S-MS). Please note, that some reconfigurations affect almost all X-MSs.

Table 3-2 Impact of management events on framework entities.

Event	**Event triggering entity**	**Impact on MOS**	**Impact on RE**	**Impact or P-MS reaction**	**Impact or F-MS reaction**	**Impact or S-MS reaction**
Reconfiguration in progress	RE	Monitoring data marked	RDB updated when finished	AEs notified, optionally DEs notified	AEs notified, optionally DEs notified	AEs notified, optionally DEs notified
Fault detected	F-MS		Stopping new reconfigurations other than F-MS		Reconfiguration expected	
Attack detected	S-MS		Stopping new, P-MS driven reconfigurations		AEs notified, optionally DEs notified	Reconfiguration expected
Resource Scaling	REP P-MS	None	RDB updated	None	None	None
	P-MS	Flows to paths update			AEs notified	AEs notified

Configuration Parameters Modification	F-MS			AEs notified, optionally DEs notified	None	AEs notified
	S-MS				AEs notified, optionally DEs notified	None
Topology modifications	P-MS	TDB and MDB modified				AEs notified, optionally DEs notified
	F-MS					
	S-MS					

The presented framework is under implementation now. Three mechanisms are planned to be deployed, priority-based approach, recommender based on the game theoretical model and AI-based solutions that analyse the reconfiguration history to approve or reject a proposed reconfiguration.

3.1.4 Programmability framework

The support of programmability in the network's infrastructure enables the flexible reconfiguration of the behaviour of this infrastructure over time. The packet and information processing part of network infrastructure in a cloud environment is made up of a heterogeneous pool of computing resources enabling both software and hardware-based programmability. Network and UE programmability can be leveraged by the management plane so that it can flexibly control and reconfigure the behaviour of the network and UE based on different smart orchestration solutions. However, as this programmability spans different domains in the network, it is important to well design the management of this feature.

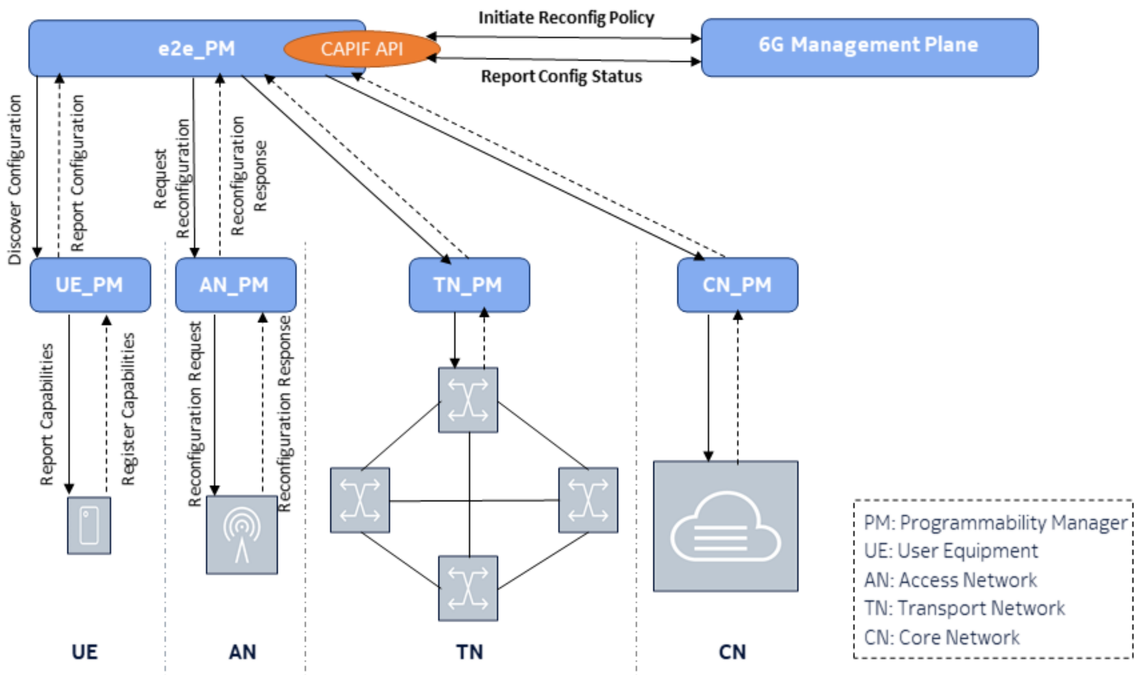


Figure 3-8 Hierarchical framework for E2E programmability management.

Figure 3-8 shows a proposed hierarchical framework for end-to-end programmability management and orchestration. This framework defines local programmability managers for the different domains: User Equipment (UE), access network, transport network, and core network. These distinct local managers are needed to deal with the different infrastructure technologies used in the different domains. For example, in [HEX-D52], we proposed a performance-aware orchestration framework that takes care of placing different NFs' workloads into P4 programmable cloud environments. This could be among the tasks of the local programmability managers in the access, transport, and core networks. The local programmability managers should also take care of different generic tasks such as:

- Discovering the programming capabilities of the underlying infrastructure: supported programming features, provisioned possible programs/realizations, supported programming language, etc.
- Abstracting the devices' details to enable easy usage of the programmable devices without worrying about the configuration details related to the devices' drivers, compilers, etc.
- Defining APIs to specify how to interact with the different programmable devices.
- Establishing the channels to reconfigure the behaviour of the programmable devices when needed.

The different local managers should be able to interact with one centralized entity that takes care of the e2e programmability of the network infrastructure in a hierarchical and harmonized way. This entity is called "e2e_PM" as shown in Figure 3-8. On the southbound interface, the role of this entity is to push reconfiguration commands to the different local programmability managers or to retrieve the configuration (behaviour) running on the programmable devices. On the northbound interface, this entity interacts via the CAPIF API with the other different 6G management functions and frameworks to enable exposing the status of the running configuration of the programmable elements to these frameworks and to offer the other frameworks the flexibility in triggering reconfiguration processes to change the behaviour of the network elements when needed. For example, the analytics framework can dynamically change the metered network metrics and features reported by the different network elements through consuming the "Initiate Reconfiguration" service offered by the e2e_PM entity, which takes care of changing the behaviour of a network element to report the newly requested data. The table below shows the different services required and offered by the local and e2e programmability managers involved in the programmability framework.

Table 3-3 The programmability framework's required and offered services.

Programmability Framework Entities	Required Services	Offered Services
UE_PM, AN_PM, TN_PM, CN_PM	Capabilities Reporting	Capabilities Registration
		Reconfiguration Initiation
		Configuration Reporting
e2e_PM	Configuration Discovery	Configuration Reporting
	Reconfiguration Initiation	Reconfiguration Policy Reporting

3.1.4.1 UE Programmability

In this subsection, a more detailed illustration related to managing the UE programmability is provided. A high-level concept including the fundamental components, challenges and benefits of UE programmability has been discussed in [HEX-D52]. A proposal on UE programmability architecture is illustrated in Figure 3-9.

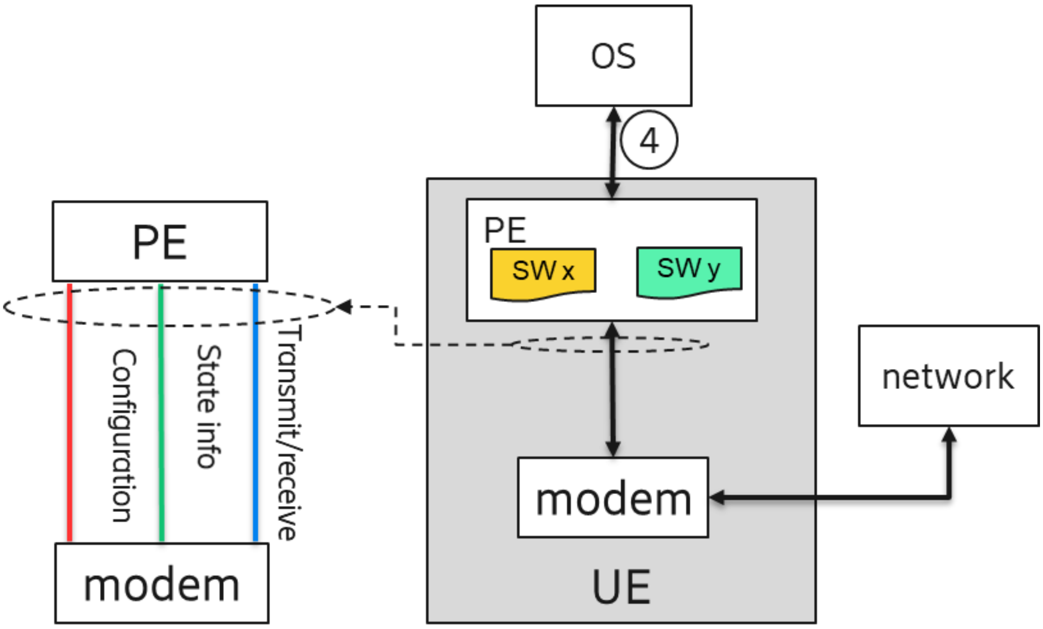


Figure 3-9 Programmable UE architecture.

In the proposed architecture the UE is composed of two entities. One is the modem which includes all the components in a legacy non-programmable UE with all the control and user plane protocol stacks components. The other entity is defined as programmability environment (PE) that is responsible for receiving, executing, and managing the software (SW) obtained from the network.

These two entities communicate with each other through a programmability interface (PI). The PI provides the following services to realize the concept.

First it provides a means for the PE and network to communicate with each other. The network needs to be able to deliver SW related information such as those related to management for example. Another possibility, among others, is to alter the behaviour of a given SW by sending relative arguments dynamically. On the reverse order also, the PE needs to be able to communicate with the network to convey SW related information. The modem facilitates this by realizing a communication medium between the PE and the network. One realization of this is to define a specific radio bearer (e.g., programmability radio bearer) that handles such communication.

The SWs can request specific information regarding the internal states of the functions and protocols for optimizing the configuration (shown as “state information” in Figure 3-9). Note that this information is directly requested from the modem itself. For example, a mobility management SW can request the modem to provide the Reference Signal Received Power (RSRP) measurements on specific cells as part of the SW execution.

Finally, there is a configuration interface that allows the SW to configure the modem. One realization of this is that the SW can generate RRCReconfiguration messages and deliver them to the modem and the modem would be configured as though it has received the message directly from the network.

The architecture is simple in the sense that the programmability concept does not impact the legacy stack since it reuses the air interface protocol and interacts with it only through the SW installed on the PE. However, this unleashes an important capability which is to reconfigure the UE with SW that can be installed on the fly. We will motivate this with an example use case in Section 3.2.6.

3.2 Applications and evaluation of intelligent network

This section describes various applications, use cases and evaluations of the intelligent network.

3.2.1 AIaaS framework AI functions for MLOps

The AI functions building the AIaaS framework, as described in Section 3.1.2, are conceived to be implemented as cloud-native applications, allowing therefore their virtualization and packaging to enable their deployment as in-network virtualized AI functions in the cloud continuum.

An initial implementation of the AI functions composing the AIaaS framework has been carried out in support of the Hexa-X Management and Orchestration Demo #5 Scenario 4 described in [HEX-D63]. Here, the AI functions are implemented and deployed as cloud-native functions to support an MLOps scenario, i.e., to showcase the combined use of DevOps and AI/ML techniques to introduce automation, pipelining, monitoring, and packaging in the AI/ML models lifecycle (i.e., from development to deployment into production). Specifically, this scenario addresses an AIaaS implementation that integrates AI in network management and orchestration with the aim to avoid network slice and service performance degradations caused by limited 5G UPF resources at the edge, while reducing data flow demands for training and inference. An AI agent serves for the optimal auto-scaling of local 5G UPFs placed at the network edge, in support of low latency communication services, with applicability to 6G use cases which depend on reliable low latency communications, including those under the umbrella of Massive Twinning and Robot to Collaborative Robots (Cobots) use case families [HEX-D12].

Beyond the details of such a demonstration scenario, which can be found in [HEX-D63], Figure 3-10 shows a set of candidate tools for the implementation of some of the AIaaS framework AI functions.

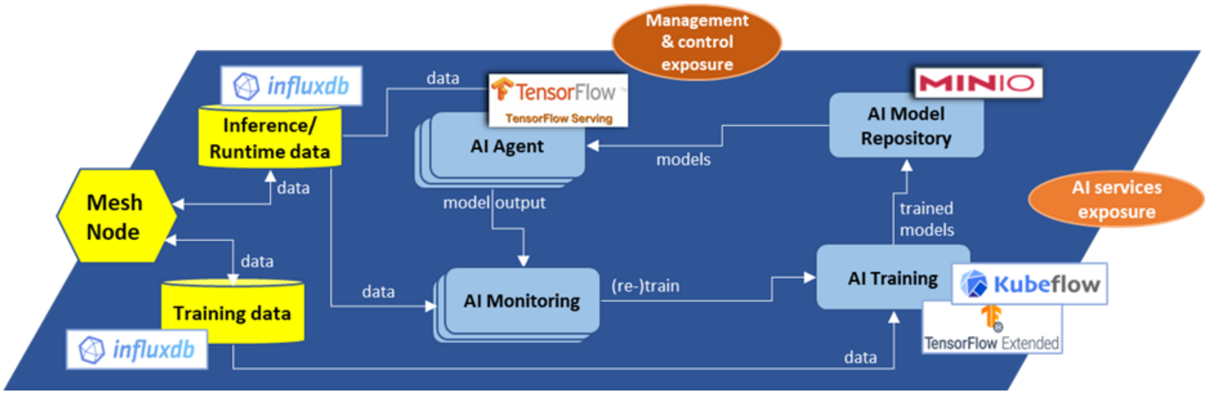


Figure 3-10 Candidate tools for AI functions implementation.

Specifically, these tools have been successfully integrated, validated and showcased in the context of the MLOps demo. The AI training function can be implemented using Kubeflow [KUB], a scalable ML platform that runs on Kubernetes and exploits all its capabilities to facilitate the development, deployment and operations of ML pipelines in virtualized environments. It provides means to flexibly pre-process data and train various models in cloud-native containerized environments, in a portable and scalable way. With Kubeflow as AI training function, TensorFlow Extended [TFX] can be also used to develop ML models and pipelines for scalable, high-performance ML tasks. The AI model repository can be implemented leveraging on MinIO, a multi-object cloud storage which offers a cloud-native solution suitable for highly distributed scenarios [MIN], together with a wide set of SDKs to ease the interaction with the stored data. The AI agent can leverage on TensorFlow Serving, which provides a flexible and high-performance serving system for ML models [TFS], facilitating the AI agent in onboarding different models and versions in cloud-native environments. For what concern the Inference/Runtime and Training data stores, InfluxDB can be used as time series data platform to enable cloud-native analytics and data manipulation applications [INFL].

3.2.2 Distributed AI services

The high-level architecture integrating AI to enable Interacting and Cooperating Robots use case proposed in [Hexa-X D5.2] is mapped to an example scenario called predictive quality of service (pQoS) in interacting and collaborative robots. An architecture is proposed as seen in Figure 3-11 for the demonstration and validation of the high-level architecture integrating AI to enable Interacting and Cooperating Robots use case.

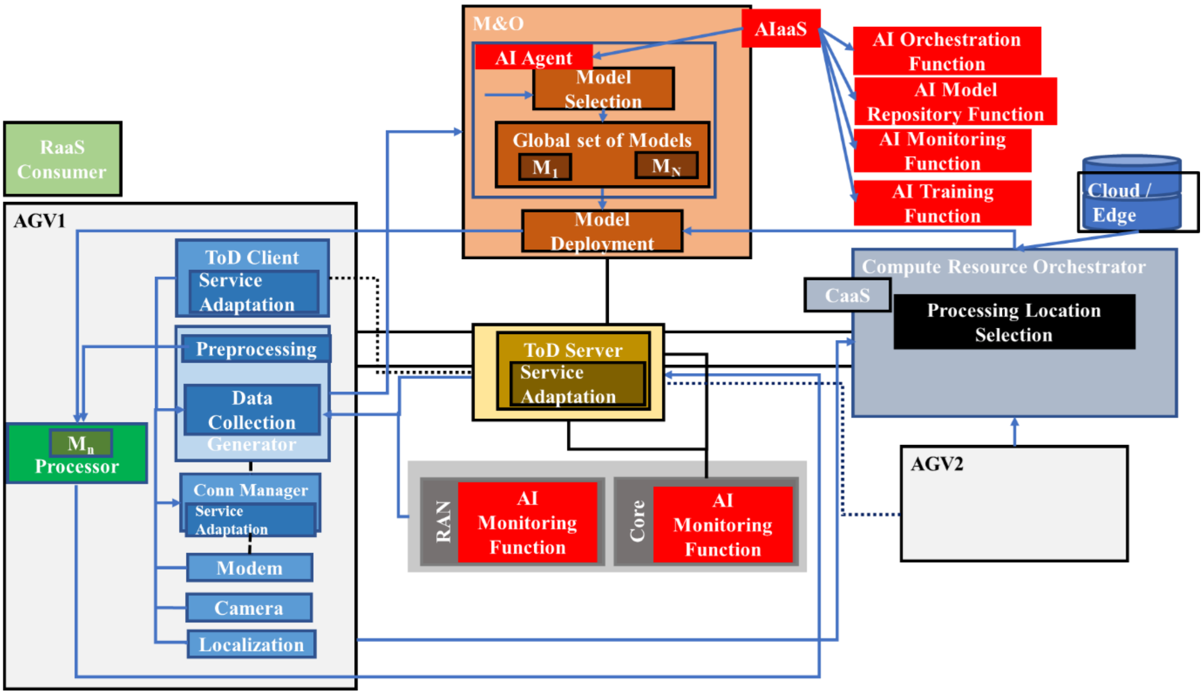


Figure 3-11 Architecture for realizing Predictive Quality of Service (pQoS) in interacting and collaborative robots use case.

The objective is to predict QoS and determine the motion pattern of a cobot, which is a Robotic-as-a-Service (RaaS) consumer, using a proactive approach in robotics scenarios. Movement patterns of cobots refer to, for example, while traveling to the target at a certain speed, and in a determined direction, a cobot may observe a decrease in the QoS value at the time t+Δ t with the help of an ML model, and this decrease may be due to shadowing. Objects that cause the shadowing effect can be a wall, a robot, or a human. Therefore, in order not to cause any accidents or to reach its target, with the help of the data and observations gathered from other cobots, humans, and the network, a cobot can halt or delay operations, re-assign tasks, or choose different movement trajectories to obtain the desired pQoS value at t+Δt time. This approach can be used by other cobots and people in the system to ensure worker safety, evade obstacles, and make the system run seamlessly.

In order not to cause any accidents and to reach its target, there are components and functionalities that help with the data and observations gathered from other cobots, humans, and the network.

Data Collection: The data collection functionality works as a gatherer of information to be used afterward. It collects data from

- Modems that allow short-distance device-to-device communication and network-based communication
- Visible light spectrum cameras, infrared cameras, and Lidar or Radar
- Global Navigation Satellite System (GNSS) patches
- The Teleoperated Driving (ToD) client, which is a local part of the ToD, manages all the information needed for the service and all the activities with the actuators in the cobots (direction, brake, accelerator, etc.).
- ToD-server, which is a remote part of the ToD from which the cobot is controlled. It shows all the needed information to the remote driver and sends its instructions back to the cobot. It sends the most recent information about the system participants to the data collection functionality. The server adaptation modules of all cobots in the system provide information to the ToD server.

- Also, the Radio Access network and Core network send analytics to the data generator.
- Connectivity Manager (Conn Manager): The application's service adaptation relates to the management of various communication links. Based on the pQoS information, a link selection, or the use of multi-connectivity, in which numerous connections are utilized concurrently, can be triggered.

After the collection of the data, pre-processing is done on the data before it is fed into the AI model. Before being fed into the model, the data is sent to an AI Orchestrator, which refers to AI-as-a-Service functionality, for training. The AI Orchestrator manages and deploys the models. This includes updating the models with updates received from cobots. The AI Orchestrator also directs the cobot to an X cloud server (cloud server IP and ID) to get the desired service and informs the cobot about the period for which the data is collected, the channel, the security settings, and the data the cobot should share. The AI Orchestrator is in charge of training and has access to a large number of Edge and Cloud servers. The AI Orchestrator keeps track of the training until it is completed. It monitors at what stage the training the desired level of success has been reached, and then if there is a bias, or feature optimization or selection, or hyperparameter selection or optimization, is required. Following this, the training locations are selected by the compute resource orchestrator, which refers to compute-as-a-service (CaaS). The compute resource orchestrator gathers the compute resource capabilities of all the available nodes. It decides on the location based on compute resources, task completion time, and trustworthiness metrics. If the server assigned during the training process crashes or becomes overloaded, it continues the training on a different server. It monitors information such as energy consumed, process resources, time, and data overhead.

Model selection takes place at the AI orchestrator. If several models have been implemented to suit different scenarios, the orchestrator evaluates the data received and decides accordingly if a change in the current model must be deployed. Assuming that the model selection is successful, it must be validated. If it is validated, a model is selected from the global set of models. In the model deployment, a suitable model for a cobot is ready for deployment based on available data in the cobot. The compute resource orchestrator shares the optimal compute node selected by the processing location information function (considering the parameters of computing resources, energy efficiency, and trustworthiness) with the AI orchestrator model deployment. The model deployment continues with the selected model and the optimal node, which is the cobot itself in this example. The selected model, the model to predict QoS, is processed (inferences are made) in the data processor of the cobot. The data from pre-processing is used by the ML model selected and deployed by the AI orchestrator. The inferences regarding the QoS parameters are sent to the ToD server. It decides if the pQoS is sufficient for the service or, if not, what measurements must be taken. Those measurements can be the reduction of speed, diverging the route to possible alternative ways, or, in the case of a drastic decrease below the security threshold, an emergency stop.

In cases where pQoS is shared, it is evaluated within the AI orchestrator, RAN, and CN. The evaluation aspects are as follows:

1. The AI Orchestrator checks if the ML model estimates the pQoS value within the determined thresholds and the cobot acts accordingly, or if the pQoS is not found in the correct estimation range, causing the cobot to lose time, make faulty manoeuvres, and wrong actions. As a result, model selection and deployment are redone.

2. RAN and Core: Resource allocation is crucial. The performance of selected ML model functions (RAN and core communication), such as availability, accessibility, mobility, and so on, is being monitored. It can be used for the overall management of AI and conflict handling.

3.2.3 Evaluation of the FLaaS framework

In 6G networks UEs will be able to exploit AI models for improved intelligent services. With Federated Learning (FL), UEs can also collaborate in building those models: this brings the benefits of potentially producing more accurate AI models while preserving UE data privacy. The envisaged FL as a Service (FLaaS) framework – described in [HEX-D52] – provides functions and protocols to allow UEs to discover federation of UEs, in order to exploit collaborative AI models and participate in their training. The aim of this section is to evaluate the learning process times in the FLaaS framework. In FLaaS, multiple FL Local Managers (FLMs) share locally-trained AI models with the FL Process Computation Engine (FPCE), which in turn aggregates them into a global AI model. The objective of the analysis is to assess the time it takes for the FPCE to retrieve all the local models from the FLMs under different network load conditions and different deployments of FLaaS functions. This evaluation considers both the communication and the computation aspects of the proposed framework and is performed via system-level simulations carried out with Simu5G [NSS+20]. All the entities involved in the FLaaS framework have been implemented within Simu5G as ETSI Multi-access Edge Computing (MEC) applications running on a MEC host. Among them, the FLM can be deployed at either the MEC host or at the UE. This choice allows us to evaluate the framework when model exchanges occur within the MEC host and via the RAN (under different network conditions), respectively.

With reference to Figure 3-12, we simulated a scenario composed of seven gNBs, deployed in a regular hexagonal grid and surrounded by a second tier of interfering cells. The inter-gNB distance is 500 meters. We assume that UEs participating in the training are served by the seven central gNBs (we refer to them as *foreground* UEs), and they suffer interference from *background* UEs (not shown in the figure) served by the interfering cells. The network is assisted by one MEC host. At the beginning of the simulation, UEs deploy their FLM as either a local or MEC application, depending on the deployment under evaluation. Then, the FLMs notify the FL Service Provider (FSP) about which FL service they want to take part in (for example, the QoS forecasting service). In turn, the FSP authorizes the FLMs to contact the FL Process Controller (FPC) handling the FL process requested by the FLMs. As soon as the required number of participants is reached – i.e., the predefined number of FLMs expressed their interests in joining the training process – the FPCE initiates the training and the following operations are performed. The FPCE sends the configuration information of the FL process (possibly including the latest version of the global AI model) to all the participating FLMs, which in turn use their local data to train a local AI model that is then sent to the FPCE. The latter aggregates the received local models only when *all* of them have been received. In the configuration where FLMs are deployed on the MEC host, the FLMs receives data for training from their respective UEs via small periodical reports.

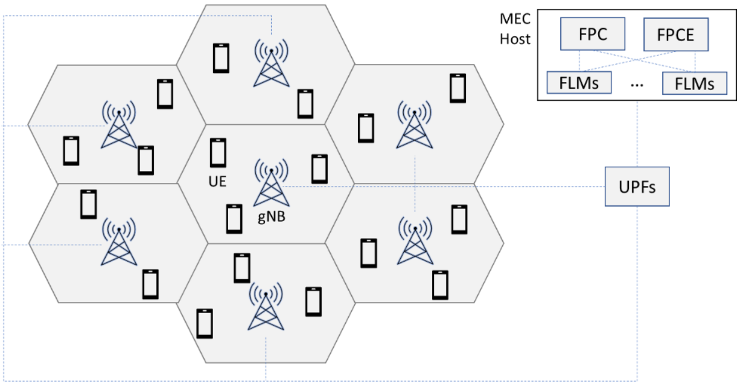


Figure 3-12 Simulation scenario for the evaluation of the FLaaS framework.

We evaluate the above use case simulating two different deployments of the FLM (on the UE, on the MEC), each with two different network loads (light, heavy). In particular, we consider 10 background UEs per cell with light load, and 30 background UEs per cell for heavy load. Each scenario has been run with an increasing number of foreground UEs participating in the training. As far as the training times of the local model are concerned, they may vary significantly as the FLMs can run on either the MEC or the UEs (which can also have heterogeneous capabilities – such as smartphones or laptops). Thus, we generated such training times according to different distributions, based on whether the FLMs are run. Shorter training times are assumed when FLMs run on the MEC since we can expect that a MEC host has more computational resources than an end device. The size of the global configuration, the training duration and other network parameters are reported in Table 3-4. The overall learning times that are the subject of this analysis are computed starting from when the FPCE selects the FLMs for the training to when it receives back the trained local model: hence they include uplink and downlink communications and training times, as well as the time needed to traverse all the protocol stack from/to the application layer (e.g., including the setup times for TCP connections).

Table 3-4 Main simulation parameters for the evaluation of the FLaaS framework.

Network parameter	Value
Bandwidth	10 MHz @ 2 GHz carrier frequency
Duplexing scheme	Frequency Division Duplexing
Path loss model	Urban Macro (UMa) [38.901]
Number of gNBs (foreground / background)	7 / 12
Number of UEs	[20, 40, 60, 80]
Number of background UEs (per background gNB)	[10, 30]
UE speed	~*U* (13.8 m/s, 41.7 m/s)
Background traffic	CBR @ 50 kB/s (DL); CBR @ 20 kB/s (UL)
FLaaS parameter	Value
Deployment of FLM	[@UE, @MEC host]
Size of global configuration	240 kB
Size of the local model	~*U* (70 kB, 80 kB)
Duration of local model training at the UE	~*Exp* (50 s) with 0.9 probability; ~*Exp* (85 s) with 0.1 probability
Duration of local model training at the MEC host	~*N* (15 s, 2 s)
Duration of model aggregation	500 ms * #received local models
Dataset chunk size and period	140 B, 1 s

Figure 3-13 shows the time needed by the FPCE to retrieve the number of local AI models on the x-axis, in the two load conditions described above. For both load conditions, the training times are extracted from the same distributions, hence the difference between the two charts is due to the radio communication latency, i.e., the time it takes to send the global configuration in the downlink and the local models in the uplink. The charts allow us to observe how many local models the FPCE should expect to receive at any given time: such information can be useful for tuning the parameters of the FL process, i.e., a maximum waiting time between the start of the training process and the start of the aggregation phase. For example, if the FPCE starts aggregating models after 100s and the number of FLMs involved in the training is 80, the FPCE will have 65-70 models to aggregate when the network is lightly loaded, whereas it will aggregate around 60 models when the network is heavily loaded.

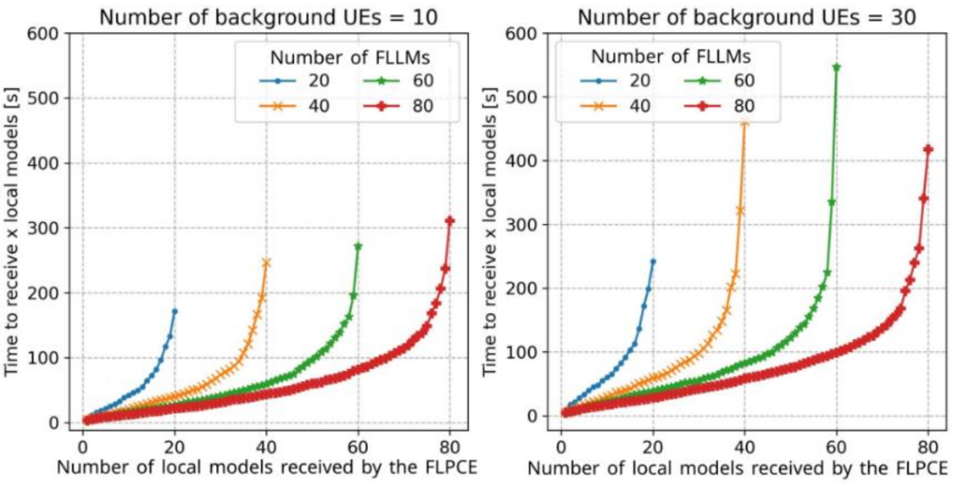


Figure 3-13 Time to receive local model, when FLMs are deployed at the UEs, with light (left) and heavy (right) load in the RAN.

The Empirical Cumulative Distribution Functions (ECDFs) of the time needed to exchange the models over the air in Figure 3-14 and Figure 3-15 confirm the above considerations and show how the load of the network affects such times. In particular, Figure 3-14 shows that when the network is lightly loaded the time to send the global model in the downlink also depends on the number of the FLMs, while with a heavily loaded network the times do not depend anymore on the number of the FLMs. This is because the data traffic generated by background UEs is prevalent with respect to the one needed to send global models to the FLMs. Figure 3-15 shows that the number of FLMs does not significantly affect the time needed to send their trained local models, but even in this case the load of the network is relevant. In particular, the probability that a model is sent in less than five seconds is 0.95 when only 10 background UEs per cell are present, while in the other case the probability is below 0.8.

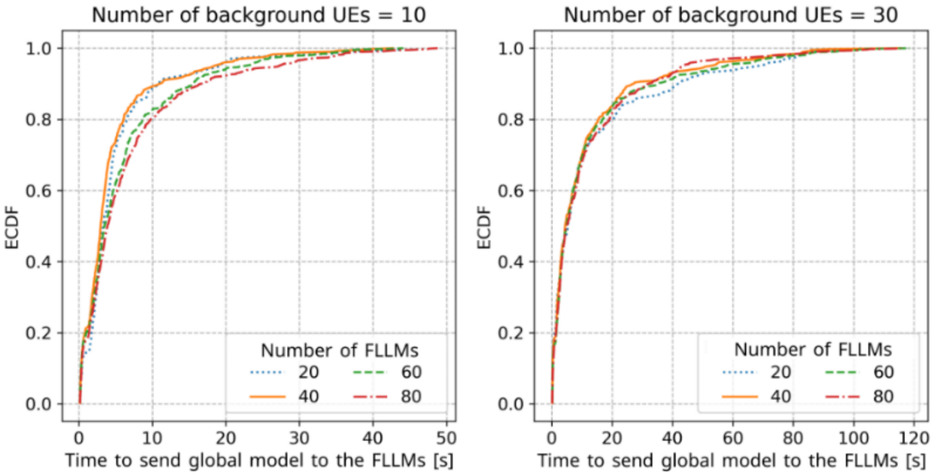


Figure 3-14 ECDF of the time required to send the global model from the FPCE to the FLM, when the latter is deployed at the UEs, with light (left) and heavy (right) load in the RAN.

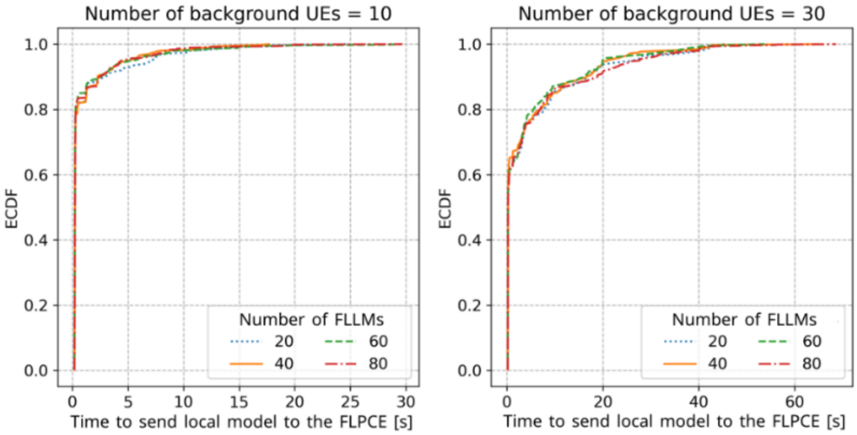


Figure 3-15 ECDF of the time required to send the local model from the FLM to the FPCE, when the FLM is deployed at the UEs, with light (left) and heavy (right) load in the RAN.

Figure 3-16 shows the time needed to receive the local models as a function of the number of FLMs, when these are deployed on the MEC host. Comparing the charts with the ones in Figure 3-13 we observe that the time needed to receive a given number of local models is reduced, hence the benefit of deploying the FLMs on the MEC is evident. Moreover, such time is independent of the mobile network load since models are not exchanged over the RAN anymore in this case. Although the data used for the training is transmitted as a data stream in the uplink, it consists of small amounts of data that occupy few resource blocks and do not suffer from high delays. When the FLM is located at the UE, instead, higher delays can be expected because model updates – which may be large in size – need to be sent over the RAN, hence occupying more resource blocks.

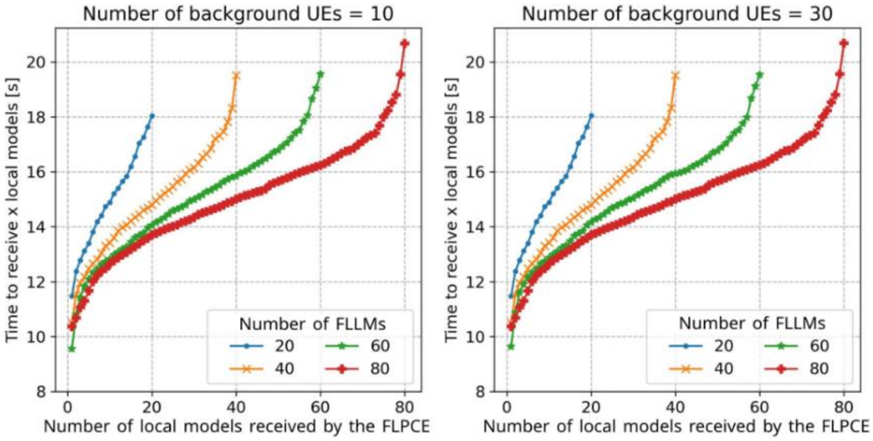


Figure 3-16 Time to receive local model, when FLMs are deployed in the MEC host, with light (left) and heavy (right) load in the RAN.

We now assess the energy savings that the FLaaS framework may provide when it is enabled in the network, specifically when the FLMs are deployed on the MEC. We compare such a scenario against a scenario without FLaaS, i.e., UEs can participate in some federation but the service is not provided by the network and the UEs need to run locally all the tasks that would be performed by the FLM. In such a scenario, the global configuration and local model updates must be transmitted over the RAN, hence the gNBs will consume more energy to perform radio transmissions that, instead, can be avoided when FLaaS is enabled and running on the MEC. Using Simu5G, we run simulations considering the same network topology as in Figure 3-12, with an increasing number of UEs participating in a federation. UEs train a local model after receiving the global configuration from the FL aggregator (which can be located, e.g., in a remote cloud) to the UEs. We focused on downlink transmissions and evaluated the number of time-frequency resources that gNBs needed to allocate to send the above global configurations to UEs. Then, we derived the corresponding energy consumption exploiting the evolutionary power model in [SRF+16], parametrized according to the values of base stations' power consumption for the year 2020. Such a model assumes that the gNB's power consumption is composed of a (fixed) baseline power and a load-dependent power that increases linearly with the cell load. Assuming that the consumed energy due to the baseline power is the same with and without FLaaS, Figure 3-17 shows the load-dependent part of the energy consumed by the gNBs to transmit the global configurations to the UEs when FLaaS is not enabled. Since the above transmissions are not required with FLaaS, values reported in Figure 3-17 can accordingly be considered as the energy saved by the gNBs when FLaaS is enabled and FLMs are deployed on the MEC.

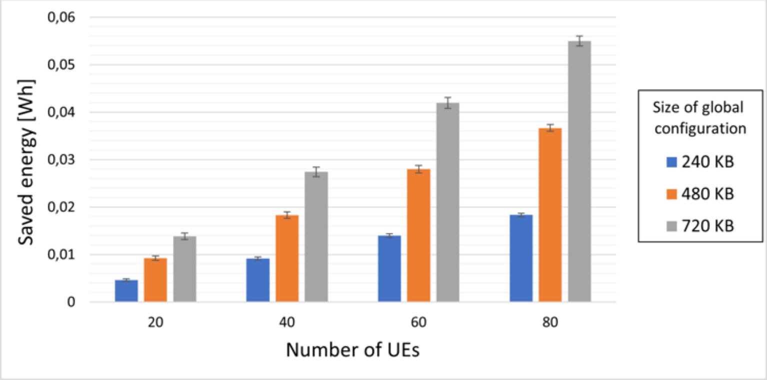


Figure 3-17 Energy saved by the gNBs when FLaaS is enabled and FLMs are deployed on the MEC.

Obviously, the higher the number of federated UEs, the higher the energy saved with FLaaS. Likewise, the energy saving increases with the size of the global configuration (which is a parameter that depends on the AI model to be trained). The benefits are more evident considering the percentage of savings: when the size of the AI model is 240KB, the resulting energy saving is 2.88%, and it increases to 8.14% when the size is 720KB. Note that the above chart refers to the energy saved when only one iteration of FL is considered. If multiple iterations are employed (resulting in multiple downlink transmissions), energy savings might be even higher. Considering that RAN power consumption is expected to be a major contributor to the overall energy footprint of 6G networks [HEX-D51][HEX-D52], integrating the FLaaS framework within the 6G network can contribute to reduce the energy consumption for supporting FL-based applications, hence can contribute to reduce the overall TCO.

3.2.3.1 FED-XAI PoC

In this section we describe how the FEDerated and eXplainable AI (FED-XAI) Proof of Concept (PoC) works and how the related testbed has been configured. The aim of this PoC is to demonstrate how XAI models trained in a federated way can be used to predict the quality of a video streaming in an automotive use case. In more detail, we consider a tele-operated driving use case (ToD), where cars connected to the 6G network send video streams to a remote-driver entity (human or machine) at the edge of the network. Intuitively, the remote driver can only drive the car safely when the video quality is good enough to ensure a smooth driving experience. Thus, it becomes important to predict when such quality will deteriorate, in order to allow the network and/or the remote driver to take appropriate countermeasures, e.g., by informing the physical driver (in the car) that in a few seconds he/she will need to take over the control of the car. Federated Learning (FL) is suitable for this scenario, as a large number of connected cars can contribute to train the forecasting XAI model. Furthermore, using an explainable model will help the mobile network operator or the remote driver to learn the root causes that generated a given forecast, hence allowing them to take the best counteraction. To do this, the FED-XAI PoC will provide a dashboard showing the prediction and its root causes in real-time.

In the context of the FED-XAI PoC, the training of the FED-XAI model and the inference operations are executed by a FED-XAI application described in [HEX-D43] and implemented according to the

FLaaS paradigm [HEX-D52] using the Intel OpenFL software framework [OPENFL]. The testbed that we realized to demonstrate the FED-XAI concept is composed of two main phases, namely an offline model training phase and an online (real-time) inference phase.

As far as the offline training phase is concerned, we exploited the Simu5G system-level simulator [NSS+20] to produce a meaningful dataset that includes a large set of QoS data produced by several video-streaming sessions. To do this, we implemented a model for a video-streaming application within Simu5G, where UEs send video streams to a remote host following a trace-based approach, i.e., sending rate and size of video frames sent by the UE are read from a trace file generated from dash-cam videos. This is useful to model video-streaming traffic that resembles the one in realistic ToD scenarios. The simulated network topology is configured as shown in Figure 3-18-, where UEs (i.e., connected cars) move along one main road and three intersecting roads. Intersections are regulated by traffic lights. Such portion of urban scenario is served by multiple BSs that provide connectivity to the UEs. The latter locally run the sender side of the video-streaming application, which streams the video to a remote-driving application hosted on a MEC host. Data extracted from the TIM's live network is elaborated by considering traffic forecasts, network counters and MDT samples. It is then exploited to make the scenario more realistic and, as a consequence, to produce datasets more meaningful of a future 6G traffic configuration. Also, the position of BSs in the simulation is set according to their actual position in the city of Turin. Moreover, the actual data volume handled by those BSs was used to configure the background traffic in the simulation, i.e., to produce realistic cell workloads. In more detail, we used data-volume values provided by cell-wise network counters from the TIM's network, which provide averaged metrics over a time span of 15 minutes. Three days of such values were extracted, resulting in 288 values for each BS. This guided the configuration of our simulation campaign to generate the dataset: we configured 288 simulation instances, each 15-minute long, during which the data volume served by the BSs (i.e., its workload) corresponds to that provided by TIM's network counters. Since an AI model is more effective when it is trained with a large amount of data, each 15-minute simulation instance was repeated five times with different seeds of the pseudo-random number generators. This also has the effect to simulate multiple UEs' mobility patterns and, in turn, it increases the variability of the scenarios learned by the training algorithms.

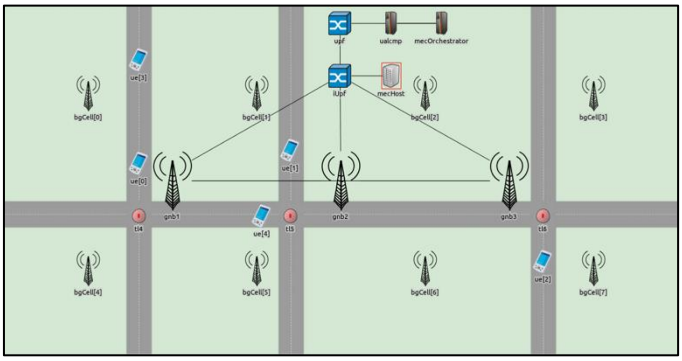


Figure 3-18 Representation of the simulation scenario used to produce the training dataset.

Figure 3-19 shows an example of QoS metric that we extracted from the above simulation campaign, i.e., the evolution over time of the end-to-end delay of video segments, where we observe that the metric changes over time due to UE mobility and variable interference produced by the background traffic.

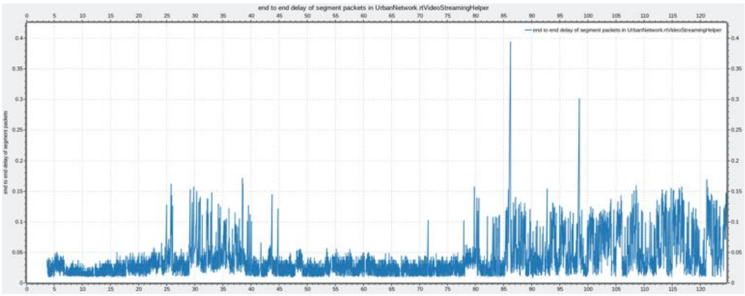


Figure 3-19 End-to-end delay of video segments over time.

Once the FED-XAI model has been trained, it can be used to perform real-time prediction about the quality of a video-streaming data flow. Thus, the online inference phase of the FED-XAI PoC is performed using the testbed shown in Figure 3-20. The testbed is composed of two laptops equipped with VLC and running the sending and receiving side of the video stream, respectively. The video traffic flows through a third PC, in the middle, running Simu5G.

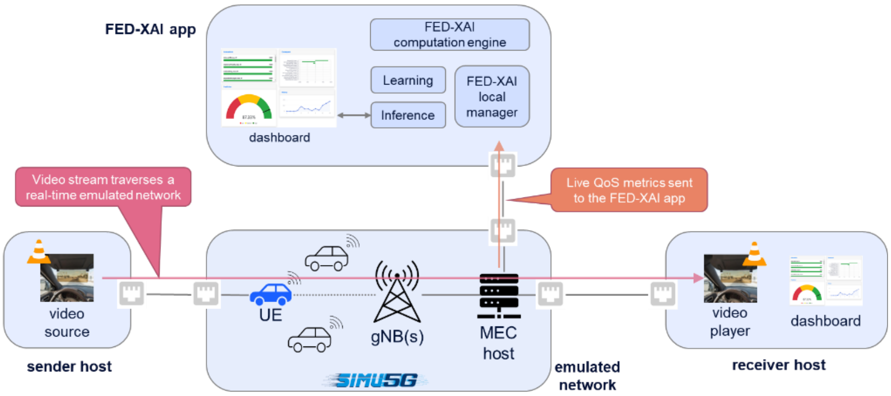


Figure 3-20 Representation of the real-time FED-XAI testbed.

In this case, Simu5G is run in real-time emulation mode: this means that the simulated time is synchronized with the real (wall-clock) time and that real video packets sent by the video source can be injected into the simulation, traverse the emulated network scenario, and be delivered to the video player. Since the emulated network is configured with the same scenario depicted in Figure 3-18, video packets will experience different network conditions, resulting in, e.g., different delays or loss probability. In turn, the quality of the video played out by the receiving laptop will be a consequence of the (emulated) network conditions. In order to enable the video quality prediction, QoS metrics of the video flow are extracted in real-time from the emulated network and sent to a fourth PC running the FED-XAI application. In particular, real-time QoS metrics will be received by the inference module of the FED-XAI application (see Figure 3-20), which will exploit the XAI model pre-trained in the offline training phase to produce a prediction of the quality of the video. Such prediction is shown in real time on a graphical dashboard. It is expected that when the inference module predicts bad video quality, then the actual video played out by the receiving laptop will be impaired or will freeze. The details about the design of the FED-XAI applications (including the dashboard) and the related algorithms, as well as the experimental results of the inference procedure are described in [HEX-D43].

3.2.4 Forecast-based recovery in Real-time remote Control of robotics (FoReCo)

As presented in [HEX-D52], one of the key aspects of Intelligent network in 6G will be to provide AI guided network enhancements to applications. We designed a Forecast-based recovery mechanism for Real-time remote Control of robotic manipulators “FoReCo”, as one example of such application where

AI is used to improve the reliability of a certain application. It is suitable for autonomous, or human assisted remote control of robot manipulators that perform repetitive tasks such as welding, materials handling, picking, and packing, or assembly. In case the robot does not receive a remote-control command on time due to IEEE 802.11 collisions or EM interference, FoReCo (i) infers the delayed command; and (ii) injects it in the robot driver loop so the operator does not perceive misbehaviour in the remote control process. In following sections, we describe in detail the FoReCo building block and how it infers delayed/lost commands through Machine Learning (ML). Then, we explain the analytical model of IEEE 802.11 [BLB+20] that we use to test FoReCo in simulated scenarios with wireless interference.

3.2.4.1 FoReCo Building Block

In this section, we present FoReCo as a forecast-based recovery mechanism to minimize the trajectory error of remotely controlled robots via wireless connectivity. Depending on the robot, the absence of the command c_i may result in the robot stops or keeps feeding the prior command c_{i-1} to the robot control loop, which is implemented with solutions as Proportional Integral-Derivative (PID) controllers (see [CSC+12]). Either way, the command c_i will not be executed and the robot trajectory will deviate from the expected, i.e., the trajectory executed by the remote controller. It is at this point that FoReCo predicts the command c_i that has not arrived on time and transparently triggers its execution into the robot. Hence, FoReCo stands as a complementary solution for any remotely controlled robot using a wireless link, while being agnostic to the implemented robot controller (control theory-based or not).

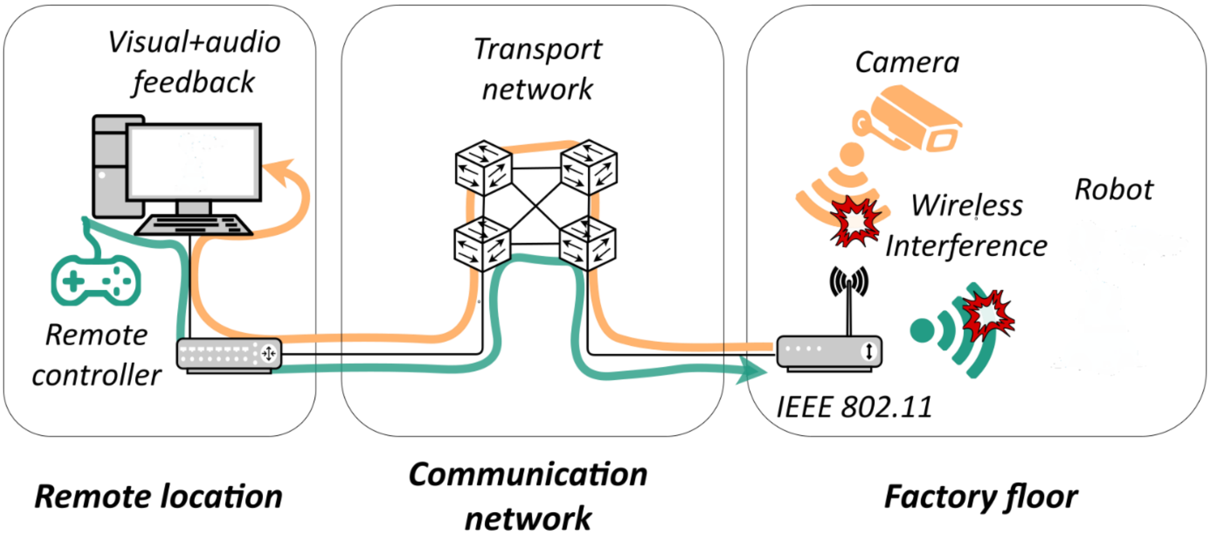


Figure 3-21 Diagram of an industrial robotic remote control.

To predict control commands out of time, FoReCo follows an ML based approach, which has been proven to be effective with intention prediction and estimation of future trajectories of objects, such as vehicles, bikes, and humans. The learning model consist of predicting incoming control commands c_i, c_{i+1}, c_{i+2},... with the help of the *prior* c_{i-1}, c_{i-2},... commands. To do so, we advocate for an ML based methodology due to (i) the repetitive nature of the industrial tasks performed by remotely operated robots; and (ii) the difficulty to solve this problem with traditional dynamic programming algorithms. Figure 3-21 shows the conceptual components of the network control system we use to remotely control a robot (in-line with Figure 3-21). The system shows the details of the interactions between the remote site (where the controller is located) and the factory floor over a communication channel. First, a real-time video stream of the robot is presented

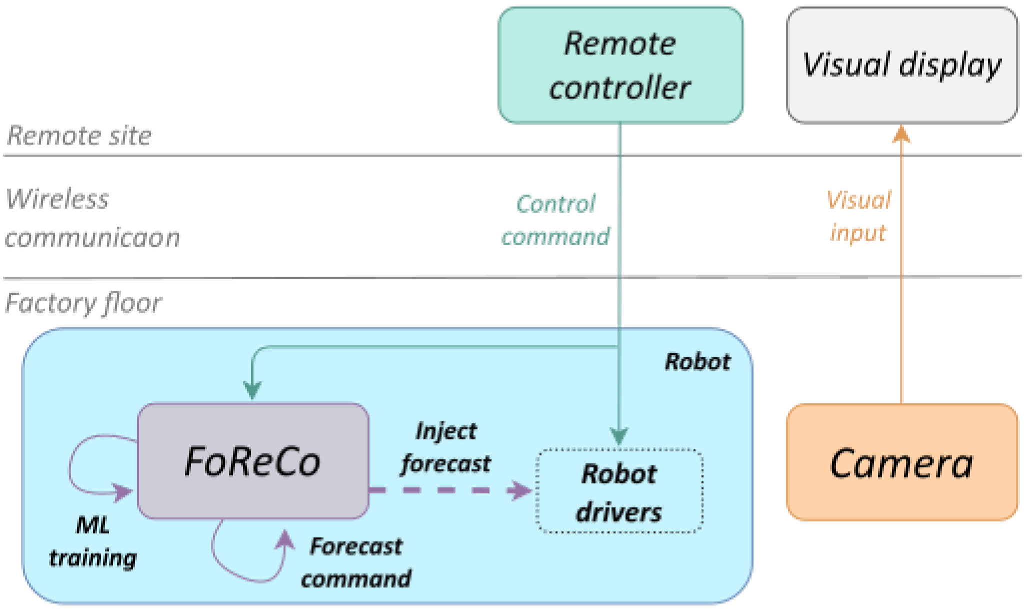


Figure 3-22 FoReCo building block and remote-control system.

to a visual display over a wired communication channel. For simplicity, we assume that the uplink channel is error and delay-free and the video input is delivered to the remote operator immediately. Then, the remote controller, with the help of the visual input, sends control commands over the wireless communication channel, and the commands are received by both the robot and FoReCo. With the received commands, FoReCo performs two actions:

1. **ML training**: FoReCo resorts to ML to derive f ({cj}, $\vec{\boldsymbol{w}}$), with $\vec{\boldsymbol{w}}$ being the weights to learn. To obtain $\vec{\boldsymbol{w}}$, FoReCo creates a dataset (see Figure 3-25) with the commands it receives from the remote controller. The dataset contains a history of H commands, and FoReCo uses αH of them for training, and βH for testing; with α + β = 1. The training procedure aims to minimize the distance between predicted commands $\hat{c}_1$, and the ones sent by the remote operator c_i. Hence, FoReCo trains its ML solution f ({c_j}, $\vec{w}$) s.t.:

$$\min_{\vec{w}} \frac{1}{\alpha H} \sum_{\mathrm{i}}^{\alpha H} d\left(c_i, \mathrm{f}\left(\{c_j\}_{i-R}^{i-1}, \vec{w}\right)\right)$$

   With the obtained weights $\vec{w}$, FoReCo tests the ML predictions accuracy in the testing set βH.

2. **Command forecast, validation and injection:** FoReCo awaits a control command c_i each Ω ms, and it triggers the forecasting if the next command c_{i+1} arrives latter than a(c_i) + Ω + τ. In this case, FoReCo will forecast the next command as $\hat{c}_\iota$+1 = $f\left(\{\hat{c}_j\}_{i-R}^{i}, \vec{w}\right)$ using the ML solution f and the weights $\vec{w}$ obtained from the training stage. Next, FoReCo will validate the forecast by checking if the forecasted command offset is within the acceptable boundaries with respect to the current position of the robot. This validation is performed by FoReCo to prevent forecasts that can lead to an accident, malfunction, or robot misuse. The valid forecast command $\hat{c}_{\iota+1}$ is then injected in the robot drivers (as illustrated in Figure 3-22). In the case a command arrives on time a(c_i+1) ≤ a(c_i)+Ω+τ, FoReCo will just store the command in the dataset and later use it for training and forecasting purposes. Thus, FoReCo receives as input $\left(\{\hat{c}_j\}_{j-R}^{i}\right)$ commands that arrived on time, and the forecasts of previous commands that did not arrive on time.

3.2.4.2 IEEE 802.11 with electromagnetic interference

So far, we have discussed how FoReCo works in previous section. In this section we explain the analytical model we consider for deriving the delay that control commands experiment $\Delta_W(c_i)$ in IEEE 802.11 wireless links under electromagnetic (EM) interference. The analytical model is latter used in

Section 3.2.4.4 to derive the $\Delta_W(c_i)$ and assess the performance of FoReCo in a simulated scenario as close as possible to real IEEE 802.11-based real-time remote control.

Here, we resort to the analytical model presented in [BLB+20] to derive wireless delays. This work models the Medium Access Control (MAC) layer of IEEE 802.11 with Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA), and studies how neighbouring nodes and non-IEEE 802.11 interfering sources impact the wireless delay. The work is based on a refinement [Pha05] of Bianchi's characterization of IEEE 802.11 [Bia00]. The particularity is that [VTS13] extends the underlying Markov chain to also capture the presence of an interference source that is active during T_{if} transmission slots, and emits with a probability p_{if}. The proposed model also captures both the back-off mechanisms and re-transmissions (RTX) of frames upon collision in the IEEE 802.11 wireless link.

With the model, [BLB+20] obtains the steady-state vector of each state, in particular, they derive the probability that a frame has to be transmitted after j unsuccessful retransmissions (RTX), which is denoted as a_j. Moreover, [BLB+20] also derives $E_j[\Delta_W(c_i)]$, that is, average delay that the command c_i experiences in the wireless transmission after j unsuccessful re-transmissions. Based on such expressions, we derive some theoretical results around the analytical model given in [BLB+20], that give some insights about the delay of control commands. In particular, the theoretical results conclude that in the considered IEEE 802.11 scenario:

- $\Delta(c_i)$ is only bounded on average, but not always.
- $\Delta(c_i)$ diverges and
- the causality assumption does not apply.

In other words, we cannot bound the delays that the remote-control commands c_i are experiencing. Still, we resort to the analytical model presented in [BLB+20], as such unbounded delay behaviours are realistic in IEEE 802.11 scenarios upon the presence of interference sources.

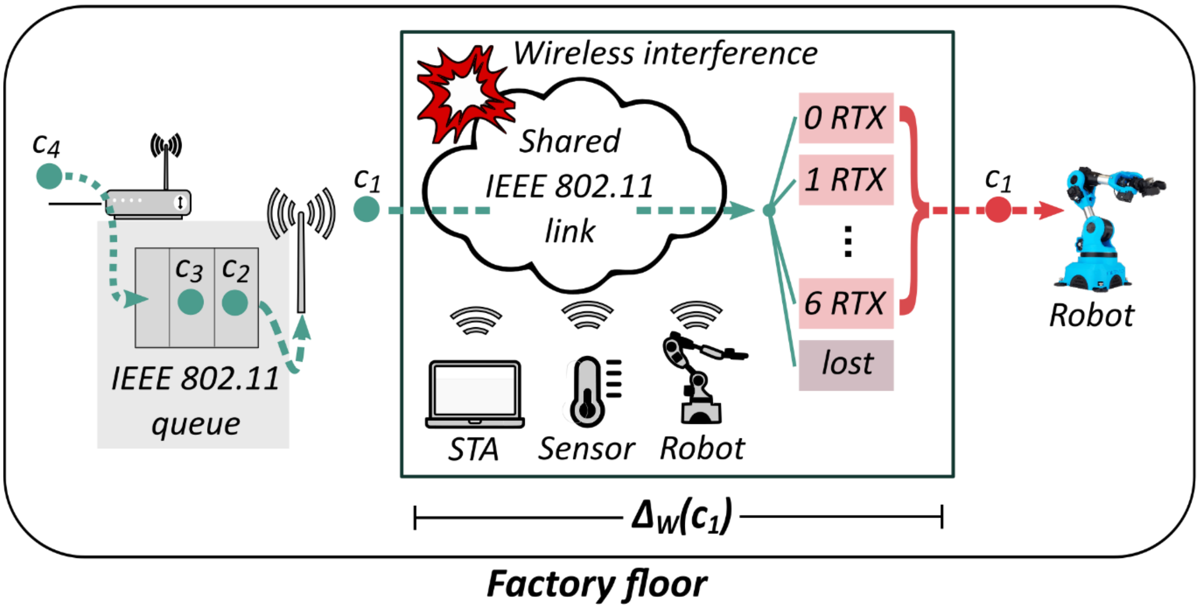


Figure 3-23 Impact of wireless interference, retransmissions (RTX), and factory devices in the delay Δ_W (c_i) that control commands experience in an IEEE 802.11 link.

To derive the value of $\Delta_W(c_i)$ we follow [BLB+20] and model the transmission of control commands c_i over IEEE 802.11 wireless links as a queuing model. As we know that control commands have an arrival rate $\frac{1}{\Omega}$. These commands are queued in the IEEE 802.11 access point before they are transmitted to the shared wireless link. Following the IEEE 802.11 standard, a frame is re-transmitted up to 7 times.

After this threshold is exceeded, the frame (and therefore, the control command) is assumed to be lost and no further re-transmission is executed (see Figure 3-23).

Depending on the number of RTX, the control command delay $\Delta_W(c_i)$ will be higher or lower. This system behaves as a $G/HEXP/1/Q$ queuing model, with Q being the length of the access point queue, and the service rates of the hyper exponential distribution corresponding to the average delay that control commands see after j RTX, i.e., $\frac{1}{E_j[\Delta_W(c_i)]}$.

Given this $G/HEXP/1/Q$ queuing model, we can derive $\Delta_W(c_i)$ in the desired IEEE 802.11 wireless scenario accounting for the number of transmitting devices and the probability and time that the wireless interference is active. These are the delay values used in the simulation scenarios in Section 3.2.4.4, and we derive them using the CIW discrete event simulation library [PKH+19]. Note that in the future, the system will highly benefit of the improved latency of the upcoming WiFi 7.

3.2.4.3 Data collection

Figure 3-24 shows part of the dataset created by performing pick and place actions, while Figure 3-25 presents a 3D representation of the complete dataset. The pick and place actions were manually repeated 100 times by two different human operators, an experienced and inexperienced human operator resulting in the creation of two separate datasets. To do so, they used the joystick as a remote controller. The continuous joystick movement is transformed into control commands generated every 20 ms. The inexperienced/experienced operators' datasets contain $H = 187109$ commands. Both datasets store the joint states c_i of the robot manipulator under ideal network conditions, i.e., low latencies and absence of packet collision. To achieve such conditions, the datasets were obtained using Ethernet to send the remote controller commands. The experienced dataset was used to train the ML models while the inexperienced data was used for remote control and testing. In this way, we ensure that the trained ML model operates on data that is tightly related but different from the training data.

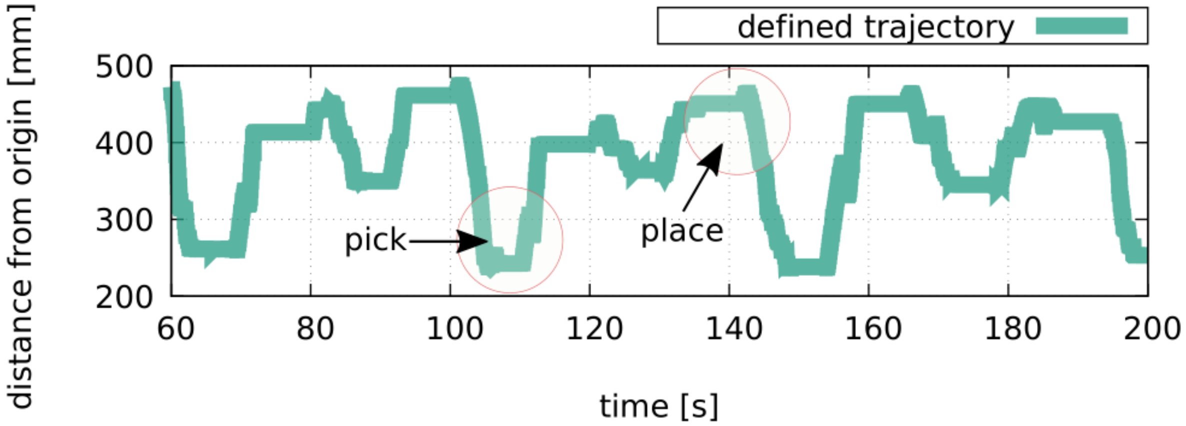


Figure 3-24 Robot trajectory dataset with pick and actions of an inexperienced operator.

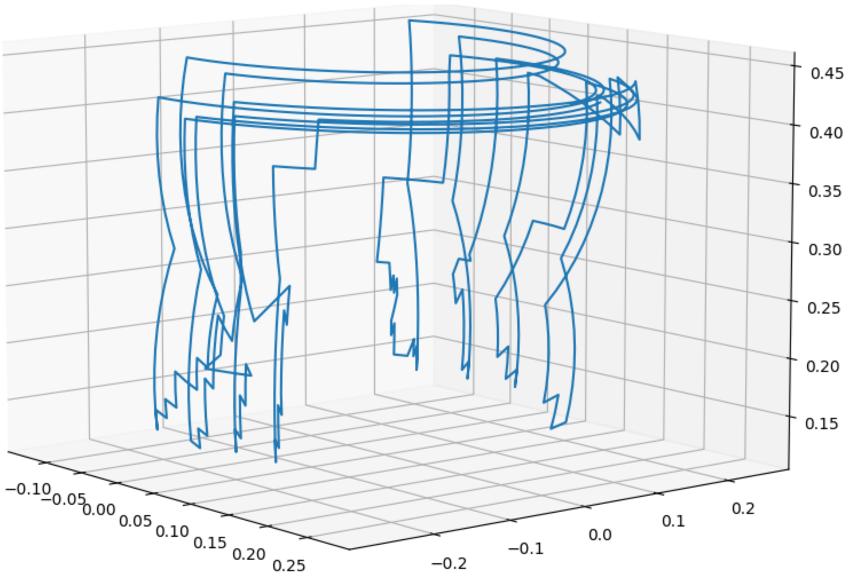


Figure 3-25 3D representation of the robot arm movement dataset

In [HEX-D52], we evaluated different ML algorithms such as VAR (Vector Autoregression), Massive Average (MA), and sequence to sequence (seq2seq) models to check which achieves the highest forecasting accuracy based on a collected dataset. Results indicated that VAR was slightly more accurate than MA, while seq2seq performed the worst. This was due to the vast number of weights to learn $|\vec{w}| = 163803$ that prevented it from reaching an optimal solution. It was expected that VAR, designed for correlated time-series such as the 6-axis time-series of the Niryo One robotic arm, would outperform MA. In Section 3.2.4.4, we use the MA and trained VAR solutions as forecasting techniques for simulation analysis.

3.2.4.4 Simulation evaluation

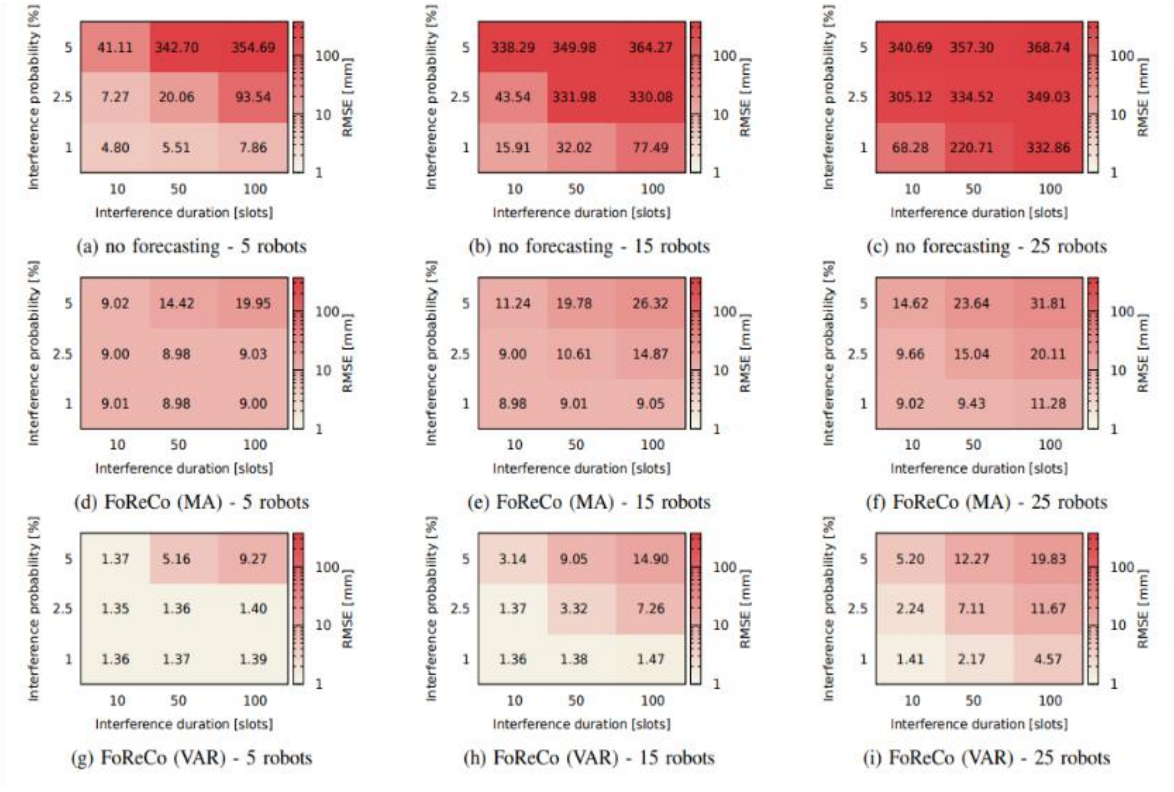
In the following, we evaluate how FoReCo behaves under a simulated environment with wireless interference. We consider a transport network with negligible transport delay, i.e., D ≅ 0 ms thus, commands' delays are dominated by the wireless delay $\Delta(c_i) \cong \Delta_W(c_i)$. To derive $\Delta_W(c_i)$, we resort to an analytical model of IEEE 802.11 with non-IEEE interfering sources [BLB+20], and use the parameters reported in [BLB+20, Table 2]. The goal of the simulation validation is two-folded: (i) evaluate the precision of the forecasted commands by FoReCo, and (ii) assess the scalability with up to 25 robotic arms sharing a wireless medium with interferences. All the details about the simulation implementation of FoReCo and the IEEE 802.11 analytical model can be found in our publicly available git repository[2].

Each simulation issues the commands of an inexperienced human operator and introduces command delays $\Delta_W(c_i)$ following [BLB+20]. shows the error experienced by the robot trajectory. Figure 3-26 (top) shows the results using the state-of-the-art solution, i.e., repeating the prior command $\hat{c}_{i+1} = c_i$ upon delays. Figure 3-26 (middle) shows the results when FoReCo recovers packets using the MA solution, and (bottom) shows the results when FoReCo uses the VAR solution to recover packets. Since the introduced wireless delay $\Delta_W(c_i)$ is a random variable, we repeat each simulation 40 times. Note that, in each simulation, we vary the time and probability of the active interference. Each square in the heatmap illustrates the averaged RMSE of the 40 simulations done for every pair of interference duration, and probability. The RMSE is computed over the entire robot trajectory induced by the inexperienced human operator, and it considers commands arriving on time $\Delta(c_i) \leq \tau$ and out of time

[2] https://gitlab.it.uc3m.es/5g-team/FoReCo

$\Delta(c_i) > \tau$, without using control command forecasting (top in Figure 3-26), and with FoReCo using MA and VAR (middle, and bottom rows in ; respectively).

The RMSE error in Figure 3-26 is represented in logarithmic scale, and we can appreciate that FoReCo command recovery constrained the robot trajectory error below 19.95 mm, 26.32 mm and 31.81 mm using MA (middle row) and 9.27 mm, 14.90 mm and 19.83 mm using VAR (bottom row) for 5, 15 and 25 robots on the factory floor, respectively. Figure 3-26shows that the VAR solution outperforms the MA solution in every simulation scenario for approximately 10 mm. On the other hand, the no forecasting solution resulted in an RMSE in the order of ~ 350 mm in the worst cases, no matter the number of robots. Thus, simulations indicate that (i) the VAR solution outperforms the MA solution by minimizing the error for additional 10 mm; (ii) FoReCo based on VAR will not exceed errors of 20 mm; and (iii) FoReCo reduces the experienced error by more than one order of magnitude. In particular, FoReCo using VAR reduces by more than a 94.4% (368.74 mm with no forecasting and 19.83 mm with FoReCo (VAR) the experienced error in factory floors of 25 robots).



(a) no forecasting - 5 robots
(b) no forecasting - 15 robots
(c) no forecasting - 25 robots
(d) FoReCo (MA) - 5 robots
(e) FoReCo (MA) - 15 robots
(f) FoReCo (MA) - 25 robots
(g) FoReCo (VAR) - 5 robots
(h) FoReCo (VAR) - 15 robots
(i) FoReCo (VAR) - 25 robots

Figure 3-26 Robot trajectory error upon interference without forecasting (top), with FoReCo using MA (middle), and FoReCo using VAR (bottom).

3.2.5 Network programmability for traffic steering and adaptive packet processing

Programmable data planes are expected to enable the rapid development of new network functionality. In Hexa-X delivery D5.2 (see [HEX-D52]) we have discussed how networks will become programmable by introducing Network Interface Card (NIC), routers, and switches supporting network programming (e.g., based on the P4 programming language).

In the following sections we propose solutions to improve existing 5G System (5GS) performance measurements and packet processing functionality by leveraging the P4 programming language. We use the 3GPP Release 17 work item Access Traffic Steering, Switching, and Splitting (ATSSS) as a baseline for the proposed enhancements. However, the proposed enhancements are generic and can be reused for other use cases including redundant transmission for high reliability communication as specified in 3GPP TS 23.501 clause 5.33.2 (see [23.501]).

ATSSS as illustrated in Figure 3-27 introduces a Multi-Access PDU (MA PDU) Connectivity Service, which can exchange PDUs between the UE and a data network by simultaneously using one 3GPP access network and one non-3GPP access network with two independent N3/N9 tunnels between the UPF PDU Session Anchor (UPF PSA) and the access networks.

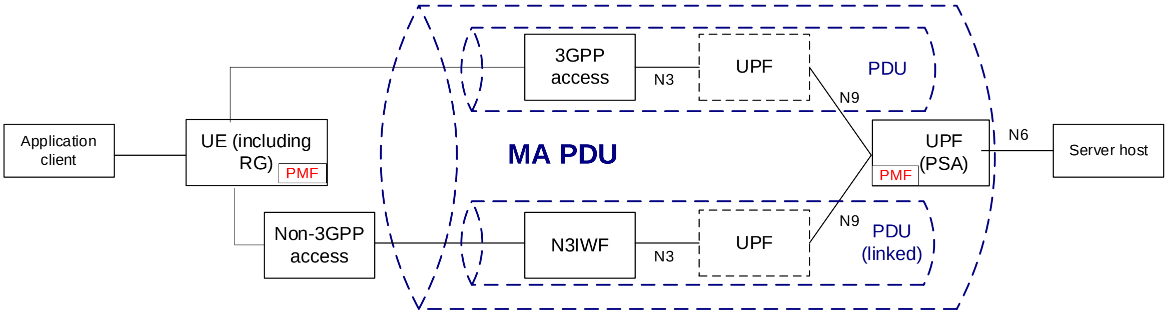


Figure 3-27 ATSSS support in the 5G system architecture according to 3GPP Rel-17.

Based on network-provided policy (ATSSS rules), local conditions (such as network interface availability, signal loss conditions, user preferences, etc.), and end-to-end performance measurements (Round-Trip-Time (RTT) and Packet Loss Rate (PLR)) the UE decides how to distribute the uplink traffic across the two access networks. Similarly, the UPF PSA based on network-provided policy (i.e., N4 rules derived by UE's serving SMF based on ATSSS policy from serving PCF), feedback information received from the UE via the user-plane (such as access network Unavailability or Availability), and end-to-end performance measurements, the UPF decides how to distribute the downlink traffic across the two N3/N9 tunnels and the two access networks.

ATSSS specifies the usage of Multi-Path TCP Protocol (MPTCP) or ATSSS Low-Layer (ATSSS-LL) as supported steering functionality. ATSSS-LL specifies a Performance Measurement Function (PMF) in the UE and the UPF PSA as shown in Figure 3-26 supporting RTT and PLR measurements per access between UE and UPF PSA. PMF measurements are per QoS Flow and can be used to decide how to steer the traffic of a specific data flow.

However, the available solution in 3GPP Rel-17 for Round-Trip-Time (RTT) and Packet Loss Rate (PLR) measurements require additional control packets (in addition to user data packets) to be exchanged over the access network. This has the following issues:

- Creates an overhead in the number of data to be exchanged over the access network.
- Provides potential inaccurate RTT measurement as the additional control packets may observe different processing delays compared to the actual user data packets.

Furthermore, the intermediate network devices (switches and routers not shown in Figure 3-26) processing the N3/N9 traffic typically perform packet processing prioritization purely based on the DSCP value in the GTP-U outer IP header where the access network or UPF derives the DSCP value from the configured QoS Flow information and sets this value for all packets associated with the QoS Flow. This has the following issues:

- Packet processing priority handling based on only static values may result in a rigid forwarding performance not able to dynamically adjust to the varying network conditions.

In the following we propose an advanced ATSSS solution to mitigate the issues outlined above by leveraging end-to-end programmability for the collection and reporting of network states (using In-band Network Telemetry (INT) as specified in [INTP4]) and for performing changes to traffic engineering and packet forwarding.

Figure 3-28 illustrates an advanced ATSSS architecture where the participating network devices support end-to-end network programmability based on P4.

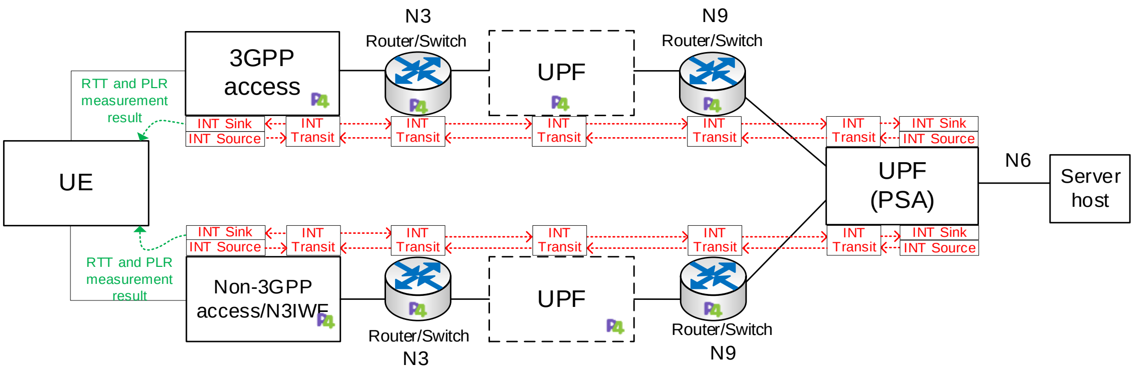


Figure 3-28 Advanced ATSSS architecture supporting end-to-end network programmability.

In the following sections we propose solutions for supporting (i) enhanced performance measurements using INT and (ii) adaptive packet processing priority handling leveraging P4 based network programmability.

3.2.5.1 Enhanced performance measurements using INT

In 3GPP Release 17 the ATSSS-LL feature specifies the exchange of PMF-Echo Request and Response messages for RTT measurements. The UE and the UPF initiate the measurement independently by sending a PMF-Echo Request over a specific access leg. The RTT delay is estimated based on the time until the initiator receives a PMF-Echo Response message over the same access leg. This solution creates an overhead as it requires the exchange of dedicated control packets in addition to the user data packets. There is also a potential issue that the estimated RTT measurements are inaccurate as those control packets may observe different processing delays compared to the actual user data packets.

Furthermore, for measuring the packet loss rate (PLR) the 3GPP Release 17 ATSSS-LL feature specifies the exchange of PMFP PLR request/response messages. The UE and the UPF initiate the measurement independently by sending a PMFP PLR count request message. The receiving UPF or UE starts counting the received packets until a PMFP PLR report request message is received and then sends a report in a response message. The initiating UE or UPF calculates the PLR based on the amount of sent and successfully received packets. Again, this solution creates an overhead as it requires the exchange of dedicated control packets in addition to the user data packets.

To mitigate the issues outlined above, we are proposing INT-based performance measurements in 6G. INT supports tracking packets through a network by inserting an INT header with instructions to collect network state metadata into the packet as it traverses the network. To measure and report latency, an INT source embeds instructions to collect the time stamp of the local egress and local ingress to report the difference as the latency for that network element. The receiving INT transit device can compute the end-to-end latency as a sum of the per-hop latencies. Per-hop latencies in the packet received at the INT transit can also be used to determine which network element(s) contributed most to the end-to-end latency.

Specific to the ATSSS-LL RTT performance measurement as specified in Figure 3-28 we propose to mimic the PMF-Echo Request/Response procedure using INT. For uplink and downlink packets, the access network collects metadata of the UE-AN interface (e.g., the packet delay and PLR observed over the radio link) in the role of an INT transit device. In addition to the standard INT behaviour to enable PLR measurements, it is proposed to enhance INT with additional instructions to request the counting of received packets associated with a specific QoS flow identified by the QFI available in the INT header. Furthermore, the performance information available at the INT sink needs to be mirrored back to the INT source to allow the initiator to consider the performance measurement result for the packet steering decision. For example, in [SPK+20] it is proposed to extend the handling of the standard INT Report packet to support the computation of the link latencies. In [SPK+20] the Report traffic packet is recirculated in the backward direction up to the INT source (the initiator of the measurement). In addition, to support the proposed architecture in Figure 3-28 where the INT source/sink is in the access network, we propose to expose the performance measurement result to the UE as needed by the steering logic in the UE. This can be done via a new access network exposure service (e.g., by extending the RRC or Non-Access Stratum (NAS) protocol).

3.2.5.2 Adaptive Packet processing priority handling

In today's 5GS deployments, transport level packet marking on a per QoS Flow basis can be supported by the RAN and UPF on the N3 and N9 interfaces in case the underlying transport is using QoS differentiation. Typically, packet processing in the transport network uses the 6-bit DSCP value in the GTP-U outer IP header for packet prioritization. However, limiting the packet processing priority handling of a specific packet of a specific QoS flow to a static value will result in a rigid forwarding performance not able to dynamically adjust to the varying network conditions. For example, the delay of a specific packet may accumulate (e.g., due to re-transmissions in RAN and network congestion) to a point where the packet is dropped at the receiver.

In 6G it is proposed to enhance INT with additional QoS related instructions derived from the associated QoS flow requirements and included in some or all packets (packet selection can be based on specific QoS requirements and/or network condition). In Figure 3-28 for uplink data, an INT header with QoS related instructions is inserted by the access network (INT source) following the GTP-U outer TCP/UDP header and removed by the UPF-PSA (INT sink) before forwarding a packet over the N6 interface. Likewise, for downlink packets the UPF PSA (INT source) inserts an INT header with QoS related instructions following the GTP-U outer TCP/IP header and the access network removes the INT header with the QoS related instructions before forwarding a packet to the UE.

For example, the INT header can include a new instruction to enforce a packet latency deadline (maximum allowed end-to-end latency) to be used by the INT transit increasing priority when the deadline approaches (assuming that the underlying P4 implementation is capable/enhanced to support the new instruction without adding noticeable packet processing delay). The latency deadline is derived from the associated QoS flow requirement and is included in the INT header or metadata by the access network (for uplink data) and the UPF (for downlink data). In the uplink and downlink path, the network devices are then programmed (e.g., by an SDN controller) to enforce prioritized packet processing based on the packet latency deadline (considering the latency of any previous hop) and to update the INT metadata with the observed hop latency. For example, in case the deadline approaches the processing is done with increasing priority, or in case the deadline has been exceeded the packet could be dropped. The INT sink decides whether to report the collected information (e.g., to the SDN controller). An example implementation of a queue management system on a P4 programmable network switch that works on a per-packet basis is specified in [TAZ+13]. The solution in [TAZ+13] demonstrates a significant improvement of network performance, especially for low-latency traffic, by significantly reducing the number of outdated packets without causing a drop in throughput.

3.2.6 UE programmability for conditional handover

Here we provide an example use case on how the concept of UE programmability can unleash programmable configurability for UEs (see Section 3.1.4.1 for the fundamentals of the concept). In a radio access network (RAN) typically new features that are introduced, e.g., for mobility robustness, are implemented incrementally via the radio resource control protocol (RRC) [3GPP TS 38.331]. The RRC protocol is one of the most important protocols controlling almost all protocols in the stack. Adding features to the RRC protocol requires 3GPP standardization which is critical in many cases. An example of a feature being added to the RRC protocol to increase the mobility robustness is Conditional Handover (CHO) [3GPP TS 38.300]. To motivate note that one problem related to robustness at HO is that the HO command is normally sent when the radio conditions for the UE have already deteriorated significantly. That may lead to that the HO Command may not reach the UE in time if the message is segmented or there are retransmissions. In order to avoid the undesired dependence on the serving radio link upon the time (and radio conditions) where the UE should execute the HO, the possibility to provide

RRC signalling for the handover to the UE earlier should be provided. To achieve this, it should be possible to associate the HO Command with a condition, e.g., based on radio conditions possibly similar to the ones associated to a handover A3 event, where a given neighbour becomes X dB better than source. As soon as the condition is fulfilled, the UE executes the handover in accordance with the provided HO Command.

One problem associated with CHO is that generalizing the scheme for various UE configuration with a rich set of conditions is difficult and time consuming due to the standardization process.

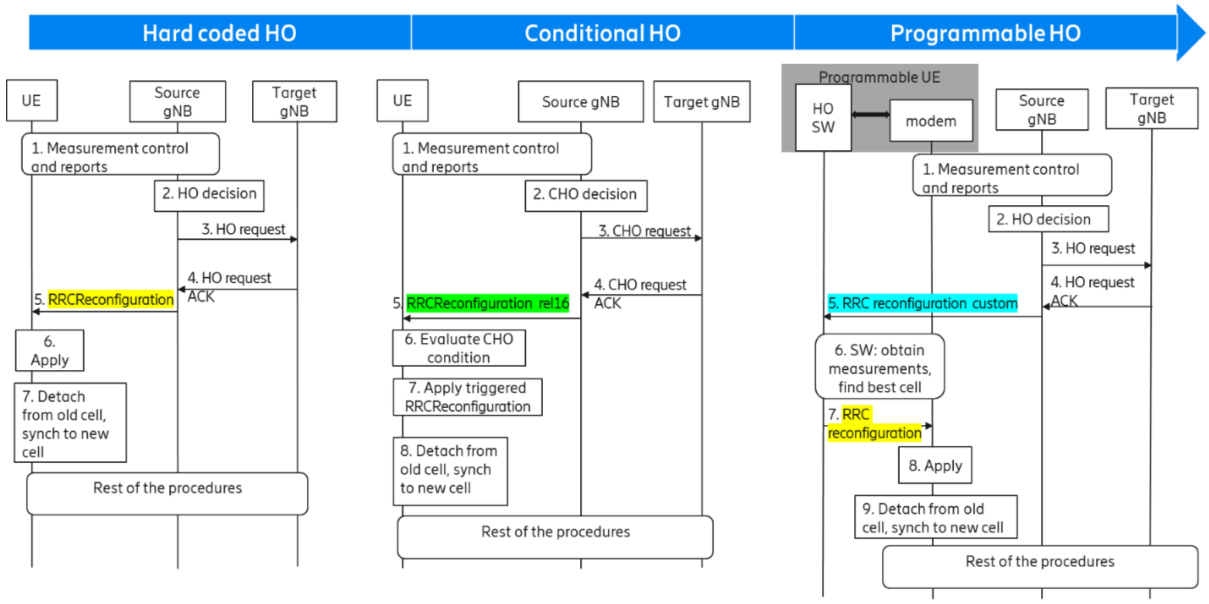


Figure 3-29 Evolution of HO towards programmable HO with UE programmability concept.

Figure 3-29 illustrates the evolution of legacy HO towards CHO, introduced in release 16, and programmable HO. The programmable HO utilizes the UE programmability architecture to enable programmable execution of HO commands. The network first installs a custom HO SW and delivers it to the programmability environment (PE) to be installed. Based on the measurements the source gNB initiates HO process by contacting the target gNBs and requesting a possible HO of the UE (step 1-4 common also to legacy and conditional HO). The source gNB prepares a custom message that contain the HO commands for the identified gNBs. Note that this differs from the CHO since the conditional HO message is standardized and needs to be in a specific format. Only UEs supporting the CHO feature will be able to understand it. This changes with programmable HO since the message is interpreted by the HO SW and hence the HO command containing the target cells configuration does not need to be standardized. The HO SW now has obtained the necessary information required to perform the HO to another cell. Next the optimal time for executing the HO needs to be decided which is done by the SW as opposed to CHO which is based on hard-coded conditions. The SW obtains necessary state information (as governed by the SW) such as the RSRP values, location of devices and determines the optimal time. When the time comes it delivers the RRCReconfiguration for the chosen cell and the UE receiving this message acts as though this has come from the network and executes the command.

The framework is not limited to the example above and various use cases can be realized with the framework. Custom messages can be generated to be delivered to the network by collecting information from the modem and processing them in the SW and triggering conditions can be programmable. For example, the measurement framework can be equipped with a programmed feature to introduce new measurement quantities to be calculated and triggering to report them can also be based on the conditions set by the SW and not the ones in the specification. Note here we cover how a legacy UE can be used in a programmable way to illustrate the power of programmability.

3.2.7 Integrated and distributed AI with supporting protocols

AI and ML are beginning to be key valuable tools in the context of mobile networks, as their complexity grows. Since 5G, mobile networks have become much more heterogeneous and complex and, therefore, the number of parameters to be configured over the whole network, to achieve optimal services, has increased almost exponentially. As a consequence, data-based approaches have been raised as the next-generation shift for legacy model-based approaches [WRS+20]. The main advantage of using AI techniques in these kinds of networks is the proven capability that they have to face humongous volumes of data and extract precise, meaningful actions/conclusions from them [GSR+21]. Therefore, in future 6G mobile networks AI is expected to be in charge or support a large set of operations over the whole network (i.e., predictive orchestration [HEX-D43], aid security functions, optimize placement, QoS/QoE monitoring and configuration, etc.). In summary, the role of AI and ML in future 6G mobile networks will be to aid those tasks where legacy techniques are not able to cope with the new conditions and requirements related to those networks e.g., high device heterogeneity, automation, multi-domain and multi-stakeholder environments, wide range of services, etc.

Machine learning algorithms can be divided into three fundamental categories: *Supervised Learning (SL), Unsupervised Learning (UL) and Reinforcement Learning (RL).* These types are mainly focused on *Centralized Learning*, but they can be applied in the field of distributed learning as part of the distributed AI model. In this domain, distributed AI models, a fundamental classification can be performed: *Federated Learning (FL), Decentralized Learning and Split Learning (SpL).* It is important to mention that other ML algorithm types exist, but they can be considered as a derivation of any of the aforementioned categories e.g., semi-supervised learning, self-supervised learning, stochastic learning, etc.

3.2.7.1 Centralised AI

In [HEX-D62] a detailed mapping between the most relevant features of these ML algorithm types and their potential impact and integration in Management and Orchestration (M&O) operations is given. This section briefly describes the main categories for Centralised AI/ML algorithms from a high-level perspective:

- **Supervised Learning**: These algorithms [HTF09] are able to map a given input to an output based on samples of input-output data-pairs. It requires a training phase where a set of labelled (classified pairs of inputs-outputs) training data is presented to the algorithm. SL is mostly used for classification and regression tasks, i.e., image/text classification, pattern recognition, time-series forecasting, etc.
- **Unsupervised Learning**: Unlike SL, UL algorithms [CA16] are able to analyse unlabelled datasets without any human intervention (data-driven processes). As with SL, it also requires a training phase to be able to correctly infer data. These ML algorithms are commonly used for anomaly detection tasks, feature learning tasks, clustering tasks, etc.
- **Reinforcement Learning**: Using an environment-driven strategy, RL [Glo00] is an ML technique that enables software agents and computers to automatically assess the best behaviour in each context or environment. It performs a learning process in which a subject's activities can be changed by following them with the proper positive or negative stimuli, aiming to reward good behaviour and inhibit inappropriate behaviour following a close-loop approach [SB98]. Compared to the previous categories, SL and UL, RL does not require a specific training stage as it learns following continuous trial and error iterations. It is a strong tool for training AI models that can help increase automation or optimize the operational efficiency of complex systems like 6G mobile networks.

Thus, to be able to create effective models in the various domains of 6G mobile networks (e.g., radio, core, M&O, services, etc.) it is of paramount importance to select the proper type of AI/ML technique depending on its learning capabilities, nature of the source data and expected outcome.

3.2.7.2 Distributed AI

Being able to access large datasets opens the road for superior AI models because data is the heart of any existing ML technique. Nonetheless, in 6G mobile networks the required amounts of data may not necessarily belong to the same specific party (i.e., same operator, vendor, vertical…). A simplistic approach to creating high-quality models would be to gather data from these sources and then train the resulting model using the gathered data. However, distributing large amounts of data is ineffective from the perspective of communication, and sharing data with other parties is typically not favoured owing to commercial and privacy considerations. From a legal standpoint, it may even be forbidden in specific circumstances. *Distributed* or *Collaborative* AI/ML techniques provide a way to leverage the parties' data without the need of actually sharing it. In fact, distributed AI learning models appeared as a reaction to privacy concerns regarding ML training.

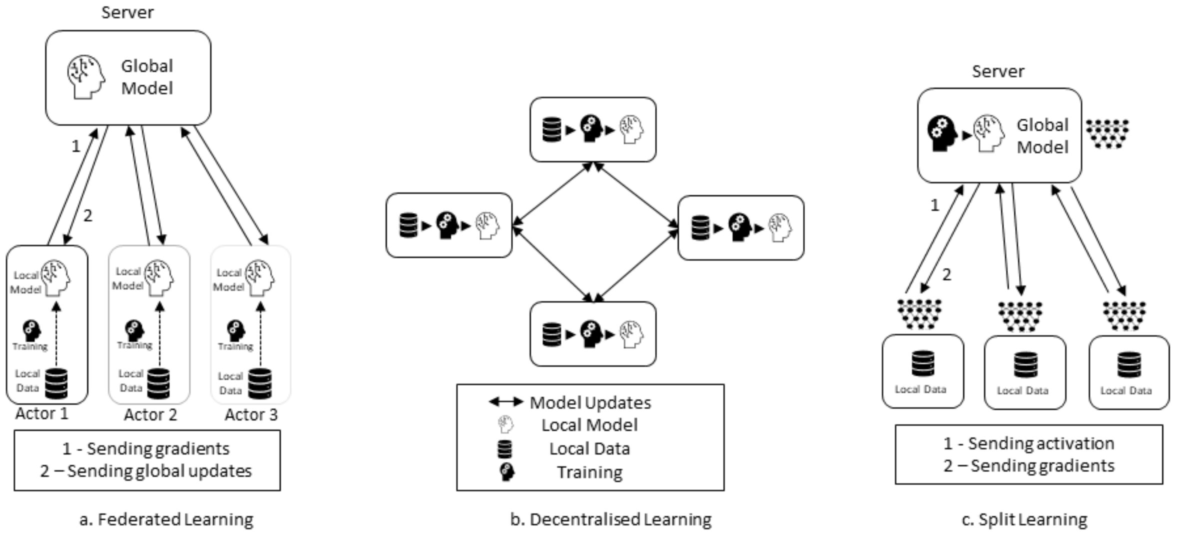


Figure 3-30 Distributed AI techniques [SKK+22].

It is important to remark that *Collaborative* AI techniques should not be considered learning paradigms by themselves but, as different implementation approaches that can rely other ML algorithms (see [HEX-D52] - FL as a Service). Bellow, the main *Distributed/Collaborative* AI techniques are described:

- **Federated Learning:** FL is a *Distributed* ML technique that allows data-owners to work with a common trustworthy server to collaboratively train an AI model without disclosing their data to the other parties, even the server. The server initialises the training and distributes the initial model to data-owners. Afterwards, the model is trained locally using the sensitive local data from each data-owner. For global model aggregation, the server receives updated parameters only from the local data-owners' models, and it is the only party allowed to update parameters in the global model. Until the training ends, these actions are repeated [ZXB+21].
- **Decentralised Learning:** Unlike FL, in decentralized learning the comprising nodes follow a peer-to-peer communication schema. Thus, avoiding a global centralised server and allowing each node to store its own data and perform its own learning. In each cycle of the decentralised AI algorithm, each node conducts a local update, and updates are shared with the neighbours. The global state of the model is achieved when the local models converge to that desired state. As it can be seen in Figure 3-30(b), the data in decentralized learning is fully distributed and there is no need of a common/centralised infrastructure. However, a centralized entity may assign tasks within this distributed technique [KMA+21], i.e., a M&O NF requesting a specific algorithm to be used.
- **Split Learning:** Focuses on training deep learning models between various cooperative parties without sharing any training data or in-depth information of the underlying AI model. In this approach, each party is in charge of training a Deep Neural Network (DNN) up to a layer known

as the “cut” layer. Then, the outputs of this layer are sent to a trustworthy server that back-propagates them until its cut layer. The gradients are sent back, from the server’s cut layer to the clients and this process goes on until convergence is achieved. It is important to remark that there is no access allowed between the server’s and the parties’ models. [TCC+20] demonstrates that SpL provides better privacy than FL and reduces the workload on the clients.

Distributed AI/ML techniques come with a wide range of advantages when facing the ecosystem of 6G networks, however it is important to consider that they also come with various challenges, as detailed by McMahan et al. in [MER+17], particularly in multi-stakeholder environments: *(i)Unbalanced data size*, local training data size may vary for each party and this may lead to data generated by a party overcoming the rest (e.g., the main operator in a given geographical area); *(ii) Communication Constraints,* synchronization issues between clients, extreme-edge devices availability or sudden dropouts, reachability, wireless noise… All may affect the overall algorithm performance. *(iii) Privacy & Security,* the threat model in collaborative/distributed AI needs to cope with greater potential risks than with centralised techniques, although the approach comes with better privacy benefits, if the parameters and the architecture are not properly protected an adversary may reconstruct the source data (i.e., membership inference attacks [NSH19], model inversion/extraction attacks [AM18, OSF19], poisoning attacks [AM18], etc.). *(iv) Data Distribution*, collaborative AI/ML techniques require the data to be identically and independently distributed for training samples as it ensures unbiased full gradient estimations, especially for heterogeneous multi-device configurations [ZLL+18]. Thereupon, considering which distributed AI/ML technique is going to be employed for AI-based frameworks should be an important design step as it could have a huge impact in the overall architecture. For instance, in the AIaaS Framework presented on Figure 3-2 a Decentralised Learning approach will require to split the “AI Training” and “AI Model Repository” into several peer-based logical modules (i.e., several vendors are working together but they prefer to avoid the existence of global centralized repositories or training nodes). On the other hand, if the Split Learning approach is taken the AIaaS framework could be left as it is (i.e., involved parties agree to have a global “AI Model Repository” at an operator’s premises).

3.2.7.3 Privacy supporting protocols

As already stablished, future 6G mobile networks will be quite device-diverse, even more than 5G networks, and multiple network components will not belong to the same stakeholder. In the context of AI applied to 6G networks, data-sets contents may include personally identifiable information (e.g., operator databases, application logs, etc.) or other private indicators that should be stored and processed. During the AI model life-cycle, multiple stages may rise privacy issues in such environments with a negative impact on the AI algorithm. [SKK+22] groups these privacy concerns into three main categories: *(i) Privacy for Training Data,* protecting training data from unauthorised access; *(ii) Privacy for Model Inference,* protection of the model parameters and query data; and *(iii) Protection of the Model,* models can be considered as intellectual properties and, thus, extracting or accessing the model without proper authorisation may lead to bad business results. Consequently, effective protocols should be used to face the challenges that may rise related to privacy. In this regard, [HEX-D62] recommends researching FL combined with Homomorphic Encryption (HE) as a potential supporting protocol. In this section, this approach will be further analysed jointly with other privacy enhancement techniques.

Homomorphic Encryption allows to perform operations on the cipher-data before unencrypting them or, even without requiring access to the private key. Data is encrypted using a public key, HE systems' algebraic structure enables functions to be applied directly to the encrypted data, only the party who holds the private key may access the outcome after applying the functions to the encrypted resource. HE is particularly helpful for computing operations using private data that might be stored by external third-parties (i.e., MNO sharing its data to a vendor to train an AI model, national health care sharing its patient records for a national AI model, FL, etc.). Basically, data can be sent to an external third-party cloud-storage service and be processed there, while still encrypted after having been homomorphically encrypted. Additionally, in the case of collaborative/distributed AI, HE opens the door to sharing training cipher-data by one party and others processing it without learning anything about the training data of the other parties e.g., FL training (see Section 3.2.7.2). Nonetheless, HE comes with some drawbacks [YHL+19]: High computational cost, large storage overhead constraints and a required trust authority.

In parallel, Secure Multi-party computation (SMC) poses an alternative approach to HE for distributed AI cases. In this protocol, each involved party computes a joint function without revealing their inputs to each other, thus there is no need for a trusted server as the involved parties only learn what is given in the output. For instance, on 6G networks, the data shared by MNOs or vendors (data owners) would act as the input for the common function and the output from the joint function could come in the form of inference results for the given inputs instead of the actual model. The main generic SMC protocol in literature, up to the date, is known as GMW [GMW19], which is based on the two-party SMC previous protocol proposed in [Yao86], and it represents the joint function as a set of *XOR* and *AND* logical gates. GMW might be used in the context of collaborative AI/ML, however, due to the high computation and communication costs, the implementation of privacy-preserving AI/ML models by employing these approaches is not very realistic. Custom SMC protocols have been made available to improve privacy in distributed AI/ML scenarios [BIK+17]. For instance, in the case of FL, resolving the secure aggregation of weight updates can be sufficient to stop the leakage of sensitive information (see 3.2.7.2).

Finally, two more technologies are worth mentioning, Confidential Computing (CC) and Differential Privacy (DP). The former, CC, generates a secure environment for running applications by using specialised HW that allows the creation of trusted execution environments via isolated-protected memory regions so that they cannot be accessed directly from RAM. CC has been widely applied to AIaaS but there is still a lot to be researched if related to collaborative/distributed AI. An example of its application was carried out by Intel and Penn in [IP22], using *Intel SGX* in a FL scenario for medical imaging. The latter, DP, is a relatively new approach that consists of performing data anonymisation using several mathematical definitions [SBE+19]. In the context of collaborative/distributed AI there are some scarce examples [PAE17, MRT+18].

4 Flexible network

Flexible network intend to enable extreme performance and global service coverage, while they can also achieve scalability to avoid overprovisioning when and where it is not needed. This deliverable D5.3 comes to detail more the research topics addressed in D5.2, while providing some experimental evidence based on the evaluation of the proposed methodologies and while trying to address the diverse research topics under the prism of a unified Flexible network vision.

Starting from top to down, we start with a conceptual map of the Flexible network areas of research in Hexa-X. This is presented in Figure 4-1. The idea is to provide an overall big picture and vision of the Flexible network vision and to explain how the different research topics and enablers designed and/or developed in Hexa-X contribute to this vision. Therefore, the Flexible network area is divided in three domains: the **network of networks** domain, the **network functions** domain and finally the **interfaces/transport** domain; then, each of the research topic/enabler discussed in this deliverable (but also in previous deliverables, mainly [HEX-D52]) are mapped to this conceptual map. In this chapter we focus on the network function layer.

The blue boxes in Figure 4-1 are abilities or functions already available today with current mobile systems, such as V2X, mobility and necessary interfaces for this. The green boxes represent the new areas developed in Hexa-X. Further on, we have focused on three different areas (see Network of network domain in Figure 4-1) for developing a more flexible network for 6G. The network of network areas are the Satellite (NTN) area, the mobility and finally the ad hoc mesh network area.

4.2

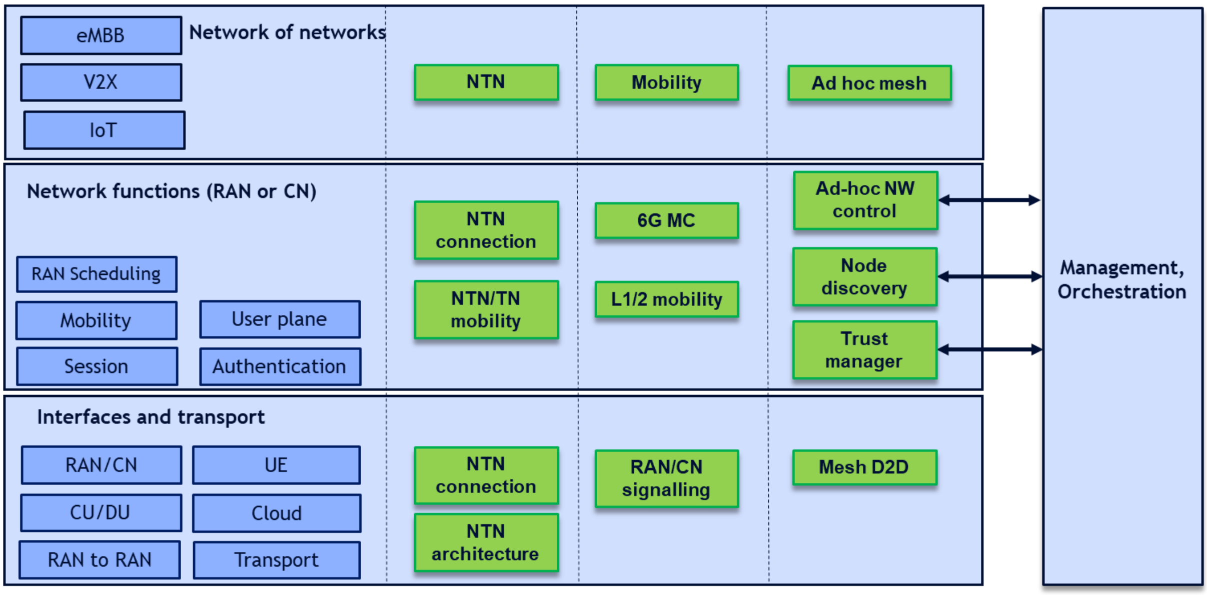


Figure 4-1 Flexible network areas in Hexa-x.

The mobility area for 6G is discussed in Section 4.1. The aim with the mobility is both reliability and a flexible use of available spectrum using 6G multi-connectivity and L1/L2 mobility.

The mesh networks aim for better flexibility when it comes to extreme capacity and coverage on demand basis (see Section 4.2). Section 4.2 continues the research and development activities related to how we can form 6G flexible topologies and local structures (with ad hoc computing and networking nodes of heterogeneous technology) as coordinated extensions of infrastructure (temporary), while operating in highly dynamic environments and different administrative domains under mutual trust. From the components introduced in [HEX-D52], we focus on the Adhoc NW Control component, which is responsible to select the best possible nodes for fulfilling the data flows and/or the computation needs under the application requirements (e.g., low latency, security) posed by the M&O layer. As a first approach, we select a centralized scheme, where the Adhoc NW Control is residing in a central node. Future work is considered for distributed schemes and also to real-time adaptations. The problem statement and formulation and the proposed solution approach are detailed. These are the basis for the FLEX-TOP demonstrator that considers indicative scenarios of a remote area in need of excessive capacity and the need to serve diverse devices (sensors or robots) through a layer of access points being served by some sinks (e.g., terrestrial or NTN etc).

Section 4.2.5 goes one step beyond and demonstrates how the proposed architecture for enabling B5G/6G flexible topologies and local structures could be fully integrated with the Hexa-X M&O architecture. This section focuses on filling the gaps left in D5.2 regarding a full integration of both architectures, by providing a final architectural mapping and detailed insights of the integration of the buildings blocks (BBs) of the D2D architecture and the WP6 M&O architecture modules. This mapping is thoroughly addressed in the different M&O architectural layers, namely *Service Layer*, *Network Layer*, *Infrastructure Layer*, *API Management Exposure Layer, API Management Exposure*, *Design Layer*.

Section 4.3 addresses how NTN enable 6G networks towards provisioning of network resources anytime anywhere, thus contributing to targeting the theoretical limit of 100% network availability, overcoming the problems in complex and rural areas, where terrestrial networks are not a viable solution. This takes the architecture of 6G towards the so-called three-dimensional networking, where satellites, High-Altitude Platform Station (HAPS, and aerial platforms in general are seamlessly integrated in the network with the terrestrial ones. This is also conceptually connected to the FLEX-TOP demonstrator since the latter considers remote areas in need of excessive capacity and the need to serve diverse devices through a layer of access points being served by both terrestrial networks and NTN.

There are some promising contributions in the [HEX-D5.2] that are not further detailed in this document, but it deserves to be mentioned as they are part of the overall flexible network vision. The contribution comprises a modular approach to flexible network integration (see [HEX-D52]), which allows the formation of a scalable and decentralised 6G system. Such a system is created out of multiple, self-managed functional elements called Functional Domains (FDs) that can be of the same or different types (access, transport, etc.), of the same or different technology (4G (E-UTRA), 5G NR (New Radio), WiFi, etc.). This research topic is not illustrated explicitly in the figure above, since its implementation provides the framework based on which the other Flexible network enablers interface and collaborate among each other and with the Management and Orchestration.

4.1 Network of network mobility

Figure 4-2 shows an example of the expected 6G Network of networks with a wide range of different cell types and frequencies as well as different type of networks interworking with each other. In Figure 4-2, the macro cells are using low frequencies of around 1-2 GHz and using high mast to achieve a very wide coverage. The 6G networks will also include smaller cells, likely using higher frequencies such as mid-band (3-6 GHz) or mmWave bands (around 30 GHz), with more spotty coverage. Finally, we will have very spotty coverage of upper mmWave nodes, around 10-100 m cell radius[HEX-D2.1].

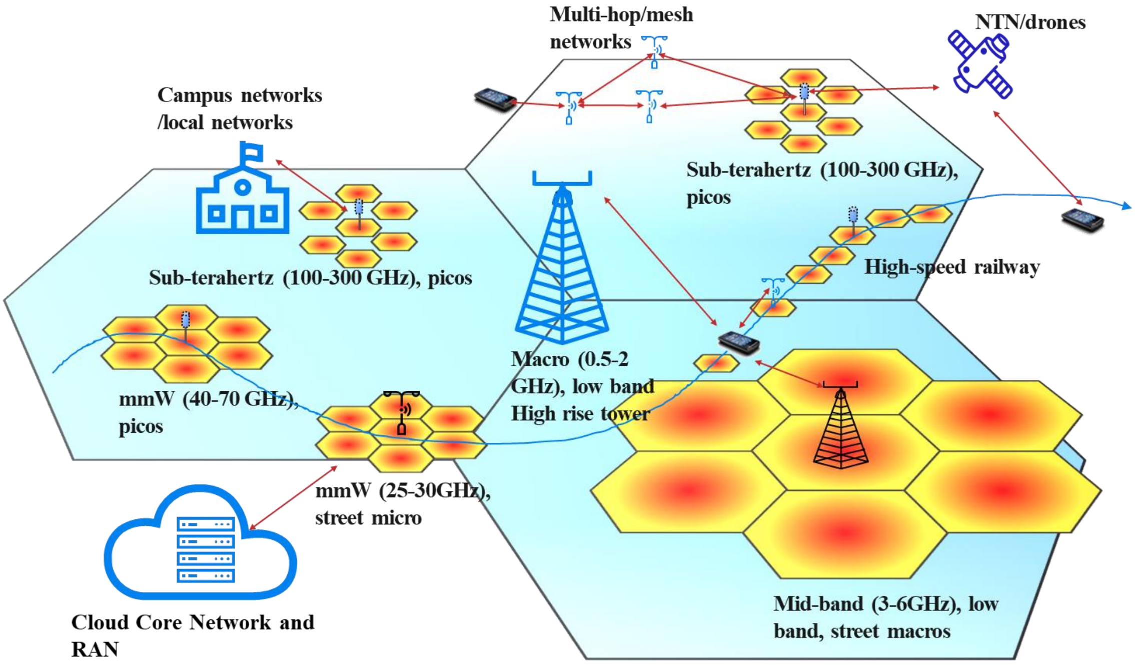

Multi-hop/mesh networks
NTN/drones
Campus networks /local networks
Sub-terahertz (100-300 GHz), picos
Sub-terahertz (100-300 GHz), picos
High-speed railway
mmW (40-70 GHz), picos
Macro (0.5-2 GHz), low band High rise tower
mmW (25-30GHz), street micro
Cloud Core Network and RAN
Mid-band (3-6GHz), low band, street macros

Figure 4-2 The 6G Network of networks will include wide range of cell types, frequencies, and deployments.

There is no single mobility solution for these different networks. For example, using mmW or sub-terahertz networks, one feasible mobility solution can be a L1/2 mobility concept together with Distributed MIMO (D-MIMO), briefly outlined in [HEX-D51]. The L1-mobility system relies on a system with several access points (APs) connected to a central unit (CU) via high-capacity fronthaul transport network. In a region, all the APs connected to the particular CU typically utilize same resources but without fixed cell borders, which is referred to as a MIMO cluster area. Mobility is handled at the Physical (PHY) and Medium Access Control (MAC) layer. This means that the UE does not need to update the Radio Resource Control (RRC) configuration while in the pre-defined area, the UE continues to use the same configuration as before [ECR+21]. Ideally, the UEs in the MIMO cluster area are connected to all APs. However, for complexity reasons and resource utilization, it may be beneficial that the UEs only connect to a subset of the APs, which is referred to as the UE's AP cluster area [HEX-D23]. This means that the UE must still select one or multiple APs in the area best suited for transmission and reception.

Another important feature for mobility is the multi-connectivity feature. When 5G was developed, the idea was to allow a gradual transition from 4G (LTE) to 5G (NR). This was achieved by leveraging on the 4G feature known as dual connectivity where a UE could be connected to two different base stations at the same time. This was then extended so that the UE could be connected to both LTE and NR at the same time (a.k.a. E-UTRA-NR Dual Connectivity, EN-DC) and use either 4G CN (Enhanced Packet Core, EPC) or 5G CN (5GC). However, since this opened several architecture options, this solution became complex. In [HEX-D52] a new 6G multi-connectivity proposal is proposed. To simplify the solution, the number of architecture options should be limited, e.g., by only allowing MC between 6G enabled base stations. The new 6G MC solution should replace the current DC and CA solutions by combining the best features to be able to handle both extreme reliability and excellent flexibility. The new solution combines current CA's ability to decouple UL and DL and DC's ability to utilize nodes located in different geographical locations. The new concept should also support "in-active" connections. The inactive connections can be activated quickly if these connections become good enough, using fast (re)activation of the connections based on volume threshold or similar. One way to enable this is to allow Conditional handover (CHO) and to use a faster L1/L2 signalling.

For the 5G and 6G migration, a possible efficient solution is to employ dynamic spectrum sharing (DSS), which already exist between 4G and 5G. DSS requires less coordination between the nodes compared to e.g., EN-DC and since the UE can use one radio at a time it simplifies the UE implementation. Since NR relies less on always-on reference signalling compared to LTE, it will be easier to share spectrum resources between 5G and 6G. Therefore, Dynamic Spectrum Sharing between 5G and 6G can be an efficient alternative to an 5G-6G EN-DC solution already from the start.

4.2 Adhoc network control for a D2D mesh network and management

[HEX-D52] presented a first architecture to enable the creation of a continuous service environment, in which ad hoc computing and networking nodes of heterogeneous technology collaborate, while operating in highly dynamic environments and different administrative domains under mutual trust. This architecture (Figure 4-3) comprised all the required components in order to form B5G/6G flexible topologies and local structures (nodes with networking, incl. Ad hoc, and computing resources, terminated at edge node) as coordinated extensions of infrastructure (temporary).

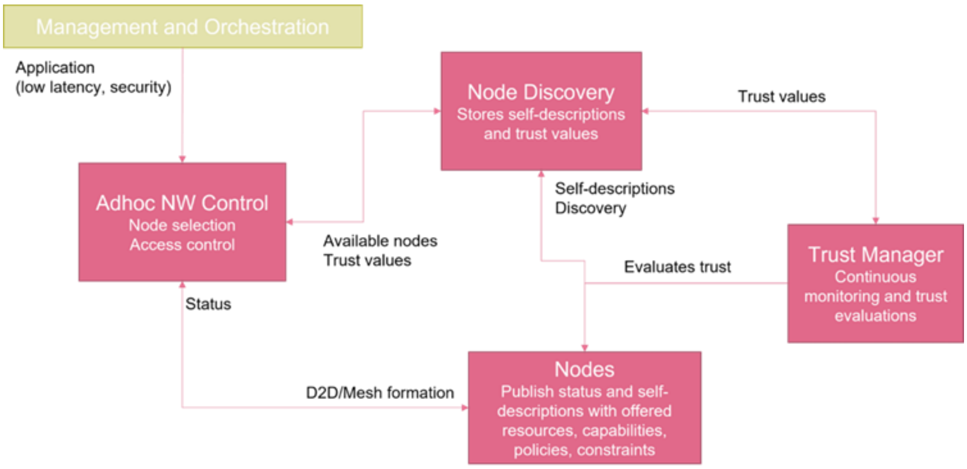


Figure 4-3 High level architecture for D2D and Mesh networks.

These enablers resulted in the following main research topics.

- Selection of nodes and far-edge devices that will be admitted in the “ad hoc” network formation;
- how much “trusted” is a node in order to be part of the D2D/mesh network.
- Unified modelling of far-edge nodes and devices, in terms of network and computational resource characteristics, capabilities, and constraints.
- Integration with network and service orchestration for seamless management, control, and enforcement.
- Discovery of nodes and far-edge devices (including synchronization aspects for capabilities advertisement).

The focus in D5.3 relies on the Adhoc NW Control component. This component selects the best possible nodes and far-edge devices for fulfilling the data flows and/or the computation needs under the application requirements (e.g., low latency, security) posed by the M&O layer, depending on specific parameters (e.g., position, signal quality, battery level, availability, reachability, available computational resources, etc.), as captured in the self-descriptions and the trust values. It takes as input both the infrastructure status from the nodes and the applications’ requirements in terms of performance (e.g., low latency) and security, as derived by the M&O. After the selection process, it configures the

D2D/Mesh formation among the selected nodes. The Adhoc NW Control function can be either in a central node (e.g., master node) or even distributed in the nodes.

In this deliverable, we will focus on the design and implementation of the Adhoc NW Control function in a central node. The management entity is assumed to be collocated.

4.2.1 Motivation and Goal

We imagine a remote area in need of excessive capacity. This area can be an agricultural context, in which we need to collect data from some ground sectors. Alternatively, it can be robots that operate in a critical situation and need connectivity.

Our aim is to serve these devices (sensors or robots) through a layer of access points. Consequently, we would like to have the access points being served by some sinks (e.g., terrestrial or NTN etc.). Here we refer to the first level (traffic sources to access points).

4.2.2 Problem statement

The problem statement can be defined as follows.

Given

- A set of traffic sources, TS.
- In each traffic source *I* of TS,
    - the load generated *Li* (load referring to communication resources, data/computing resource requirements, etc.)
    - the energy that it will consume from a server for computing/communication, denoted as *Ei*
- A set of candidate access points (locations), APs.
    - In each "location" there can be an access point.
    - Access points can be drone mounted or moving on ground.
- Each access point j of AP is defined via its
    - capacity ***Cap(j),***
    - trust index ***T(j),***
    - energy capacity ***CapE(j),***
    - cost of using an Access point (location), denoted as ***K(j).***
- The cost of interconnecting each traffic source i with an action point j, ***c(i,j)***
    - This is associated to the distance and therefore the power that needs to be expended.
    - Also, to the bandwidth allocated for the specific link (s.t. the capacity constraints of the AP). Intuitively,
        - the power consumption scales with the bandwidth of the link.
        - also associated to the source load Li (the required bandwidth of the involved links is dependent on the Li requirement to be transmitted from the traffic source to the server/intermediate APs)
    - if the connection is not possible (e.g., due to communication range constraints) the interconnection cost is "infinite"

Find

- The access points that should be included in the solution
- The interconnection of traffic sources to access points

so that the Objective Function (see below) is maximised.

- All the traffic sources must be served by exactly one access point.
- The capacity and energy capabilities of each access point must be respected.

Objective function: Maximize the difference between trustworthiness minus the cost of the selected nodes and interconnections.

The principles, under which the optimization is performed are the following:

- Maximisation of the overall system's trust index.
- Selecting APs (AP locations) with "low cost".
- Selecting interconnections that lead to minimum energy requirements

4.2.3 Problem formulation

In order to formulate the problem, we are doing the following assumptions.

I Traffic source/devices (TSs) need to be served by J Access Points (APs) in a rural area.

Assumptions

- Each AP j has an initial trust index, $\mathrm{T_j}$, a cost of deployment, $\mathrm{K_j}$, a capacity, $\mathrm{Cap_j}$, and an energy capacity, $\mathrm{CapE_j}$.
- Each TS i has a load, I, and an energy it will consume from a server, $\mathrm{E_i}$.

Let $\mathbf{Y} \in \{0,1\}^{\mathrm{Jx1}}$ the vector that denotes which APs are used and $\mathbf{X} \in \{0,1\}^{\mathrm{JxI}}$ the matrix that states which $\mathrm{TS} - \mathrm{AP}$ links are active.

To conclude, let $\mathbf{C} \in \mathrm{R}^{\mathrm{JxI}}$ denote the connection cost matrix of any TS with any AP.

Thus, the total cost the connections induce will be

$$\mathrm{CC} = \sum_{\mathrm{j=1}}^{\mathrm{J}} \sum_{\mathrm{i=1}}^{\mathrm{I}} \mathrm{X_{ji} C_{ji}}$$

If $\mathbf{T} \in \mathrm{R}^{\mathrm{1xJ}}$ denote the initial trust and $\mathbf{K} \in \mathrm{R}^{\mathrm{1xJ}}$ the cost of deployment of all APs, the overall remaining trust of the deployed APs will be

$$\mathrm{RT} = \sum_{\mathrm{j=1}}^{\mathrm{J}} [\mathrm{T_j} - \mathrm{K_j}] \mathrm{Y_j}$$

Object: maximize $\mathrm{RT} - \mathrm{CC}$

s.t.

$$\sum_{\mathrm{j=1}}^{\mathrm{J}} \mathrm{X_{ji}} = 1$$

$$\sum_{\mathrm{i=1}}^{\mathrm{I}} \mathrm{X_{ji} L_i} \leq \mathrm{Cap_j Y_j}$$

$$\sum_{\mathrm{i=1}}^{\mathrm{I}} \mathrm{X_{ji} E_i} \leq \mathrm{CapE_j Y_j}$$

$$\mathrm{X_{ji}, Y_j} \in \{0,1\} \mathrm{I} \forall \mathrm{i, j}$$

$$\mathrm{C_{ji}, T_j, K_j}, I, \mathrm{IE_i, Cap_j, Ca}I \geq 0 \ \forall \mathrm{i, j}$$

Linear programming techniques can be applied to solve this optimization problem.

If $\mathbf{x} = [\mathrm{X_{11}\ X_{12} \ldots X_{1I}\ X_{21}}\ X_{22} \ldots \mathrm{X_{J1} X_{J2} \ldots X_{JI}}] =$

$[\mathrm{x_1\ x_2 \ldots x_I\ x_{I+1}\ x_{I+2} \ldots x_{(JI-I)+1}\ x_{(JI-I)+2} \ldots x_{JI}}] \in \{0,1\}^{\mathrm{JI}}$

Then, expanding the first constraint we get

$$\sum_{j=1}^{J} \mathrm{X_{ji}} = \mathrm{X_{1i}} + \cdots + \mathrm{X_{Ji}} = \begin{bmatrix} \mathrm{X_{11}} \\ \mathrm{X_{12}} \\ \vdots \\ \mathrm{X_{1I}} \end{bmatrix}^{\mathrm{T}} + \cdots + \begin{bmatrix} \mathrm{X_{J1}} \\ \mathrm{X_{J2}} \\ \vdots \\ \mathrm{X_{JI}} \end{bmatrix}^{\mathrm{T}} =$$

$$\begin{bmatrix} \mathrm{X_{11}} + \mathrm{X_{21}} + \cdots + \mathrm{X_{J1}} \\ \mathrm{X_{12}} + \mathrm{X_{22}} + \cdots + \mathrm{X_{J2}} \\ \vdots \\ \mathrm{X_{1I}} + \mathrm{X_{2I}} + \cdots + \mathrm{X_{JI}} \end{bmatrix} \leq 1$$

Which can be rewritten as

$x_1 + \mathrm{x_{I+1}} \ldots + \mathrm{x_{(JI-I)+1}} \leq 1$

$\mathrm{x_2} + \mathrm{x_{I+2}} \ldots + \mathrm{x_{(JI-I)+2}} \leq 1$

…

$\mathrm{x_I} + \mathrm{x_{2I}} + \cdots + \mathrm{x_{JI}} \leq 1$

Expanding the second, we get

$$\sum_{i=1}^{I} \mathrm{X_{ji}L_i} \leq \mathrm{Cap_j Y_j} \Rightarrow \mathrm{X_{j1}L_1} + \cdots + \mathrm{X_{jI}L_I} = \begin{bmatrix} \mathrm{X_{11}} \\ \mathrm{X_{21}} \\ \vdots \\ \mathrm{X_{J1}} \end{bmatrix} \mathrm{L_1} + \cdots + \begin{bmatrix} \mathrm{X_{1I}} \\ \mathrm{X_{2I}} \\ \vdots \\ \mathrm{X_{JI}} \end{bmatrix} \mathrm{L_I} =$$

$$\begin{bmatrix} \mathrm{X_{11}L_1} + \mathrm{X_{12}L_2} + \cdots + \mathrm{X_{1I}L_I} \\ \mathrm{X_{21}L_1} + \mathrm{X_{22}L_2} + \cdots + \mathrm{X_{2I}L_I} \\ \vdots \\ \mathrm{X_{J1}L_1} + \mathrm{X_{J2}L_2} + \cdots + \mathrm{X_{JI}L_I} \end{bmatrix} \leq \begin{bmatrix} \mathrm{Cap_1} \\ \mathrm{Cap_2} \\ \vdots \\ \mathrm{Cap_J} \end{bmatrix} \mathrm{Y_j}$$

Which can be rewritten as

$\mathrm{x_1L_1} + \mathrm{x_2L_2} + \cdots + \mathrm{x_IL_I} \leq \mathrm{Cap_1}$

$\mathrm{x_{I+1}L_1} + \mathrm{x_{I+2}L_2} + \cdots + \mathrm{x_{2I}L_I} \leq \mathrm{Cap_2}$

…

$\mathrm{x_{(JI-I)+1}L_1} + \mathrm{x_{(JI-I)+2}L_2} + \cdots + \mathrm{x_{JI}L_I} \leq \mathrm{Cap_J}$

The same applies for the third constraint.

Therefore, the constraints are

$x_1 + \mathrm{x_{I+1}} \ldots + \mathrm{x_{(JI-I)+1}} \leq 1$

$\mathrm{x_2} + \mathrm{x_{I+2}} \ldots + \mathrm{x_{(JI-I)+2}} \leq 1$

…

$\mathrm{x_I} + \mathrm{x_{2I}} + \cdots + \mathrm{x_{JI}} \leq 1$

$\mathrm{x_1L_1} + \mathrm{x_2L_2} + \cdots + \mathrm{x_IL_I} \leq \mathrm{Cap_1}$

$\mathrm{x_{I+1}L_1} + \mathrm{x_{I+2}L_2} + \cdots + \mathrm{x_{2I}L_I} \leq \mathrm{Cap_2}$

…

$$\mathrm{x}_{(\mathrm{JI}-\mathrm{I})+1}\mathrm{L}_1 + \mathrm{x}_{(\mathrm{JI}-\mathrm{I})+2}\mathrm{L}_2 + \cdots + \mathrm{x}_{\mathrm{JI}}\mathrm{L}_\mathrm{I} \leq \ \mathrm{Cap}_\mathrm{J}$$

$$\mathrm{x}_1\mathrm{E}_1 + \mathrm{x}_2\mathrm{E}_2 + \cdots + \mathrm{x}_\mathrm{I}\mathrm{E}_\mathrm{I} \leq \ \mathrm{CapE}_1$$

$$\mathrm{x}_{\mathrm{I}+1}\mathrm{E}_1 + \mathrm{x}_{\mathrm{I}+2}\mathrm{E}_2 + \cdots + \mathrm{x}_{2\mathrm{I}}\mathrm{E}_\mathrm{I} \leq \ \mathrm{CapE}_2$$

…

$$\mathrm{x}_{(\mathrm{JI}-\mathrm{I})+1}\mathrm{E}_1 + \mathrm{x}_{(\mathrm{JI}-\mathrm{I})+2}\mathrm{E}_2 + \cdots + \mathrm{x}_{\mathrm{JI}}\mathrm{E}_\mathrm{I} \leq \ \mathrm{CapE}_\mathrm{J}$$

$$\mathbf{A}_1 = \begin{bmatrix} \underbrace{1}_{\mathrm{I-I\ zeros}} \ \underbrace{0\ 0 \ldots 0}_{\mathrm{I-1\ zeros}}\ 1\ \underbrace{0\ 0 \ldots 0}_{\mathrm{I-1\ zeros}} \ldots 1\ \underbrace{0\ 0 \ldots 0}_{\mathrm{I-1\ zeros}} \\ \underbrace{0}_{\mathrm{I-(I-1)zeros}}\ 1\ \underbrace{0\ 0 \ldots 0}_{\mathrm{I-1\ zeros}}\ 1\ \underbrace{0\ 0 \ldots 0}_{\mathrm{I-1\ zeros}}\ 1\ \underbrace{0\ 0 \ldots 0}_{\mathrm{I-2\ zeros}} \\ \vdots \\ \underbrace{0\ 0 \ldots 0}_{\mathrm{I-1\ zeros}}\ 1\ \underbrace{0\ 0 \ldots 0}_{\mathrm{I-1\ zeros}}\ 1\ \underbrace{0\ 0 \ldots 0}_{\mathrm{I-1\ zeros}}\ \underbrace{1}_{\mathrm{I-I\ zeros}} \end{bmatrix}_{\mathrm{IxJI}}$$

$$\mathbf{A}_2 = \begin{bmatrix} \mathrm{L}_1\ \mathrm{L}_2 \ldots\ \mathrm{L}_\mathrm{I}\ \underbrace{0\ 0 \ldots 0}_{\mathrm{JI-I\ zeros}} \\ \underbrace{0\ 0 \ldots 0}_{\mathrm{I\ zeros}}\ \mathrm{L}_1\ \mathrm{L}_2 \ldots\ \mathrm{L}_\mathrm{I}\ \underbrace{0\ 0 \ldots 0}_{\mathrm{JI-2I\ zeros}} \\ \vdots \\ \underbrace{0\ 0 \ldots 0}_{\mathrm{JI-I\ zeros}}\ \mathrm{L}_1\ \mathrm{L}_2 \ldots\ \mathrm{L}_\mathrm{I} \end{bmatrix}_{\mathrm{JxJI}}$$

$$\mathbf{A}_3 = \begin{bmatrix} \mathrm{E}_1\ \mathrm{E}_2 \ldots\ \mathrm{E}_\mathrm{I}\ \underbrace{0\ 0 \ldots 0}_{\mathrm{JI-I\ zeros}} \\ \underbrace{0\ 0 \ldots 0}_{\mathrm{I\ zeros}}\ \mathrm{E}_1\ \mathrm{E}_2 \ldots\ \mathrm{E}_\mathrm{I}\ \underbrace{0\ 0 \ldots 0}_{\mathrm{JI-2I\ zeros}} \\ \vdots \\ \underbrace{0\ 0 \ldots 0}_{\mathrm{JI-I\ zeros}}\ \mathrm{E}_1\ \mathrm{E}_2 \ldots\ \mathrm{E}_\mathrm{I} \end{bmatrix}_{\mathrm{JxJI}}$$

And the total **A** matrix will be $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \end{bmatrix}_{(\mathrm{I}+2\mathrm{J})\mathrm{xJI}}$.

The vector **b** will be

$$\mathbf{b} = \left[1\ 1 \ldots 1\ \mathrm{Cap}_1\ \mathrm{Cap}_2 \ldots \mathrm{Cap}_\mathrm{J}\ \mathrm{CapE}_1\ \mathrm{CapE}_2 \ldots \mathrm{CapE}_\mathrm{J}\ \right]^\mathrm{T}_{1\mathrm{x}(\mathrm{I}+2\mathrm{J})}$$

Using LP, we need to find a vector **x** that maximizes $\mathbf{c}^\mathrm{T}\mathbf{x}$ subject to $\mathbf{Ax} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}.$ The matrix **A** and the vector **b** have been already found using the previous analysis.

4.2.4 Solution approach

The following lines present the rationale to find the optimal solution for our problem, i.e., to maximize the objective function subject to constraints given.

Let

$\mathrm{AP}\ =\ 2$

$\mathrm{TS} = 5$

$\mathrm{T} = [0.9224\ 0.0342]^{\mathrm{T}}$

$\mathrm{K} = [0.0268\ 0.8133]^{\mathrm{T}}$

$\mathrm{Cap} = [0.5884\ 0.7286]^{\mathrm{T}}$

$\mathrm{CapE} = [0.7673\ 0.6806]^{\mathrm{T}}$

$\mathrm{L} = [0.4689\ 0.2870\ 0.1989\ 0.1918\ 0.0050]$

$\mathrm{E} = [0.3233\ 0.0578\ 0.2152\ 0.1110\ 0.1521]$

$\mathrm{C} = \begin{bmatrix} 0.2827 & 0.3711 & 0.7379 & 0.3420 & 0.4548 \\ 0.3456 & 0.9088 & 0.4058 & 0.6027 & 0.2748 \end{bmatrix}$

The objective is to select the matrix X from the set of possible solutions, $\widetilde{\mathrm{X}}$, which not only does satisfy the constrains given but also maximizes the objective function (OF)

$$\max_{\mathbf{X}} \mathrm{OF} \quad \mathrm{s.t.\,constrains}$$

For instance, the matrix $\mathrm{X} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$ is not contained in the set $\widetilde{\mathrm{X}}$ because it violates the first condition which states that each TS must have exactly one connection active. Another example is $\mathrm{X} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$ which does not violate the first condition (each TS is connected to exactly one AP) but it does violate the second condition because the required capacity for the second AP is 0.7608 ($\mathrm{L_1X_{21} + L_2X_{22} + L_3X_{23} + L_4X_{24} + L_5X_{25}} = 0.4689 * 1\ +\ 0.2870 * 1\ +\ 0.1989 * 0\ +\ 0.1110 * 0\ +\ 0.005 * 1$) whereas it should not exceed 0.7286.

Running the algorithm, we obtain 5 feasible solutions for the matrix X which are

$$\mathrm{X_{sol_1}} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix} \leftarrow \left\{\begin{matrix} \mathrm{TS_1} \rightarrow \mathrm{AP_2} \\ \mathrm{TS_2} \rightarrow \mathrm{AP_1} \\ \mathrm{TS_3} \rightarrow \mathrm{AP_2} \\ \mathrm{TS_4} \rightarrow \mathrm{AP_1} \\ \mathrm{TS_5} \rightarrow \mathrm{AP_1} \end{matrix}\right\}$$

$$\mathrm{X_{sol_2}} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix} \leftarrow \left\{\begin{matrix} \mathrm{TS_1} \rightarrow \mathrm{AP_2} \\ \mathrm{TS_2} \rightarrow \mathrm{AP_1} \\ \mathrm{TS_3} \rightarrow \mathrm{AP_1} \\ \mathrm{TS_4} \rightarrow \mathrm{AP_2} \\ \mathrm{TS_5} \rightarrow \mathrm{AP_2} \end{matrix}\right\}$$

$$\mathrm{X_{sol_3}} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \leftarrow \left\{\begin{matrix} \mathrm{TS_1} \rightarrow \mathrm{AP_2} \\ \mathrm{TS_2} \rightarrow \mathrm{AP_1} \\ \mathrm{TS_3} \rightarrow \mathrm{AP_1} \\ \mathrm{TS_4} \rightarrow \mathrm{AP_2} \\ \mathrm{TS_5} \rightarrow \mathrm{AP_1} \end{matrix}\right\}$$

$$\mathrm{X_{sol_4}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix} \leftarrow \left\{\begin{matrix} \mathrm{TS_1} \rightarrow \mathrm{AP_1} \\ \mathrm{TS_2} \rightarrow \mathrm{AP_2} \\ \mathrm{TS_3} \rightarrow \mathrm{AP_2} \\ \mathrm{TS_4} \rightarrow \mathrm{AP_2} \\ \mathrm{TS_5} \rightarrow \mathrm{AP_2} \end{matrix}\right\}$$

$$X_{sol_5} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \leftarrow \begin{Bmatrix} TS_1 \rightarrow AP_2 \\ TS_2 \rightarrow AP_1 \\ TS_3 \rightarrow AP_2 \\ TS_4 \rightarrow AP_1 \\ TS_5 \rightarrow AP_1 \end{Bmatrix}$$

Which lead to the set of possible solutions $\widetilde{X} = \{X_{sol_1}, X_{sol_2}, X_{sol_3}, X_{sol_4}, X_{sol_5}\}$.

Therefore, for the example presented, only 5 (out of the total $2^{APxTS} = 2^{10}$) combinations for the matrix X are allowed. Plotting the OF value for each one of the 5 possible solutions gives us the following figure where we can easily observe that the optimal solution is $\widetilde{X}\{1\} = X_{sol_1} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix} \leftarrow \begin{Bmatrix} TS_1 \rightarrow AP_2 \\ TS_2 \rightarrow AP_1 \\ TS_3 \rightarrow AP_2 \\ TS_4 \rightarrow AP_1 \\ TS_5 \rightarrow AP_1 \end{Bmatrix}$ which gives the OF the value -1.8027.

The figure below (Figure 4-4) depicts the comparison of the different values the OF takes for each one of the valid combinations of the matrix X. This graphical representation is useful for demonstrating purposes as it shows the impact of each set of active connections on the objective function.

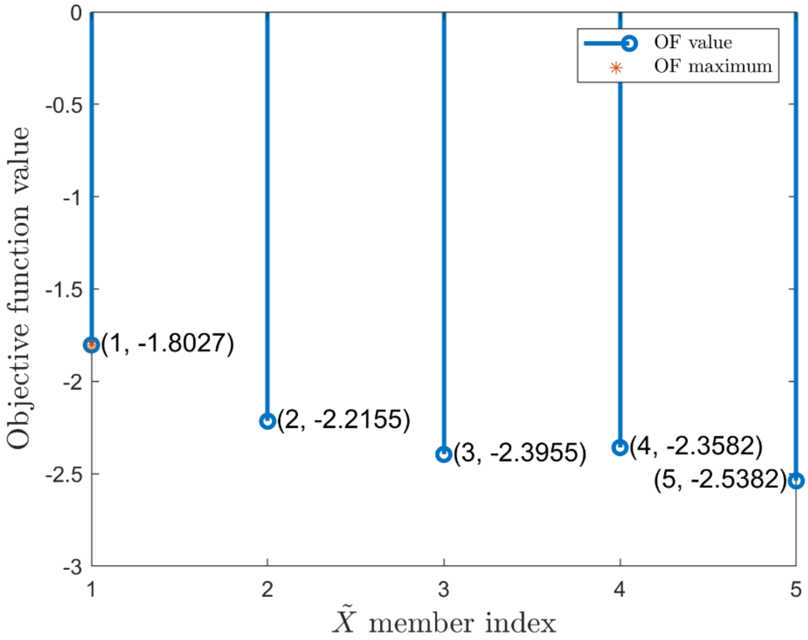


Figure 4-4 Objective function (OF) values for different feasible solutions' inputs.

However, for the sake of clarification, by examining the objective function, we can easily see that it comprises of two distinct KPIs, trust and cost, which are jointly optimized. We can represent the OF in a more general fashion as follows,

OF: $$\max_{X} \sum \text{trust} - \sum \text{cost}$$

Which states that the goal is to maximize the overall system trust by selecting the most appropriate AP nodes that both have high enough trust values and at the same time decrease the total cost.

Note that in order to quantify the KPIs trust and cost, a reverse engineering approach should be followed based on the selected matrix X. For instance, in the above scenario, the selected matrix was $\begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix}$ and thus we conclude that both APs are used which means $\mathrm{Y}_1 = \mathrm{Y}_2 = 1$. Based on the values of the vectors T and K, we find out that the value of the trust is $(0.9224 - 0.0268) * 1 + (0.0342 - 0.8133) * 1 = 0.1166$ and the value of the cost is $\sum\sum \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 0.2827 & 0.3711 & 0.7379 & 0.3420 & 0.4548 \\ 0.3456 & 0.9088 & 0.4058 & 0.6027 & 0.2748 \end{bmatrix} = 1.9193.$

Hence, the resulted values for the trust and the cost are indeed the optimal since the OF takes its maximum value $(0.1166 - 1.9193 = -1.8027)$. It can easily be examined that the use of the other feasible solutions of the matrix X result in suboptimal values for the concerned KPIs.

Figure 4-4 relates to a scenario where multiple APs need to establish connections with many TSs in order for the information to be transferred to the CN (Core Network, but in general we imply any point in which the flexible topology may be terminated or has a link to the rest of the Network of networks world). However, the information needs to get transferred with the minimum cost regarding the errors, latency, etc. Thus, it emerges the need for the proper selection of the APs that the TSs will be connected to.

Given AP trust values, Tj, assigned by the ‘Trust Manager’, cost of deployment Kj, and other costs denoted with Cji, we need to find the matrix X that denotes the links that should be established so as the overall system trust will be maximized. That is, we need to maximize the following objective function given constraints related to the energy, load, etc.

$$\max_{\mathrm{X}} \sum_{\mathrm{j}=1}^{\mathrm{AP}} (\mathrm{T_j} - \mathrm{K_j})\mathrm{Y_j} - \sum_{\mathrm{j}=1}^{\mathrm{AP}} \sum_{\mathrm{i}=1}^{\mathrm{TS}} \mathrm{XC} \quad \mathrm{s.t. constraints}$$

Note that Yj refers to the active APs and is closely related to the matrix X.

In this indicative example of the figure, we used a Mixed-Integer Linear Programming (MILP) algorithm that obtained 5 feasible solutions for the matrix X. The figure portrays the OF values for each one of the 5 possible solutions of the matrix X. The selected one though is the first because it can easily be seen that, by establishing the connections that the first solution suggests, the OF obtains its maximum value.

To ensure complete comprehension of the prior approach, an indicative example along with its graphical illustration is provided in the following lines. Thus, assume a scenario where numerous sensors placed in a rural environment generate data, which need to be collected by an external entity. In order to overcome challenges imposed by static infrastructure solutions, such as high energy consumption, increased bit error rate due to fading, etc., the collection of the information can be gathered on demand with the aid of unmanned aerial nodes. Figure 4-5 presents such a scenario where aerial nodes are collecting the available information from various traffic sources.

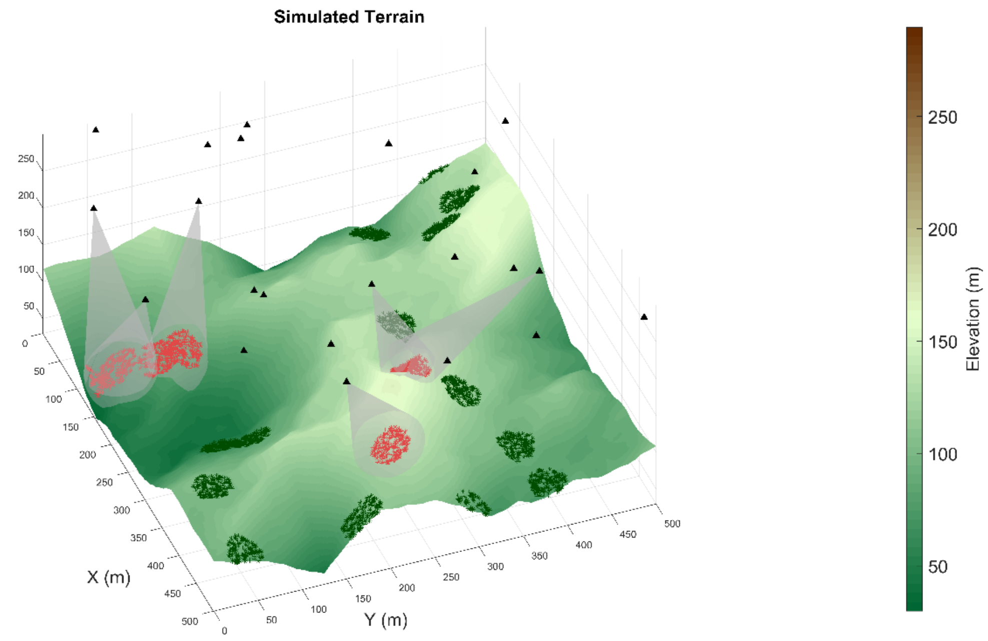


Figure 4-5 Simulated rural environment comprising monitoring/sensing traffic sources served by a number of UAV access nodes.

However, different aerial nodes have different capabilities and cost; hence a proper selection of the nodes to be used should be done in order to account not only for the maximization of the system's trust but also for the minimization of the cost. Leveraging the optimisation problem presented in the previous paragraphs, an example where a subset of the nodes is selected based on the capabilities and the cost of each and their connection with the traffic sources is presented. More specifically, Figure 4-6 illustrates the normalized value of different KPIs/KVIs of interest (deployment cost, inclusion as the percentage of traffic sources served with the required (minimum) bitrates, number of access nodes, utilization and UAV battery consumption) for different solutions of the connection matrix, which do not violate any constraint regarding the energy and capacity limitations.

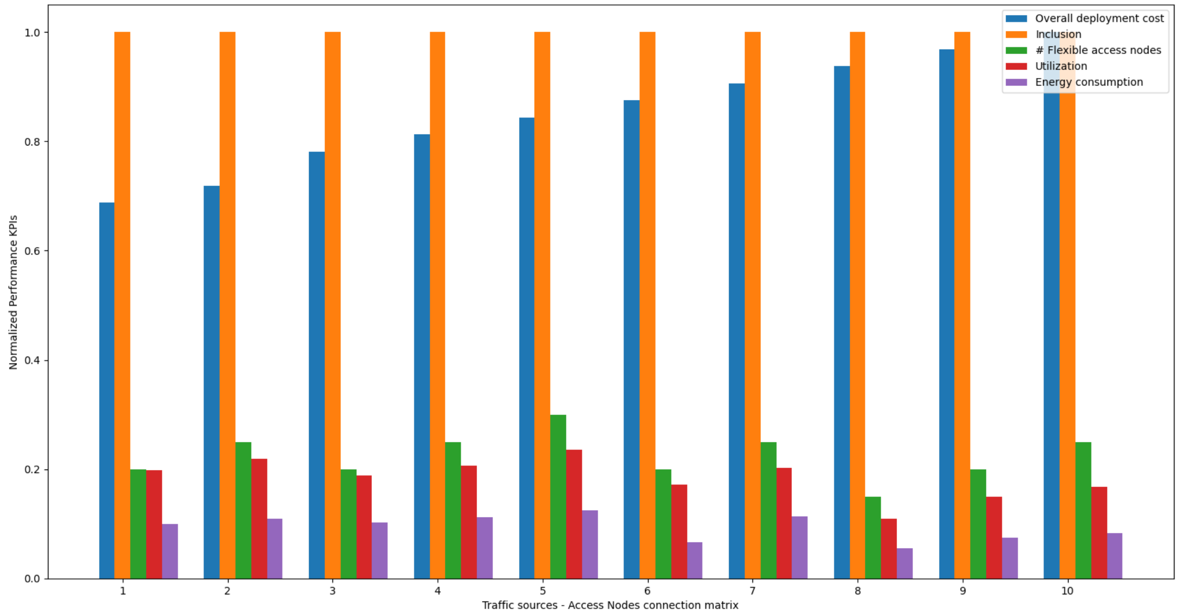


Figure 4-6 Performance KPI/KVI for different traffic sources - access nodes associations.

Note that only the first solution for the matrix of connections is the output of the optimization problem, i.e., the maximization of the objective function presented previously, and thus it is the optimal. The next nine valid combinations of the connection matrix are suboptimal solutions, which lead to deterioration of KPIs/KVIs, such as increased cost, or higher need of commissioned UAVs for serving the traffic sources. Each of them was generated by forcing the optimization algorithm not to produce the previous ones as solutions for comparison reasons. Therefore, selecting the optimal solution the optimization function provides, an overall decrease in cost is made whereas at the same time the needs for connection are satisfied.

4.2.5 D2D mesh network management and orchestration

As described in [HEX-D52], the Management and Orchestration (M&O) of future D2D ad hoc networks (e.g., WANETs or MANETs of any kind) requires to be able to cope with all the constraints associated to this kind of networks i.e., energy efficiency, scalability, trustworthiness, blurry infrastructure boundaries, etc. To that extent, a mapping between the buildings blocks (BBs) of the D2D architecture and the WP6 M&O architecture modules was provided at the *Network Layer*. This mapping helped to have a clearer view of the role that the different D2D architectural BBs have within a M&O architecture and to clarify which of them act as Managing Objects and/or Managed Objects i.e., the M&O relation between the D2D BBs. Besides, it was focused on giving a functional view of the mapped components and a general idea of how, at the *Network Layer,* this per-BB mapping could be approached. Nonetheless, the figure lacks a complete view of the different framework layers (i.e., Service, Network and Infrastructure) that allows the reader to fully understand the integration of both architectures. This section will focus on filling the gaps left in [HEX-D52] regarding a full integration of both architectures, by providing a final architectural mapping and detailed insights of the BBs integration.

It is important to remark that the Hexa-X WP6 M&O architecture has not been designed to have a clear alignment between the architectural design and any particular SDO's standard. This is done on purpose since the goal is to provide an agnostic architectural design that is not explicitly aligned with a single standard (or a few) and that it is abstract enough, to allow for implementations that are aligned with a wide range of pertinent SDOs standards or even possible future standards i.e., CAPIF, ETSI NFV MANO, 3GPP SA2, 3GPP SA5, etc. The key justifications for adopting this strategy are depicted in [HX22-D62] and they can be summed up into the following idea: given that Hexa-X long term goal is to lay the groundwork for the creation of new 6G technology, it is considered risky to strongly align the M&O architectural design with a particular standard (or a few of them). If that standard were not widely adopted, the Hexa-X M&O design might be rendered obsolete before its time. However, this M&O architectural design is abstract enough to be able to be potentially aligned with different SDO standards for any particular implementation.

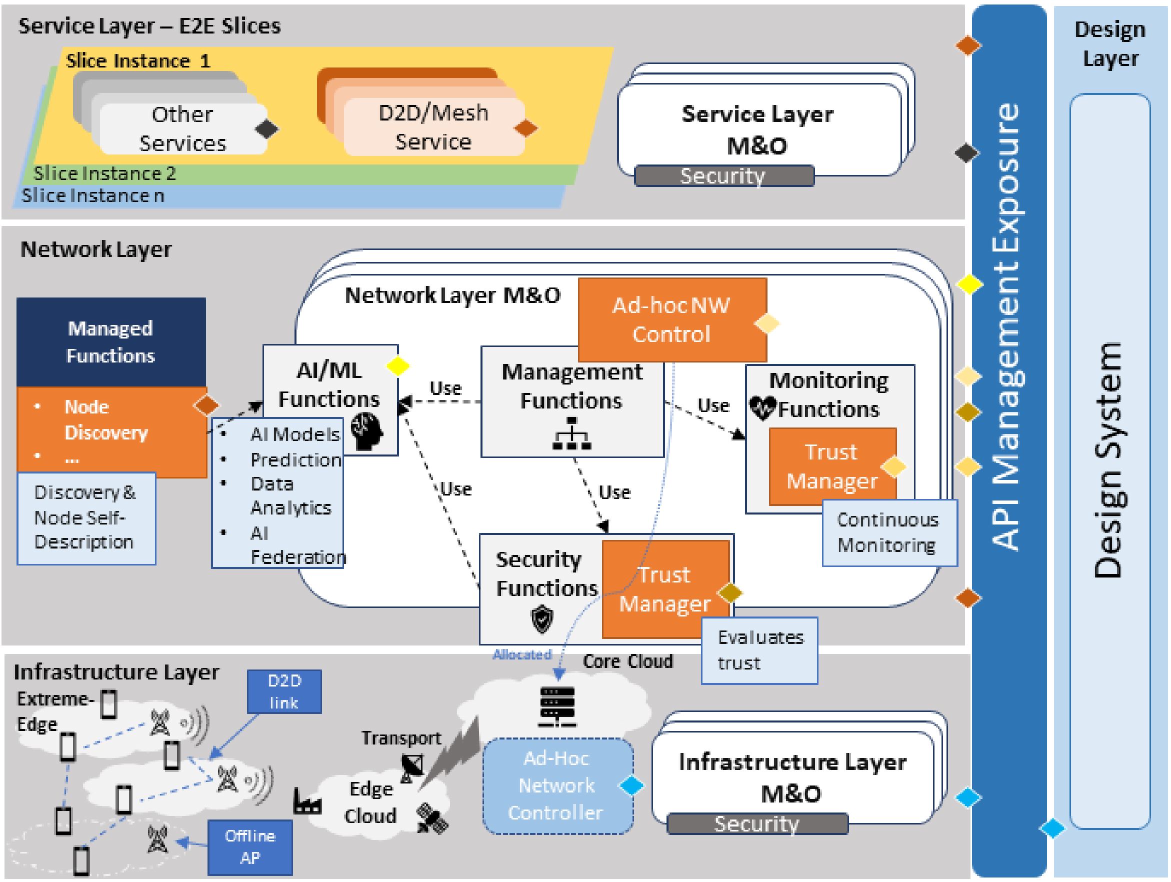


Figure 4-7 D2D Architecture components allocation within the WP6 M&O architecture.

Taking into account the afore-mentioned considerations, Figure 4-7 depicts a full mapping between the D2D architectural BBs proposed for WANET/MANET ad hoc networks in [HX22-D52] and each of the different M&O architectural layers. This figure aims at giving a detailed vision, at every layer, of how the D2D BBs might be allocated within the Hexa-X M&O architecture. There are several improvements, if compared with the previous deliverable figure, in terms of architectural and inter-module relations details:

1. Every layer is composed of *Managing Objects (*i.e., those within each layer "M&O" white module) and *Managed Objects* (i.e., those BBs outside of the M&O white module). This division clarifies the role of each BB and if it is part of the M&O system or if it is managed by it.
2. The *Service Layer* has been added to reflect the capacity of the proposed D2D network architecture to be able to face and deploy several types of services (i.e., it is not limited to a specific type of service). Besides, it also demonstrates its capacity to work with independent Slice Instances and, thereupon, the proposed D2D architecture is capable to create the ad hoc network topologies based on the traffic load, computation, and communication needs. After the service is finished, it once more releases the allocated resources for the given Slice.
3. The *Network Layer* has been modified accordingly to follow the *Managing Vs. Managed Object* fashion of the rest of the layers. Directional arrows have been included to clarify the relation between the D2D BBs that comprise the *Network Layer Managing Functions.* As it can be seen, the only D2D BB that has been mapped outside of the M&O is the "Node Discovery" as it was presented as a potential 3rd party *Managed Function* in the previous deliverable [HX22-D52]. Moreover, two additional relationships have been established: *(i) AI/ML Function and Security Functions*, the "Trust Manager" functions related to security may use *AI/ML* to evaluate trust; *(ii) AI/ML Functions and Node Discovery*, in order to generate

node self-descriptions and discover available modules with optimized discovery algorithms the “Node Discovery” D2D BB may use *AI/ML* techniques to optimize those processes.

4. The *Infrastructure Layer* has been depicted in such a way that there is a clear view of the different infrastructure domains that might be involved in future D2D ad hoc networks (i.e., extreme-edge, edge and cloud) as a compute continuum. Furthermore, in the extreme-edge domain a D2D physical network has been exemplified with different Access Points (APs) and D2D links. Finally, the *Ad Hoc Network Controller* server has been highlighted to remark what would be allocated within its resources, as described in Section 4.2.
5. The *API Management Exposure Layer* has been represented as in the original Hexa-X M&O architecture [HX22-D62], as a cross-layer functional block that enables network elements capability exposure in the various architectural levels both, inside and across, administrative domains. All the network components in the various architectural layers can interact and communicate with one another using this block at a variety of granularity levels while adhering to a unified pattern. This model may be applied with a larger scope to reflect future federation-based interactions in addition to communication among M&O resources. In a nutshell, it imitates the behaviour of the cross-domain integration fabric for Zero-Touch Network and Service Management (ZSM) [zsm-002] and is able to exploit CAPIF concepts and capabilities [23.222].
6. The small, coloured diamonds within each layer represent the endpoints or APIs associated to individual M&O or *Managed* resources and how the *API Management Exposure* functional block could act as a common and unified framework that regulates the exposure and management of the various APIs provided by each BB on each layer and, even more, in different domains i.e., multi-domain federated D2D scenario.
7. The *Design Layer* is one of the main innovations introduced by the Hexa-X M&O architecture. It has been included as it has the capabilities to design, define, model, and distribute software components that may be used to create and run the 6G infrastructure. It exemplifies the implementation of cloud-native concepts in terms of bringing together development and operational teams through the use of DevOps approaches, facilitating how services are delivered and updated with a very high degree of automation.

As it can be seen, this upgraded mapping approach shown in Figure 4-7, demonstrates how the D2D architecture proposed in section could be fully integrated with the Hexa-X M&O architecture.

4.3 NTN and 3D architecture

6G networks are promising the provisioning of network resources anytime anywhere, targeting the theoretical limit of 100% network availability. This means that rural areas, where terrestrial networks are not the main viable solution, NTN may play a pivotal role. This takes the architecture of 6G towards the so-called three-dimensional networking, where satellites, HAPS, and aerial platforms in general are seamlessly integrated in the network with the terrestrial infrastructure.

However, 3D networking design can follow different approaches according to the area targeted and to the specific users on-ground, requiring specific values of KPIs. The following text shows the two different and complementary solutions. First solution is when the terrestrial devices directly connect to the satellite platforms. This approach can have various pros with the main cons of limited data rates and high latency. Second, the terrestrial devices connect to an aerial platform (e.g., a UAV), which play the role of the radio unit (physical layer) that subsequently connect to satellites or HAPS at low orbits (they host the rest of the upper layers and the base band unit).

This second solution tries to take advantage of the benefits given by the functional split of the softwarized baseband unit. Another benefit with this approach is that it may provide a more stable connection for the devices since they can connect via an almost stationary UAV unit.

4.3.1 NTN global coverage

One objective of WP5 is to find architectural solutions that support full global coverage. In [HEX-D52] we investigated if Inter Satellite Link (ISL) hops are needed and if different types of ISL hop schemes are needed for global coverage. The results showed that ISL hops are required for global coverage and that the performance is good for a simple scheme using “closest orbits” hops. This contribution presents a continuation and extension will continue the global coverage by simulating the coverage over the Atlantic Ocean for different Satellite settings.

4.3.1.1 Simulation setup and methodology

The settings for the satellite constellation are inspired by [PPC+21]. The constellation employs two different Low Earth Orbits (LEO)with different altitude and inclination, see Table 4-1. Orbit 1 is the “polar” option, i.e., the satellites are traversing the polar areas, and orbit 2 is more aimed towards to more populated areas. The gateways (ground stations) are placed on North America (NA) east coast and Europe west coast.

Table 4-1 Different time phases of the satellite constellation, i.e., number of total satellites in orbit.

	Altitude [km]		Inclination [degrees]		No. of planes		No. of satellites per plane		
Phase	Orbit 1	Orbit 2	Orbit 1	Orbit 2	Orbit 1	Orbit 2	Orbit 1	Orbit 2	Total no. of satellites
0	1015	1325	98.98	50.88	3	6	5	5	45
1	1015	1325	98.98	50.88	6	20	13	11	298
1.5	1015	1325	98.98	50.88	12	20	13	22	596
2	1015	1325	98.98	50.88	27	40	13	33	1671

The simulator computes the position in terms of latitude, longitude, altitude of each satellite in the constellation around the globe over time. The users are then deployed in the region of interest and uniformly distributed over the area of interest. The number of users is around 300 users placed over the Atlantic Ocean (and some devices are placed on the coast areas too), see Figure 4-8. Once the gateways are deployed, the simulator calculates the position of each satellite with respect to each ground node in terms of elevation angle, distance, and azimuth. Table 4-2 shows the involved nodes and links as well as the most important parameters used in the simulation, based on [38.821]. The UL service link is assuming a handheld device with no extra antenna gain but using 2W as maximum transit power. Each satellite can support up to 32 concurrent steerable beams. Further, each satellite uses a frequency reuse of 7 spread out over the beams.

Table 4-2 Simulation parameters, see also [38.821].

	UL Service link (UE to Sat.)	DL service link (Sat. to UE)	ISL	Feeder link
Carrier frequency	S-band (2 GHz)	S-band (2 GHz)	KA-band (20 GHz)	KA-band (20 GHz)
Bandwidth	4 MHz	30 MHz	400 MHz	400 MHz
Tx power	33 dBm	N/A	N/A	N/A
Number of beams	32	32	1	1
EIRP	N/A	40 dBW/MHz	40 dBW/MHz	40 dBW/MHz
Antenna gain	0 dB	30 dBi	30 dBi	30 dBi

Shadow fading	0 dB	0 dB	0 dB	0 dB
Misalignment loss	0 dB	0 dB	0 dB	0 dB
Link delay	5 ms	5 ms	5 ms per hop	5 ms
Min elevation angle	30°	30°	N/A	30°
Reuse factor	7	7	N/A	N/A

During the simulations, all possible connections between the UEs, the satellites and the gateways are calculated. Note that satellites in the two different satellite orbit options from Table 4-1 (called orbit1 and orbit2) can also create connections between them. Thereafter the following steps are done:

- Calculate pathloss for all connections.
- For the service connection, calculate the SINR per cell area assuming no interference (see [38.821] eq. 6.1.3.1-2).
- Select the optimal connections based on distance (ISL and feeder links) and SINR (service links).
- Calculate the number of ISL hops and the ISL delay for each UE to ground station connection.
- Calculate the number of UEs per satellite and beam
- Calculate the available spectrum per UE based on a certain beam spectrum reuse
- Calculate the maximum throughput based on the DL and UL SNR and the available spectrum using the Shannon's channel capacity equation [Sha48].
- Apply a congestion avoidance TCP model [MSM97] based on the delay and the maximum throughput to achieve a more realistic cell throughput

Note that we assume that the ISL and feeder links do not limit the throughput here. Further on, there is no interference between each beam and the beam is always pointing at the device.

4.3.1.2 Simulation results

The simulations were performed for a varying number of total satellites in orbit according to Table 4-1. Figure 4-8 shows the simulated area for phase 1.5. Solid lines show connection for LEO orbit 1 (magenta) and orbit 2 (black). Dotted lines are the satellite connection to the ground station, located in the coast areas. Dashed lines are the ISL hops.

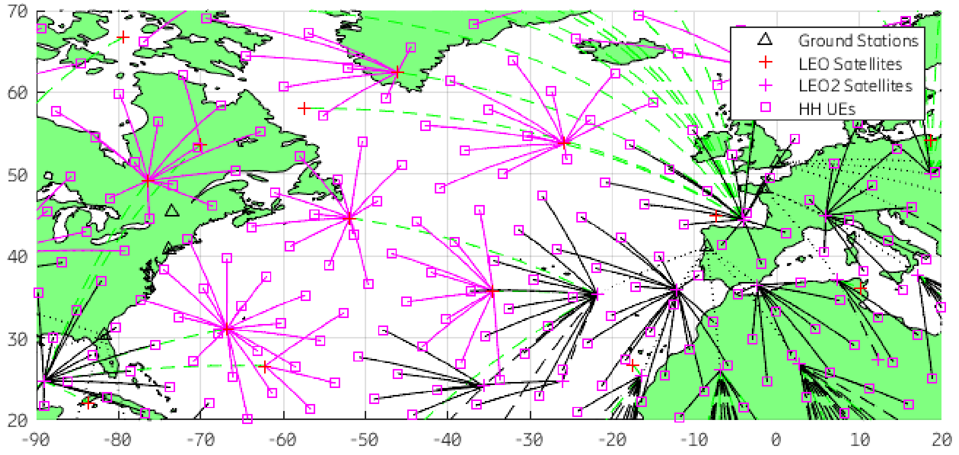


Figure 4-8 The area simulated is the area between north America east coast and Europe / north Africa west coast.

Figure 4-9 (left) shows the coverage, i.e., the number of connected UEs as a function of the total number of satellites. As can be seen when the number of satellites reach 600 in total (at phase 1.5 from Table 4-1) almost all UEs have a connection, both in UL and DL. Figure 4-9 (right) shows the cumulative distribution function (CDF) of the total delay for the connected users for the case when the number of Satellites reach 596. The total delay includes the inter-satellite link hops and the service and feeder link delays, but not the remaining delays in the terrestrial network after the ground station.

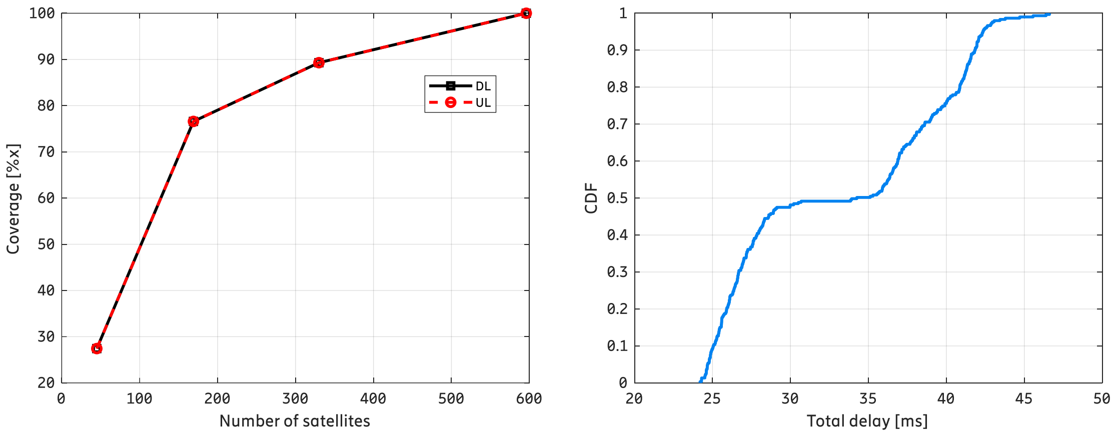


Figure 4-9 Separate DL and UL coverage in percentage for different number of Satellites.

With more available satellites per UE there is also an increase in available spectrum per UE, which increases the median TCP throughput, see Figure 4-10 (left) and the number of UEs with more than 1 Mbps in TCP throughput Figure 4-10 (right). For 600, satellites around 95% (slightly lower in UL) of the users experience more than 1 Mbps. Figure 4-10 (left) also shows that the DL throughput is limited by the delay and the high internet loss of 0.1% we assumed [MSM97]. Note that the results depend to a large extent on the simulation parameters used, such as the antenna gain, transmit power, bandwidths, etc.

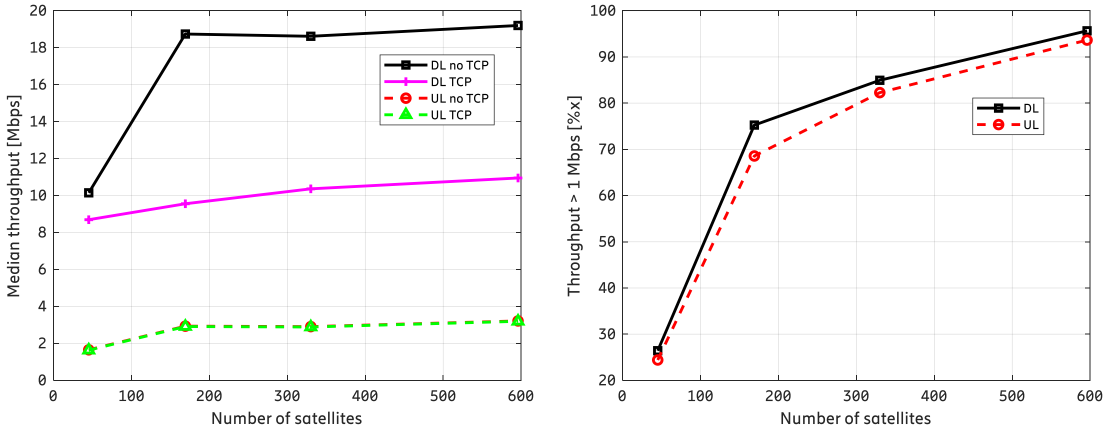


Figure 4-10 The median non-TCP (upper bound) and TCP throughput (left) and UEs with more than 1 Mbps TCP throughput (right).

4.3.2 NTN RAN split for 3D and mobility

Significant challenges arise when realising a 3D network based on UAVs and multi-layered NTN (including nanosatellites with very low orbit and HAPS). If UAVs are used as mobile base stations, they have a limited power supply. The first concern is when running Baseband Unit (BBU) functions in the UAV due to its computing resources required. Increasing the number of BBU subfunctions deployed in the UAV also increases its energy consumption, resulting to a shorter flying time since UAVs are battery powered (see the scenario in Figure 4-11). To reduce the computational complexity, different functional splits are considered and evaluated (e.g., 7-1, 7-2, 7-2x and 7-3) where RU is implemented in the UAV and the remaining RAN stack is realised in the nanosatellites (see Figure 4-11 for different splitting solutions). Another concern is that UAV should also support the transmission and reception of data in the fronthaul interface between UAV and the satellite which performs other 5G BBU functions. This means the provisioning of both downlink and uplink data rates according to the specific split option chosen. 3GPP defined the fronthaul bandwidth and latency requirements of different functional split options [38.801]. If these latency limits are not satisfied, the link between UAV and satellite experiences an increase in errors, which has been shown in Deliverable 5.2. In that deliverable, error correction codes were applied to mitigate this issue and make the system feasible. In the following text, the focus is only on the design of fronthaul link technology between the UAV and the nanosatellite, assuming the latency requirement is satisfied by the choice of a specific very low orbit. Next, it is useful to study the feasibility of the different options in terms of bandwidth and available throughput. This aspect is fundamental for the design of future 3D flexible and effective networks. Using lower layer split, the fronthaul bandwidth in the downlink direction is around 9.8Gbps for option 7-1, while 10.1Gbps for options 7-2, 7-2x and 7-3. The slightly higher bandwidth is due to the overhead added by number of layers as mentioned in the formulas in [38.801].

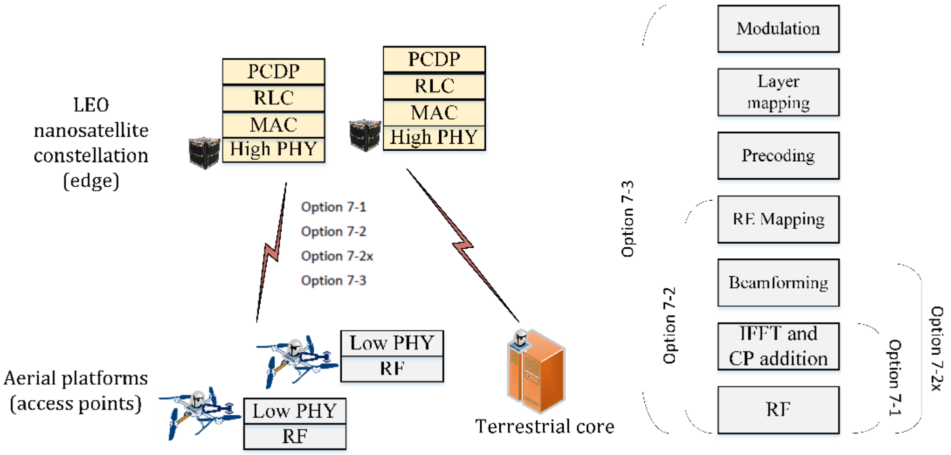


Figure 4-11 High-level representation of a possible 3D network. On the right side, number of functions implemented in the RU using different lower layer split options.

To achieve the required bandwidths between UAV and nanosatellite, operation in millimetre wave (mm-wave) is being considered for an integrated nanosatellite-5G system with some configurations and trade-offs [BCC+20]. However, there are still open challenges when using mm-wave, such as channel modelling considering the impact of Doppler effect, fading, and multipath components, which is challenging at higher frequencies. Thus, this initial research focuses on using the current satellite communication frequency bands (e.g., S- and Ka-band) and on adjusting the 5G physical layer specifications to achieve a fronthaul bandwidth that can be supported by the S- or Ka-band. Using satellite communication frequency bands, which are the S-band at 2GHz and the Ka-band at 20 GHz, multi-layered NTN have been evaluated in [WGA+21]. Based on their results, more than 0.3 Gbps in the former and 3 Gbps in the latter can be achieved when a satellite layer is assisted by HAPs operating in the lower layer. Thus, this design study focuses on the analysis of a multi-layered NTN using the Ka-band frequency.

Five different performance parameters are considered to analyse the 5G NR support for a satellite-UAV multi-layered NTN scenario, namely: fronthaul bandwidth, theoretical throughput, connection density, number of functions implemented in the UAV, and the energy consumed in receiving the fronthaul data in the downlink direction. These metrics have to be concurrently satisfied in a proper way to make the 3D network effective in order to ensure the level of coverage and capacity for flexible networks in complex and rural environments. The use of nanosatellites at very low orbits and HAPs can ensure the required low-latencies and optimal decentralisation of the BBU subfunctions. Fronthaul bandwidth is the data rate that can be transferred between the RU and the RAN in the nanosatellites, see Figure 4-12. This value depends on number of subcarriers of each OFDM symbol, number of OFDM symbols in a subframe, number of layers [3GPP18], number of antenna ports, IQ bitwidth, and MAC layer information. In parallel, connection density refers to the average number of user/devices that can be connected to the UAV. This can be achieved by dividing the theoretical cell throughput to the average user throughput.

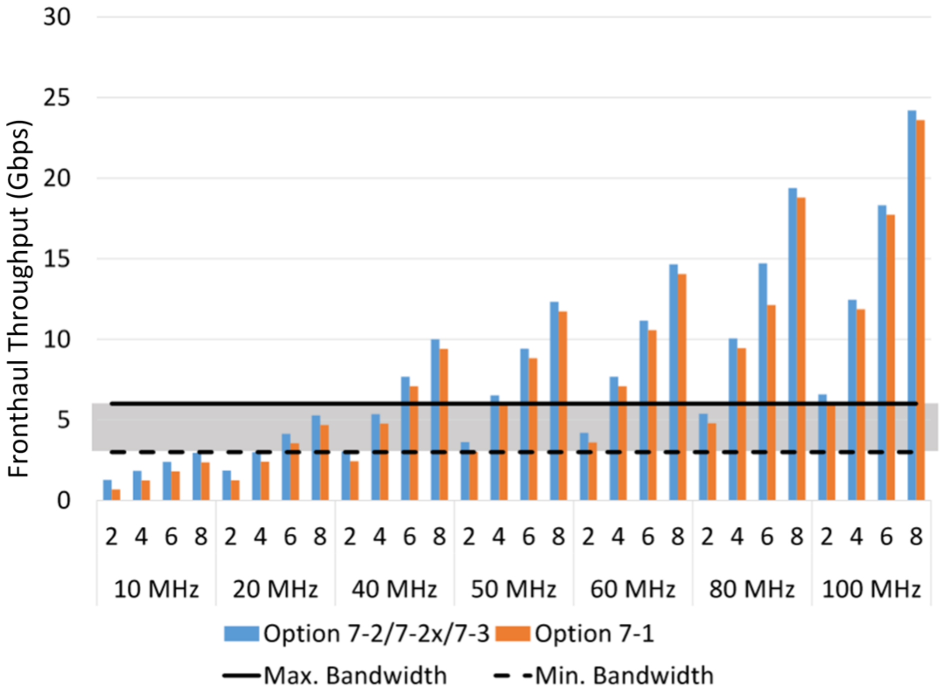


Figure 4-12 Fronthaul bandwidth with varying channel bandwidth, number of layers, and number of antenna ports.

Figure 4-12 shows the fronthaul throughput of different functional split options with varying channel bandwidth, and number of layers on Frequency Range 1 (FR1) frequency range (10 to 100 MHz). As shown in the figure, the fronthaul bandwidth increases with the increasing channel bandwidth and the number of layers. Since we are looking into the fronthaul bandwidth that can possibly be supported by the Ka-band, a minimum and maximum bandwidth limit is considered and PHY layer specifications are varied to achieve a fronthaul bandwidth within this limit. In this case, we are specifically considering the following physical layer specifications: 80 MHz with 2 layers; 40 MHz with 4 layers; and 20 MHz with 8 layers. Results achieved in Figure 4-12 only show the fronthaul bandwidth without considering the number of antenna ports used to transmit to the end devices. The number of antenna ports also affects the throughput and quality of communication between UAV and end devices. To achieve the maximum throughput, a combination of higher SNR, lower number of layers, higher number of antennas should be considered.

To test the quality of transmission, simulations were done using the 5G toolbox. Figure 4-13 shows the theoretical fronthaul throughput with varying channel bandwidth, number of layers, and number of component carriers. Based on the fronthaul bandwidth results in Figure 4-12, it is possible to achieve a maximum theoretical throughput of 0.91 − 0.93 Gbps when using 1 component carrier (CC1) while 1.81 − 1.86 Gbps with 2 component carriers (CC2). Considering a use case scenario where the average user's throughput is 10Mbps, the multi-layered NTN can support around 91 − 93 users with CC1 and 181 − 186 users with CC2.

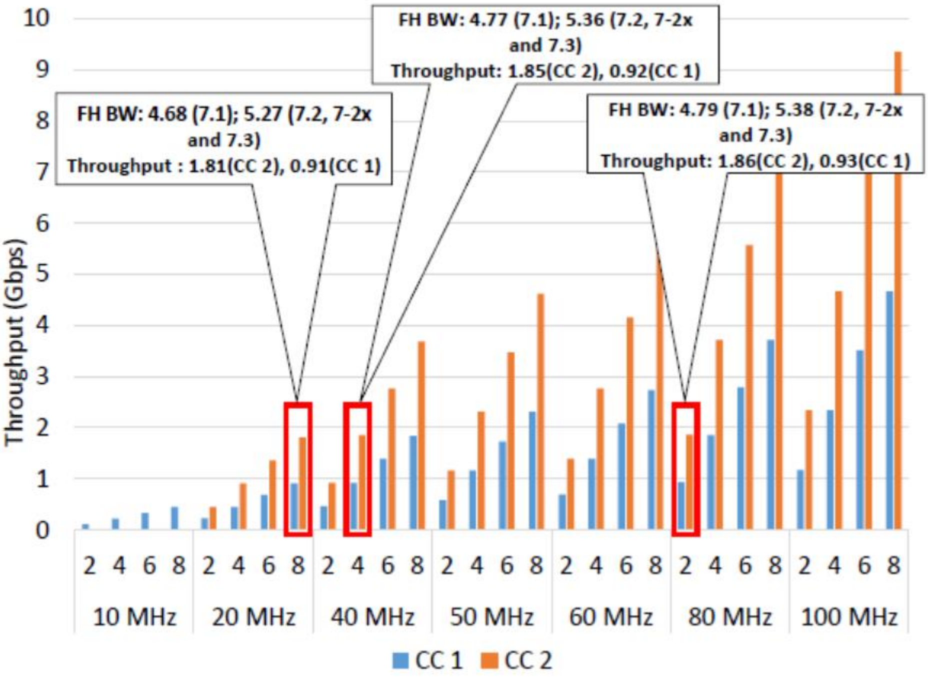


Figure 4-13 Theoretical throughput with varying channel bandwidth and number of layers.

Figure 4-14 shows the comparison of four different functional split options in terms of required fronthaul throughput (Gb/s), throughput (Gb/s), connection density, energy consumption (pJ/b), and number of functions implemented in the UAV.

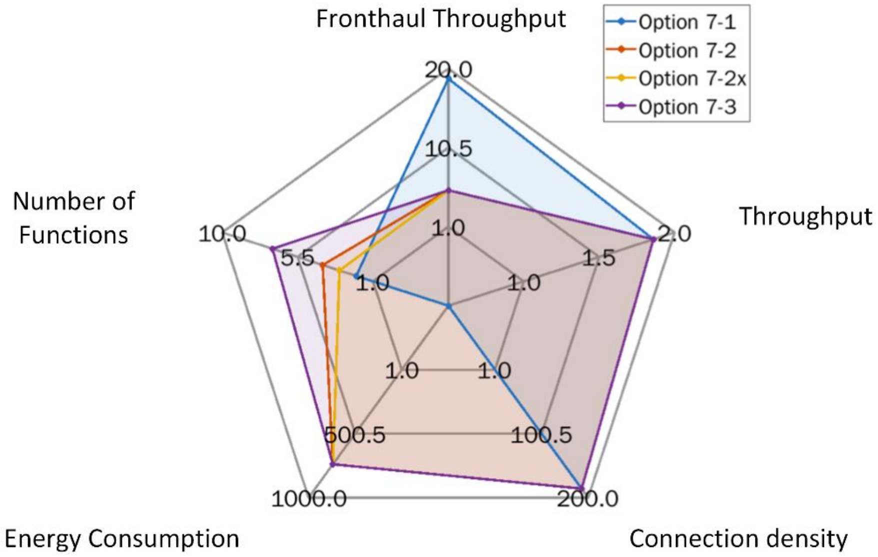


Figure 4-14 Overall comparison of different physical layer functional split options.

As shown in the figure, for all functional split options the same throughput and connection density is set as target. Given this, option 7-1 requires more fronthaul throughput (18.79 Gbps) to provide the same throughput compared to the others (5.38 Gbps). Since the fronthaul bandwidth for option 7-1 cannot be supported by the Ka-band frequency, the energy consumption of transmission is only measured for options 7-2, 7-2x, and 7-3. Next, the energy consumption per bit on the fronthaul when using these split options is around Eb = 739 pJ/bit. As for the number of functions that will be deployed

in the UAV, option 7-1 has the least functions while option 7-3 has the most functions to be deployed in the UAV.
After comparing four physical layer functional split options, it shows that using option 7 −2x is the most optimal solution on multi-layered NTN to balance the amount of computing at the UAV and the theoretical feasibility of the system. The same results are achievable with option 7-2 and 7-3 in terms of fronthaul bandwidth, throughput, connection density and energy consumption, but with a smaller number of functions in the UAV. Also, option 7-1 has a really high fronthaul bandwidth requirement that cannot be supported by the Ka-band even if it has a smaller number of functions in the UAV compared to option 7-2x.

With the limited capacity of wireless communication on NTN, it is important to analyse and determine which physical layer specifications achieve a fronthaul bandwidth that can be supported by the Ka-band spectrum. The above results show that 5G NR physical layer with 256 Quadrature Amplitude Modulation (QAM) modulation, 80 MHz channel bandwidth with two component carriers, two layers, and eight antenna ports is the ideal specification for multi-layered NTN. Using these specifications, we compared different functional split options in the physical layer in terms of fronthaul throughput, throughput, connections density, energy consumption and number of functions implemented in the UAV. Unfortunately, the fronthaul bandwidth required by option 7-1 is too high and cannot be supported by the Ka-band. The results also show that using option 7-2x for multi-layered NTN can achieve a smaller number of functions on the UAV compared to option 7-2 and 7-3.

5 Efficient network

The studies in previous deliverables have set the stage for the studies presented in this deliverable (D5.3) with a clear objective to investigate how a 6G architecture can be as efficient as possible from a set of perspectives. As a start, in [HEX-D51] a set of design principles were described. Three of them are addressed with the work on Efficient network, namely:

- Exposed interfaces are service-based, where network interfaces should be designed for cloud use (i.e., cloud-native) with care taken to design proper service separation enabling service reuse, and ease of adding new services to the network.
- Separation of concerns of network functions, which means that interaction among services, through their APIs, ensure minimal dependency with other network functions, so that network functions can be developed and replaced independently from each other.
- Network simplification in comparison to previous generations, which would be orchestrated by utilising cloud-native RAN and CN functions with fewer (well-motivated) parameters to configure and fewer external interfaces.

In the previous deliverables [HEX-D51] [HEX-D52], input from partner companies demonstrates, with examples, how the above design principles are fulfilled in a future 6G architecture. In this final deliverable, the various inputs extend the description of technical enablers, all of them supporting and enabling these principles. In a first Section 5.1, some of the principles are revisited to show how they help in the process of designing independent functions. In the next section (Section 5.2), there are details on RAN cloudification, how this can be realized for an NTN use case, e.g., how to optimize placement of NFs in terms of latency. Further, in Section 5.6, improvements to Compute as a service (CaaS) are proposed to allow delegating/offloading generic application-related workloads (besides radio signal processing ones) to networked compute nodes based on different computing platforms. Further input on CaaS proposes (Section 5.7) a new method for device mobility where the main idea is to incorporate latency requirements in the handover decision, making use of the so-called q-offset for each cell, which could be set to prioritize cells with low latency and down-prioritize cells with high latency.

Finally, methods for how to evaluate these updated enablers for a 6G architecture are presented. There are two sections, one (Section 5.4) presenting a method for how to estimate TCO of the network when new enablers are added and one section (Section 5.3) further investigating the dimensions of signalling-based KPIs affected by the enablers.

5.1 Service-based architecture

5.1.1 Introduction

In previous deliverables [HEX-D51][HEX-D52], it is assumed that the 6G architecture is service-based, however, the reasons why and benefits may need to be reiterated. Service-based architectures (SBA) have been in use in the software industry to improve the modularity of products [CHA23]. This really means that a software product can be broken down into communicating services so that the developers can theoretically mix and match services from different vendors into a single offering.

Also, in telecommunications networks, the ability to develop new functions easily and use of off-the-shelf technology, where applicable, drive changes in the network functions (NFs) themselves. With this in mind, there is a push to migrate from classic interfaces to web-based APIs. In the initial release of the 5G core network, this has been made possible and the core network then is based on what is called an SBA, centred around services that can register themselves and request/subscribe to other services. This is believed to enable a more flexible development of new services, as it becomes possible to connect to other components without introducing specific new interfaces. The system architecture following the SBA approach is specified in 3GPP technical specification 23.501 [23501].

In a future 6G architecture, the selected aspects of SBA design can be extended also to applicable parts of the RAN. This was an assumption from the beginning of Hexa-X [HEX-D51]. Once again, the assumption is that it is easier to develop new functions that manage parts of RAN functionality. With such an evolution, the distinction between core network (CN) and RAN in previous generations of cellular networks will change as it becomes possible to rearrange functionality. It makes more sense to group network functions (NFs) as “radio near” or not. “Radio near” functions or radio network functions (RNF) are responsible for a smaller area in the network, e.g., such functions cover a “single base station”, an area like D-RAN or a larger area like a Centralized RAN (C-RAN) deployment. RNFs design should strive for full function inter-working with other “radio-near” functions and other functions, at least for standardized functions. These functions need to support multi-vendor handover to other RNFs. Some advantages over the current architecture that can be anticipated are: Less duplicated functionality, improved cross-layer AI/ML with full knowledge of UEs and resources in one place.

In this SBA it would not only be RAN accessing CN functions but also the UE that communicates with individual NFs. Proposals that make this possible comprise function elasticity [HEX-D52] and in particular 6G-RAN-CN function elasticity as well as service-based interfaces (SBI) to enable signalling directly between NFs. The first change is achieved by co-locating some of the common 6G-CN NFs, i.e., NFs that are often used such as those providing mobility, with the 6G RAN-CP in the cloud environment, which allows placement of signalling procedures, such as mobility and session management, in the regional edge cloud. As a result of placing critical signalling processing together with 6G-RAN-CP in the regional edge cloud, signalling performance is improved thus reducing latency. This approach can be applied for 6G-UE associated services since the 6G-UE context handling would remain within the control of the 6G mobility management without creating new or additional dependencies. The second change, introducing SBI, enhances the possibilities for signalling directly between NFs. Today many services require information transfer from one Next Generation RAN (NG-RAN) node to another NG-RAN via the 5GC. In 5GC the information is relayed via the Access and Mobility management Function (AMF) with limited or even no processing steps by the AMF [HEX-D52]. With SBI there is no need to pass through the AMF. Note that there may be cases when proxying is useful, e.g., to help with discovery of the correct NF.

Applying the abovementioned ideas to the 6G architecture gives us the architecture seen in Figure 5-1. This architecture is in line with what is described in [HEX-D63] where a complete view of the 6G architecture is presented. The part discussed in this section corresponds to what is referred to as network layer in the complete 6G architecture. In addition to the network layer the architecture in [HEX-D63] has an infrastructure layer below and a service layer above. The layers are all connected to an API that manages exposure functions . This report does not discuss the exposure functions in detail, but they are included in Figure 5-1since some features mentioned in later sections, e.g. Sections 5.7 and 5.7.2, require support for data exposure. The infrastructure layer is omitted in Figure 5-1.

The other functions or shared network functions (SNF) are responsible for larger areas in the network (or the whole network). Therefore, relocation is not critical for these functions. The functions support reusable self-contained services, independent scaling, while striving to use main-stream solutions (e.g., for security).

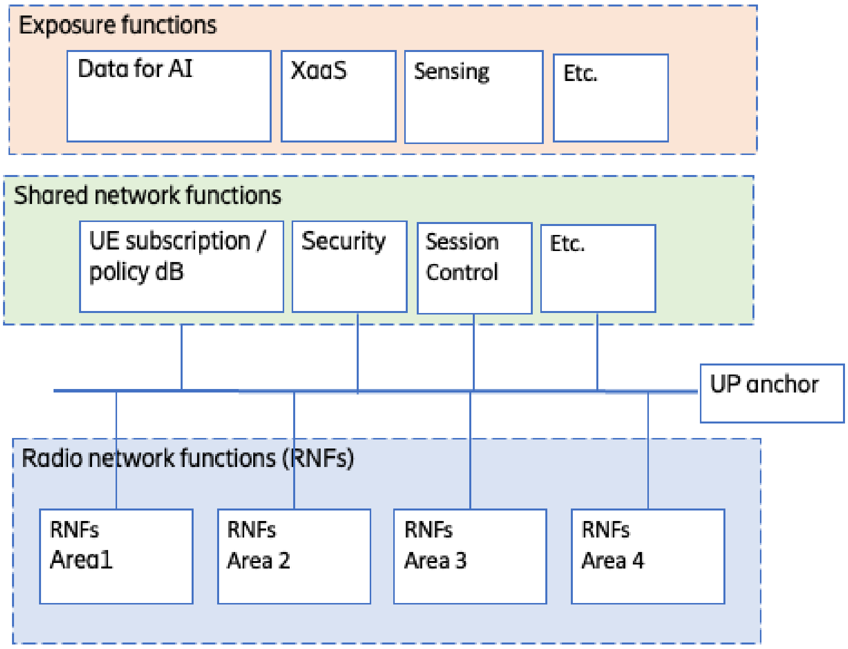


Figure 5-1 Architecture for efficient network.

In [HEX-D52] there is a list of actions for how to optimize network functions in order to meet the objectives set for this activity. First, if a procedure requires that more than two NFs need to communicate try to split the NF into separate procedures and avoid synchronism.

Further, procedures should be independent (as the ones in Figure 5-2). In this way, procedures can be updated separately. Procedures are more generic and therefore a procedure does not need to know if another procedure is changed. Also, combinations of different procedure can be used to achieve different outcomes. Finally, procedures can rely on each other, i.e., they need to be executed in a particular order, but procedures should not be nested. The gain from applying these principles does not come from having smaller functions but the overall principle.

As already mentioned, this design avoids functional proxies. For example, RNFs can talk to other RN NF and shared NFs when this is needed, as a result of how the functional architecture and service composition evolve. In other words, authorized functions can communicate without AMF.

5.1.2 Functionality example

In [HEX-D52] security in this new SBA was mentioned briefly, in a list of candidate actions to enable independent network functions. The proposal is to introduce separate security associations per service. Such a change could increase the security granularity but could also provide complexities in case of service/function relocations or general authentication/authorization. So, in this section some more details for how "independent" security associations can be introduced, with fundamentals as an authentication run, and the credentials to base the authentication upon. In the current 3GPP security architecture [23.501][33.501], a UE receives keys for different security associations without multiple authentication runs, but still cryptographically separated in a useful way.

To delimit the description, the focus is on a case where the UE communicates with a NF. To enable the communication, the UE has CP security associations with each NF that it wants to communicate with. With such a setup of NFs, design and deployment become independent. Although the assumption is a CP connection, communication could also be carried over UP. Having such an independent NF comes at a cost since all NFs need to cater for idle UEs and possible paging. So even if this results in a clean architecture, optimizations of signalling may be more complicated.

As seen in Figure 5-2, from the start the UE subscription information is defined and the network is up and running. This means, among others, that the cell is ready to accept new users. A further assumption is that at this stage in the process, different functions in the network have established secure relations to protect inter-function signalling.

The (first) authentication creates a security context for the UE in the Authenticator function. Any function that needs security information and/or keys request that from the Authenticator (and should subscribe to future updates). Communication between the UE and other functions/services is passed via the RNF. Within the RNF there is functionality that only allows communication by authenticated and registered users.

Connection establishment requires the UE to register to the network and be authorized and authenticated. In this process, registration and authentication are kept as separate procedures so that the latter procedure, with focus on security related aspects, can be reused. To begin, the UE performs mutual authentication with the authenticator in the network thereby generating a security context. The UE connects directly to the authenticator (via the RNF) for that UE. The authenticator accepts the UE and the temporary UE ID or token.

Registration is initiated by the UE, using the same credentials, with a message to the RNF. The RNF requests security information from the authenticator, which after deriving connections keys, acknowledges. The RNF then compares registration information with UE subscription data and with the information creates the UE context (including subscription information) and also subscribes to any changes of the subscription. After this the registration is complete.

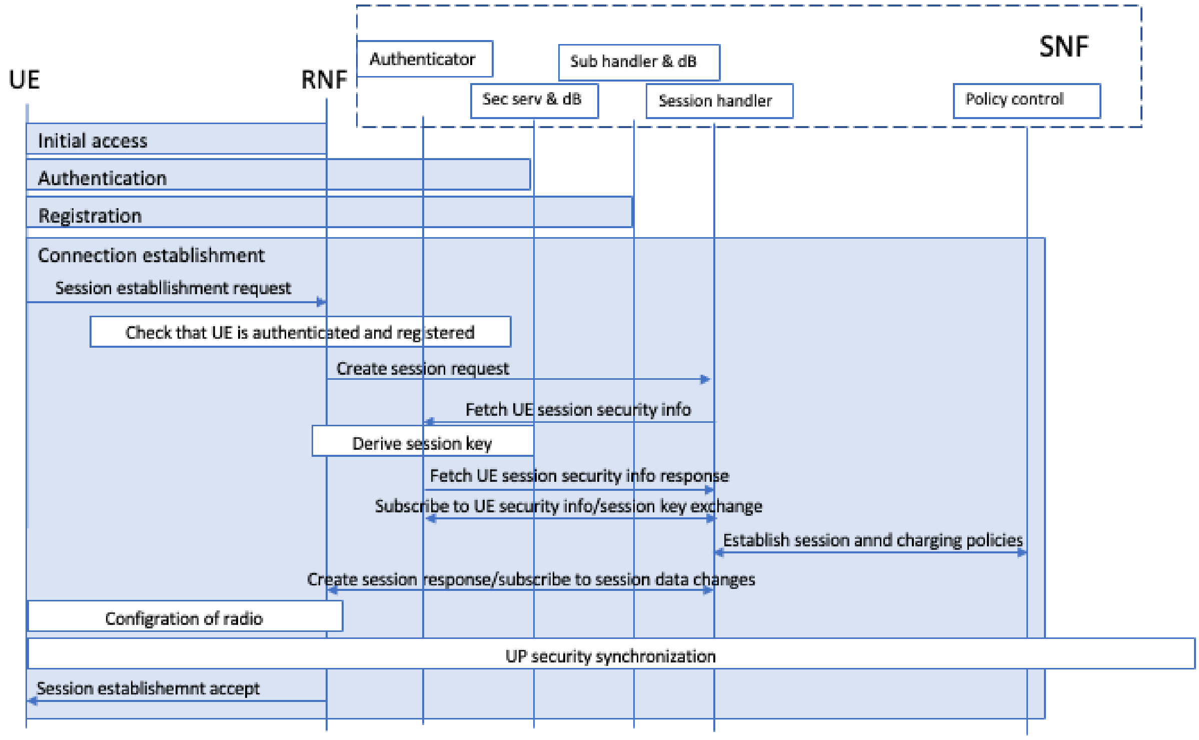

SNF
UE
RNF
Authenticator
Sub handler & dB
Sec serv & dB
Session handler
Policy control
Initial access
Authentication
Registration
Connection establishment
Session establlishment request
Check that UE is authenticated and registered
Create session request
Fetch UE session security info
Derive session key
Fetch UE session security info response
Subscribe to UE security info/session key exchange
Establish session annd charging policies
Create session response/subscribe to session data changes
Configration of radio
UP security synchronization
Session establishemnt accept

Figure 5-2 Connection establishment in the SBA architecture outlined in the previous section.

Once the procedures described above, including those in Figure 5-2, are completed, the UE has completed mutual authentication and is registered in the network. Therefore, the network is able to charge the UE. Also, the UE can trust that it is connected to a "safe" network. The UE has connectivity for the wanted service and the network functions have all the information about the UE that is needed to provide the service.

5.2 RAN cloudification for supporting edge computing in satellite backhaul and fronthaul scenarios

In the following section, it is outlined how existing Edge Computing in satellite backhaul and fronthaul scenarios can benefit from the architectural enablers defined in Hexa-X Deliverable 5.2 [HEX-D52].

3GPP TR 23.737 [23.737] specifies a scenario for the usage of a satellite backhaul between the core and terrestrial access network providing a transport for the N2/N3 reference points. The satellite system transparently carries the communication payload of the 3GPP reference points.

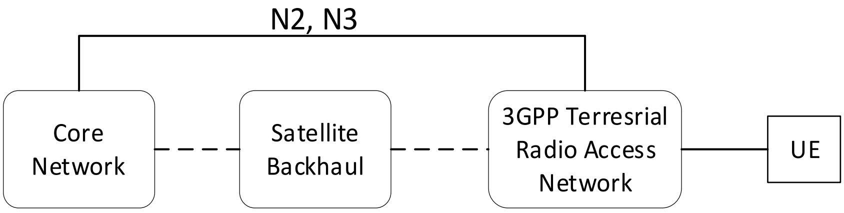


Figure 5-3 5G System with a satellite backhaul.

In this scenario edge computing can be supported by collocating a UPF with the terrestrial access network allowing for efficient service delivery through the reduced end-to-end latency and load on the transport network (e.g., reducing traffic on the satellite backhaul). In [23.737] an architecture is studied where a network function at the edge (edge NF) is capable of storing content files (e.g., video segments in HTTP-based video streaming applications) provided by a Content Distribution Network (CDN) server through a satellite link and making them available at the edge cache. This architecture is illustrated in Figure 5-4.

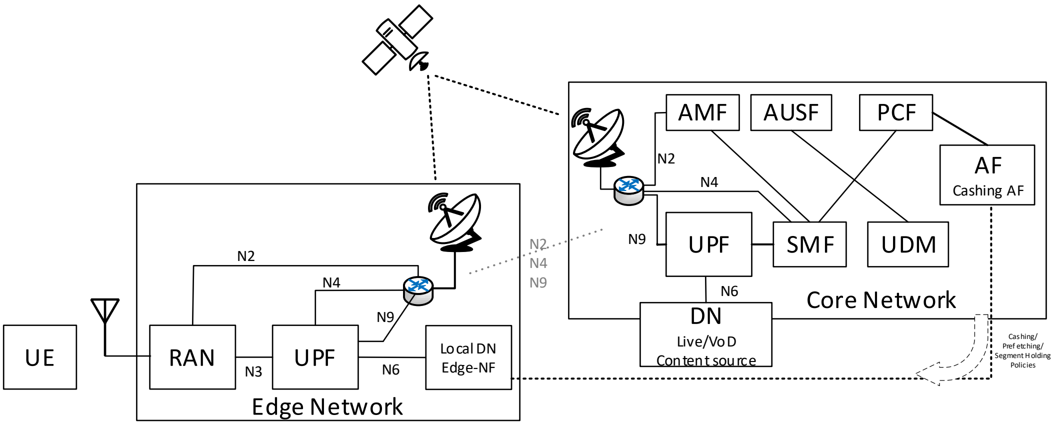


Figure 5-4 Architecture overview with edge-based content storage and request handling from [23.737].

In Figure 5-4 the UPF in the edge network exposes the N4 and N9 interfaces over the satellite link (feeder link plus service link) towards the core network. For a specific PDU session, a single SMF in the core network controls the UPFs in the edge network and the core network.

However, besides the achieved optimization in efficient service delivery, this solution requires forwarding a significant amount of control messages (e.g., N1/NAS, N2/ NG Application Protocol

(NGAP), N4/ Packet Forwarding Control Protocol (PFCP)) over the satellite link (feeder link plus service link) which has the following issues.

- Satellite systems feature much larger propagation delays than terrestrial systems. According to 3GPP TR 38.811 [38.811] the one-way delay between the satellite-gateway (edge site) and satellite-gateway (CN site) may reach up to 274ms for GSO (Geostationary Synchronous Orbit) systems and is greater than 15.5ms for NGSO (Non-Geo stationary Synchronous Orbit) systems.
- In [CCM21], the Local Offload Split Model measurements performed in a testbed using a Geostationary Equatorial Orbit (GEO) satellite backhaul indicate that e.g., the Registration and PDU session establishment procedures observe a significant delay compared to an Ethernet backhaul.

Table 5-1 Satellite backhaul.

Procedure	Register	PDU Session Establishment	PFCP Messages	Deregister	Control: Backhaul RTT
With satellite backhaul (ms)	1369	1415	630	1320	582
With Ethernet backhaul (ms)	16.7	42.8	2.6	14	2

A second scenario is a satellite fronthaul scenario as illustrated in Figure 5-5 where a UPF is deployed on a GEO satellite with gNB on board as studied in 3GPP TR 23.700-27 [23.700-27]:

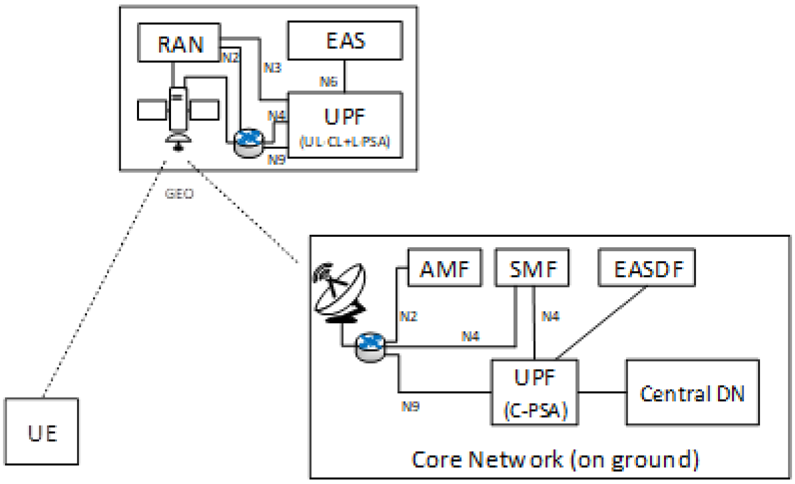


Figure 5-5 Satellite Edge Computing via UPF on-board.

Like in the satellite backhaul scenario in Figure 5-4, this solution requires the exposure of the N2 and N4 interfaces over a satellite link (see Figure 5-5) leading to higher control plane latencies compared to a local Ethernet connection between RAN, AMF, SMF, and UPF. Like for the satellite fronthaul scenario, the excess latency needs to be addressed in 6G.

5.2.1 Latency optimizations for Edge computing in satellite backhaul scenarios

In the following subsection latency- aware NF function placement is proposed for Edge Computing in satellite backhaul scenarios by leveraging architectural building blocks proposed in Hexa-X delivery 5.2 [HEX-D52] for: *(i)* dynamic function placement (DFP), *(ii)* the replacement of the N2 interface with a service-based interface (SBI), and *(iii)* distributed NAS enabling per NF service signalling.

Figure 5-6 illustrates an architecture with an SMF included in the edge network:

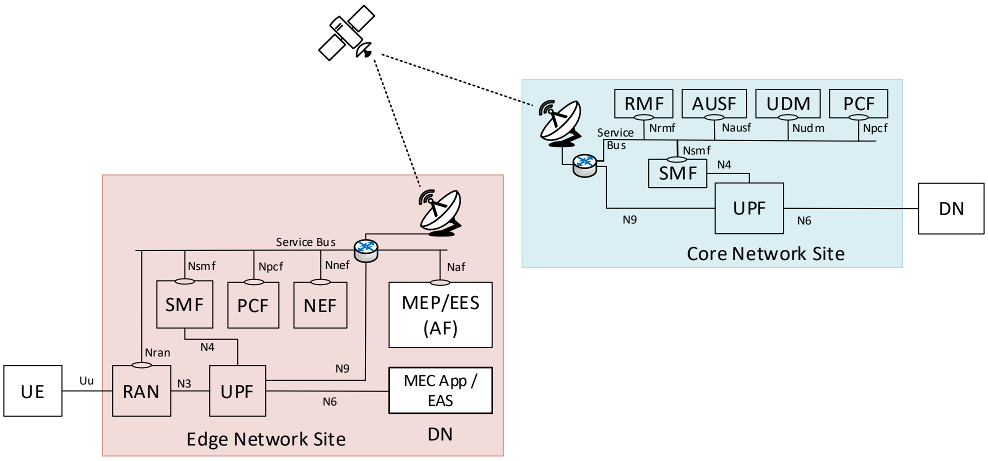


Figure 5-6 Edge Computing with Satellite backhaul and latency-aware NF function placement.

In the architecture illustrated in Figure 5-6 the terrestrial RAN is connected to the service bus in the edge network supporting direct communication between RAN and core NFs (e.g., SMF) without relaying via the AMF. The SMF in the edge network terminates the N4 interface towards the UPF in the edge network to avoid the need to send PFCP messages over the satellite link. The SMF in the core network terminates the N4 interface towards the UPF in the core network. While in a distributed NAS as specified in [HEX-D52] (which is for further study) the AMF is no longer involved in forwarding messages between RAN and CN, its remaining functionality for registration and mobility management is moved in the newly introduced Registration and Mobility Management network function (RMF). For session management the PCF in the edge network directly interacts with the SMF in the edge network. The PCF retrieves the policies from the UDM in the core network, if not locally available from MEC Platform (MEP) / EES (Edge Enabler Server) can be allowed to access services exposed by the NEF. Note also that both Edge application server (EAS) and EES (and their counterparts in ETSI MEC architecture [MEC003], i.e., MEC Application and MEC platform) may reside outside the PLMN domain. Practically, in some cases, they can be physically placed in the same edge PoP (point-of-presence) but running over different Network Functions Virtualization Infrastructure (NFVI￼￼Figure 5-6￼Figure 5-7specifies the registration and PDU session procedure supporting latency-aware NF function placement specifies the registration and PDU session procedure supporting latency-aware NF function placement.

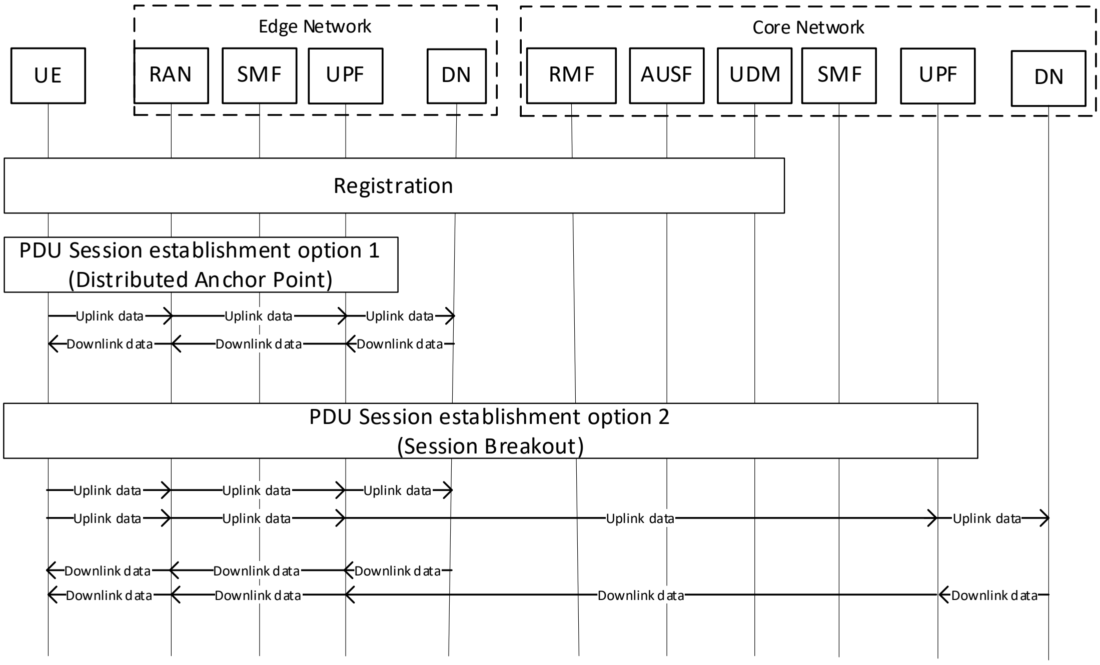


Figure 5-7 Registration and PDU session establishment procedure (simplified version without PCF and UDM interactions).

In Figure 5-7 the registration procedure is anchored in the RMF. The new functional split between RAN and RMF is optimized for latency by reducing the amount of control messages sent over the satellite link.

The PDU session establishment procedure supports two options. Option 1 uses a distributed anchor point with the UPF PSA and DN in the edge network (see 3GPP TS 23.501 [23.501]). In this option the PDU session is controlled by the SMF of the edge network. The concept of a distributed NAS as specified in [HEX-D52] (which is for further study) supports direct communication between RAN and SMF may allow to reduce the number of control messages sent over the satellite link. Option 2 uses session breakout with multiple UPF PSAs in the edge network and core network (see [23.501]). In this option the PDU session is controlled by the SMF of the edge network and the SMF of the core network. Both the SMFs act as a single logical SMF. The functional split between the edge network SMF and the central core network SMF is optimized for latency by reducing the amount of control messages sent over the satellite link.

5.2.2 Latency optimizations for Edge computing in satellite fronthaul scenarios

This scenario is based on 3GPP TR 23.700-27 [23.700-27] where UPF is deployed on a GEO satellite with a gNB on-board. The proposed solution in [23.700-27] also requires exposing the N2 and N4 interface over a satellite link (feeder link) which has issues as outlined above. To mitigate those issues latency-aware NF function placement for Edge Computing in satellite fronthaul scenarios is proposed by leveraging architectural building blocks introduced in [HEX-D52] for (i) dynamic function placement (DFP), (ii) the replacement of the N2 interface with a service-based interface (SBI), and (iii) distributed NAS enabling per NF service signalling.

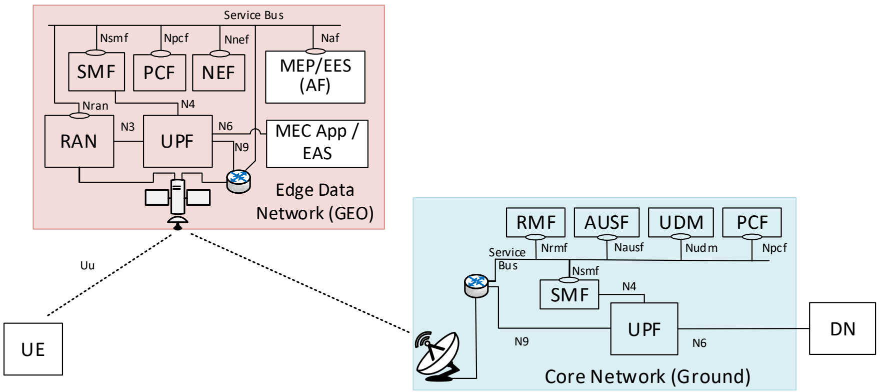


Figure 5-8 Satellite Edge Computing with latency-aware NF function placement.

In Figure 5-8 the satellite RAN is connected to the service bus of the on-board edge network supporting direct communication between RAN and selected core NFs (e.g., SMF). Like in Figure 5-6 the SMF in the edge network terminates the N4 interface towards the UPF in the edge network to avoid the need to send PFCP messages over the satellite link. The SMF in the core network terminates the N4 interface towards the UPF in the core network. And like in Figure 5-6 the AMF is no longer involved in forwarding messages between RAN and CN and its remaining functionality for registration and mobility management is moved into the new Registration and Mobility Management network function (RMF). Also, here the edge services consumptions from MEC Platform (MEP) / EES (Edge Enabler Server) can be allowed as usual by means of the service exposure via NEF. Note: as already clarified for the satellite backhaul scenario, also here both EAS and EES (and their counterparts in ETSI MEC architecture [MEC003], i.e., MEC Application and MEC platform) may reside outside the PLMN domain. Practically, in some cases, they can be physically placed in the same edge PoP (point-of-presence) but running over different NFVI infrastructures. Finally, also the MEC orchestrator (not shown in the figure for simplicity) as a 5G AF interacts with NEF and with other relevant NFs with regards to overall Monitoring, Provisioning, Policy and Charging capabilities. The MEC orchestrator can be typically deployed in a more centralized location, i.e., on the right-side of the Figure 5-8. For the registration and PDU session procedure, please refer to Figure 5-7.

5.3 Efficient signalling performance in 6G architecture

In the first attempt to measure how much more efficient the proposed architecture is a few examples of signalling were analysed with regard to latency [D52-HEX]. The examples showed that the latency for some procedures could be reduced. However, an optimization only of latency will not provide the architecture that 6G needs, since the requirements for different NFs will differ. Hence, this section provides an investigation of other KPIs and their pros and cons.

Assuming that different NFs have different requirements, e.g., some are “radio near” and (latency) critical while some NFs can process data under relaxed time constraints, it seems likely that to show that the proposed changes to the network are really efficient we need KPIs with more than one dimension, e.g., a spider diagram.

The following is a demonstration of how a KPI map can be designed. The axes provide a set of measures; however, they may change during the process of optimizing functionality of NFs for the 6G architecture.

- Latency to execute a procedure is still an important KPI. To be a bit more precise, latency is the time to complete a defined procedure.
- The number of functional dependencies indicates how many times a certain entity depends on another entity to complete a task. This measure impacts latency and also, e.g., error handling as a result from failure to signal between NFs. The KPI is discussed in [HEX-D52], as "good separation of concerns". By separating the concerns of a function, the function can be made smaller or larger. Allowing the NF to become larger may be a reasonable assumption. This should be understood as if the NF becomes more capable and thus less external signalling will be needed. However, in some cases, having too good separation may affect the context handling poorly, i.e., rather than reducing the number of signalled messages they need to be increased.
- The number of functional processing occasions or points indicates how many times a functional entity has to process messages received from another entity. Once again latency is affected by the individual processing times.
- The number of failure points indicates how many times a functional entity would require a re-start of a procedure resulting from a failure to send/receive a message. Note that the number of failure points is not only an indication of the number of dependencies between NFs but also an indication of the likelihood that a process is interrupted.

In Figure 5-9 the principle of the proposed KPI map is demonstrated for different deployments. The different deployments can for example be how the NFs are defined, distributed RAN and CN and centralized solutions.

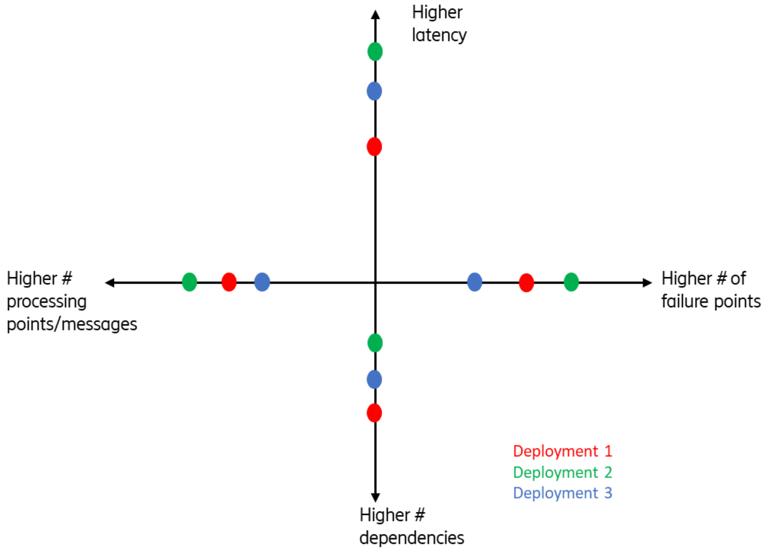


Figure 5-9 Principle for evaluating assumptions on NFs and network. Note that the points in the figure only depict an illustrative example.

Figure 5-10 depicts a baseline procedure, namely a 5G handover for a split-RAN deployment. The y-axis counts the number of messages and processing points, discussed above. In the following figures we show how values for one of the axes are determined for three different deployment scenarios. For a complete plot, evaluations are needed for the remaining parameters as well.

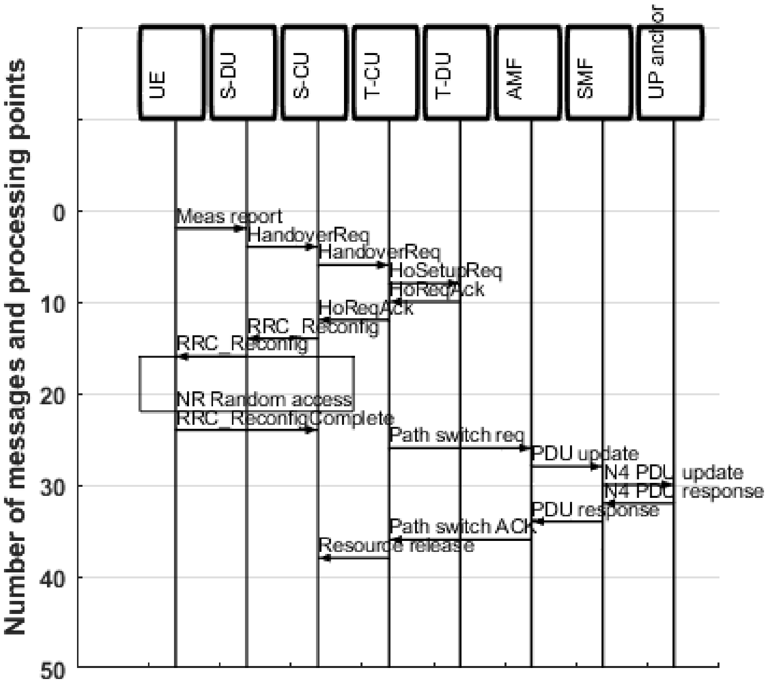


Figure 5-10 5G handover for a split RAN deployment and path switch.

The next figure, Figure 5-11, shows a handover with some attempt to optimize parts of the network.

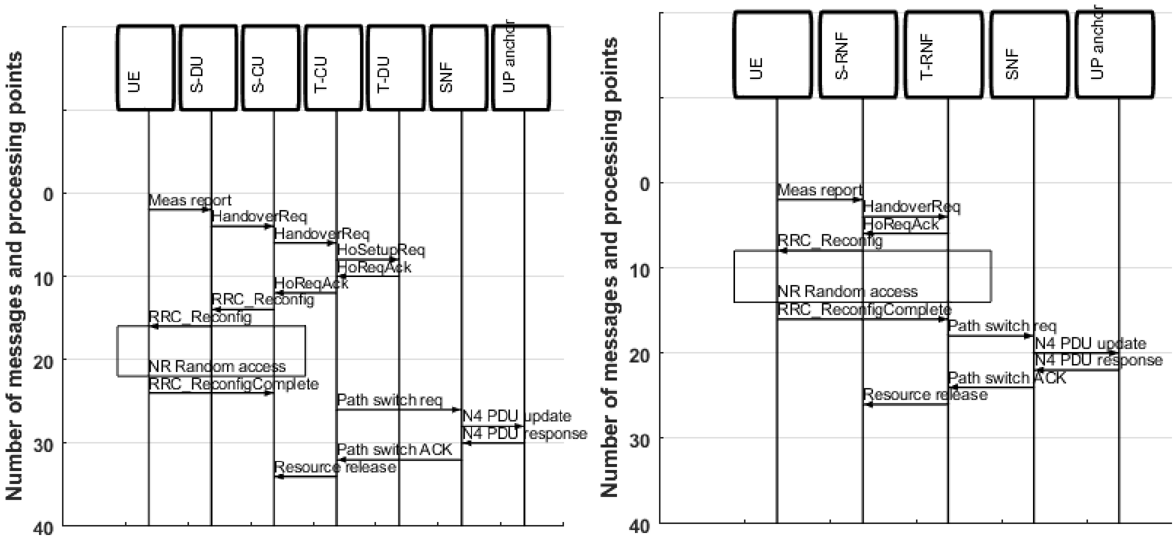


Figure 5-11 5G handover for a split RAN deployment and path switch using "Shared Network function" (left) and 6G handover for a centralized RAN deployment and path switch using "Shared Network function" (right).

The box labelled SNF will in reality comprise a set of functions, the shared network functions described in Section 5.1, and in this example includes the functions of AMF and SMF needed for the HO. This example shows fewer messages than the baseline however this depends on how the SNFs are designed. Finally, in Figure 5-11 right, 6G SBA architecture is assumed with the RNF comprising the "radio near" functions in a centralized deployment. Naturally, this deployment will have fewer interfaces and fewer processing points. With these procedures it is possible to grade NF designs in different types of deployments during various stages of the development process.

5.4 TCO aspects

In [HEX-D51] some initial Total Cost of Ownership (TCO) considerations for 6G have been drafted, that is, Capital Expenditures (CapEx) and Operational Expenditures (OpEx) breakdown for a typical mobile network as well as their major cost components. While in [HEX-D52] considering that one of the Hexa-X project's objectives relates to the TCO reduction by at least 30% for 6G networks, a methodology for achieving such objective has been developed which takes the 5G NR Standalone (SA) as the baseline architecture for cost dynamics' comparison. Moreover, the network's cost structure in terms of RAN infrastructure, energy consumption, backhaul, CN infrastructure, and other network costs (people, network management and maintenance, etc.) as well as the "weight" of each cost item have been defined based on the analysis performed by GSMA in [GSM19]. According to [GSM19], RAN subcomponents include passive infrastructure (towers, cabinets), and active infrastructure (radio antennas, as well as baseband processing, and related power and cooling, equipment).

In this deliverable a qualitative TCO analysis for some exemplary Hexa-X use cases will be reported: the considered use cases have been selected in order to match the characteristics of the GSMA's deployment scenarios considered for 5G in [GSM19] – see below Table 5-2 – and then qualitatively evaluating the impact of each cost item – RAN infrastructure, energy consumption, backhaul, core infrastructure, other costs – when deploying the most significant and use case specific 6G technical enablers among the ones identified by the Hexa-X project technical Work Packages (WPs).

Table 5-2 Three deployment strategies as considered by GSMA for 5G [GSM19].

	Strategy #1: ***rapid, full-scale 5G deployment***	**Strategy #2:** ***enterprise-focused 5G deployment***	**Strategy #3:** ***capacity-backfilling 5G deployment***
5G strategy	Target new 5G use cases in both Consumer and Enterprise segments	Existing and selected new Enterprise use cases	Existing use cases, capacity backfilling and eMBB services
5G network rollout	Rapid 5G rollout covering 80% of the population with high-capacity 5G network by 2025	Fast-paced deployment covering 65% of the population with high capacity 5G network in enterprise hubs by 2025	Measured 5G deployment covering 50% of the population with additional 5G capacity by 2025
2018-2025 data traffic CAGR	40% CAGR	30% CAGR	20% CAGR
2025 vs 2028 traffic multiple	10x	6x	3x

For the use case selection, a proper analysis of the Hexa-X use cases in [HEX-D12] – and corresponding details and refinements in [HEX-D13] – has been performed in order to map some of the most representative 6G use cases to the deployment scenarios as in Table 5-2. This is needed in order to consider the TCO breakdowns for the above reported 5G deployment strategies in [GSM19] as the baseline evaluations for successive 6G TCO considerations. Results of such mapping activity is reported in Table 5-3. Note that the deployment scenario termed as "*Strategy #3: capacity-backfilling 5G deployment*" has not been considered in the 6G use cases mapping activity since it is a more cautious (i.e., measured) deployment strategy than the others: it does not fit with the challenging requirements of 6G use cases and it addresses existing 5G use cases and Enhanced Mobile Broadband (eMBB) services (as per GSMA's description).

Among the 6G use cases listed in Table 5-3 only a single use case per deployment strategy has been chosen: the "*Fully merged cyber-physical worlds*" mapped to the "*Strategy #1: rapid, full-scale 5G deployment*" and the "*Interacting & cooperative mobile robots & flexible manufacturing*" mapped to the "*Strategy #2: enterprise-focused 5G deployment*" – these use cases are highlighted in ***green*** in Table 5-3. The main reason for selecting these two use cases is that they are in line with the Hexa-X vision of 6G being the technology that connects three worlds and revolves around their interactions: a human world of human senses, bodies, intelligence, and values; a digital world of information, communication, and computing; and a physical world of objects and organisms.

Table 5-3 Mapping of exemplary Hexa-X use cases to the GSMA's 5G deployment strategies.

		Hexa-X use cases
GSMA 5G deployment scenarios	**Strategy #1: rapid, full-scale 5G deployment**	• eHealth for all • ***Fully merged cyber-physical worlds*** • Immersive smart cities & integrated micro-networks for smart cities
	Strategy #2: enterprise-focused 5G deployment	• Digital Twins for manufacturing • ***Interacting & cooperative mobile robots & flexible manufacturing***

The following two subsections will detail qualitative TCO evaluations for the selected Hexa-X use cases. Such evaluations have been derived based on the impact that a certain technical enabler needed for the actual implementation of the use case has on the cost items considered in the TCO analysis, i.e., RAN infrastructure (towers, cabinets, radio antennas, baseband processing, related power, and cooling equipment), energy consumption, backhaul, CN infrastructure, and other network costs (people, network management and maintenance, etc.). It should be noted that, according to [GSM19], the above-reported TCO cost items are listed based on their "weight" in the overall TCO: this means that, e.g., RAN infrastructure has a higher impact on TCO with respect to energy consumption and so on. The technical enablers considered in the evaluations are the ones identified by the Hexa-X technical WPs and grouped as follows:

- enablers for the ***Intelligent network***: UE and Network Programmability, dynamic function placement/network meshes, analytics, network automation, AI-as-a-Service (AIaaS), AI-driven orchestration,
- enablers for the ***Flexible network***: integration of sub-networks, flexible topologies (D2D, Mesh Networking), campus, Edge-to-Network-Cloud integration enablers,
- enablers for the ***Efficient network***: efficient RAN/CN signalling, function refactoring, Compute-as-a-Service (CaaS),
- enablers for the ***6G RAN***: high data rate radio links, distributed large MIMO, localization, and sensing,
- enablers for the ***Service Management***: Continuum management and orchestration, AI-driven orchestration.

An example of quantitative TCO analysis will be provided in the Hexa-X deliverable D1.4 as part of fulfilment of quantified targets related to the Hexa-X Objective 1 "*Foundations for an end-to-end system towards 6G*", in particular the "*Total Cost of Ownership (TCO) reduction by (>30%)*" target. It should be noted that deriving such kind of quantitative TCO evaluations is a complex and challenging task since not only 6G is a completely new system whose architecture is still under definition but also 5G NR SA, i.e., the baseline architecture for the TCO study, is still in a deployment phase in most of the countries worldwide [EBS+22], hence its costs – especially OpEx – cannot be derived based on the actual experience of having such kind of network in place and properly operating. Moreover, the 5G system is expected to be further improved in the coming years with optimized and innovative features introduced in the 3GPP releases beyond Rel-15 – e.g., with Rel-18 and subsequent releases, 3GPP is going to standardize the so-called 5G-Advanced [3GPP-22] – which could have an impact of the TCO of the baseline architecture that has been considered in this activity.

5.4.1 Qualitative TCO evaluation for the “Fully merged cyber-physical worlds”

One of the use cases chosen for the TCO qualitive evaluations is the "Fully merged cyber-physical worlds". This use case is part of the "Telepresence" use case family defined in [HEX-D12] and elaborated further in [HEX-D13]. All the use cases belong to this use case family focus on the possibility of being able to be present and interacting anytime anywhere, using all human senses. The Fully merged cyber-physical worlds use case, in particular, will benefit from Mixed Reality (MR) [HEX-D12] - a term for advanced augmented reality bringing immersive experiences with more than visuals and audio, adapted to the environment you are in, to make the holographic presence at work and social scenarios a reality and norm. Via holographic telepresence it will be possible to appear as though one is in a certain location while really being in a different location – for example, appearing to be in the office while actually being in the car. A wide range of day-to-day examples exists that can benefit from this use case. As an example, “non-material fashion” and “Augmented shopping” in which digital objects and overlays are being used to create a personal expression that can be viewed and or otherwise sensed in MR by others. The user can choose who can see their digital outfits, and swap, sell and purchase digital items as well as create his/her own outfits.

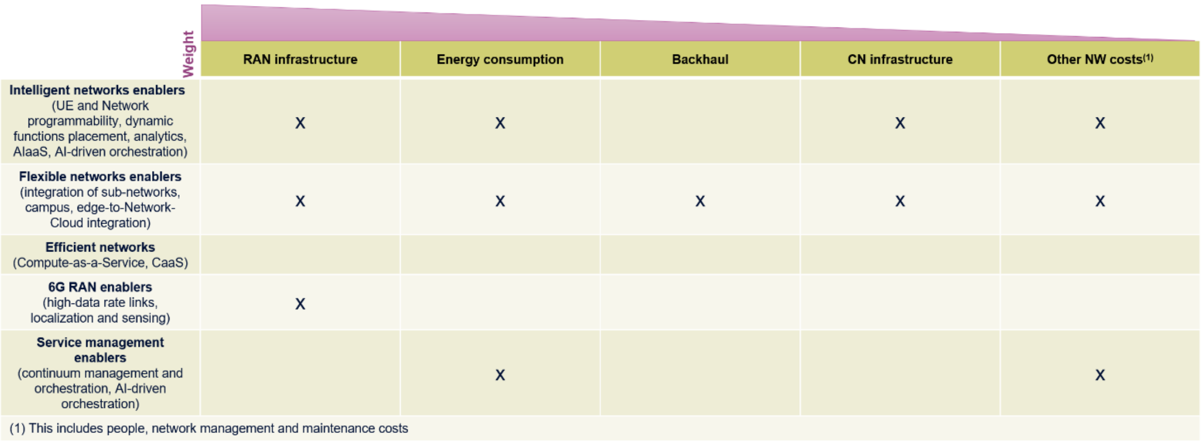
More and more people will have multiple wearables that seamlessly interact with each other, through natural, intuitive interfaces. The devices and applications will be fully context-aware, and the network will become increasingly sophisticated at predicting our needs. Considering the near reality expectation, all the relative research challenges need to be met. Extreme experience is needed to be able to meet the needed data rates. Low latency with high data rates and acceptable reliability is needed to avoid an incomplete experience or even nausea. Fully merged cyber-physical worlds is one of those use cases that highly depends on the device as well as the network to support the high requirements of the service. Although AR/VR technology has existed for a couple of years, adaptation at scale needs 5G technologies such as edge computing, ultra-low latency, and high bandwidth. Since the expectation from MR is that the user can see and interact with both digital and physical elements stimulatingly, the requirements on the network are strict and may go beyond what 5G can offer.

From TCO perspective, for the Fully merged cyber-physical world use case, the ownership of the network can be completely or partially in the hands of another stakeholder in the 6G ecosystem other than the Mobile Network Operator (MNO). This is due to the use of the network of networks concept in which cells can be seen as subnetworks, which are integrated with other micro/macro area wireless infrastructures for offloading them from some of the most demanding services. Subnetworks can target both services with extreme requirements and provide the required service level at any location where they are placed as well as, scenarios that can be with extremely dense deployments, such as the case of intra-vehicle cells in a congested road.

Table 5-4 indicates which TCO cost items are impacted by a certain 6G technical enabler (or group of technical enablers). Intelligent network, Flexible network and 6G RAN enablers contribute to the reduction of the RAN infrastructure cost, which is the highest among the five cost items identified (due to the high requirements necessary for this use case). The high cost in the RAN infrastructure comes from the necessity of dynamically adding and removing resources in order to satisfy the subnetwork’s requirements. Obviously, the intelligent network enablers are not only affecting the RAN infrastructure but also have influence on edge and core network infrastructure for large scale fast computation.

Due to the characteristic of the use case, which is based on the subnetwork concept, the enablers for Flexible network are playing the major role in the TCO evaluation. As stated before, the Fully merged cyber-physical world use case has a strict requirement on the RAN infrastructure as well as a low latency requirement on the backhaul. Further, the traffic from this use case involves the core network as well as management and orchestration due to the heavy computation and usage of AI/ML models to provide a realistic mixed reality for the users.

Table 5-4 Hexa-X technical enablers impact to the TCO cost items for the “Fully merged cyber-physical worlds” use case.



Weight	RAN infrastructure	Energy consumption	Backhaul	CN infrastructure	Other NW costs(1)
Intelligent networks enablers (UE and Network programmability, dynamic functions placement, analytics, AIaaS, AI-driven orchestration)	X	X		X	X
Flexible networks enablers (integration of sub-networks, campus, edge-to-Network-Cloud integration)	X	X	X	X	X
Efficient networks (Compute-as-a-Service, CaaS)					
6G RAN enablers (high-data rate links, localization and sensing)	X				
Service management enablers (continuum management and orchestration, AI-driven orchestration)		X			X

(1) This includes people, network management and maintenance costs

5.4.2 Qualitative TCO evaluation for the “Interacting & cooperative mobile robots & flexible manufacturing”

In [HEX-D12] the “*Interacting and cooperative mobile robots*” and “*Flexible manufacturing*” use cases are introduced as part of the “*Robots to cobots*” use case family. In short, for consumer-oriented applications, there might be the need for robots to identify others, connect, exchange intent, and negotiate actions via automated communication, e.g., in construction/building scenarios where different robots need to sync/coordinate their movements to lift of move objects. While, in industrial environments, some production tasks can be conducted by collaboration among mobile machinery, for example, robots collaboratively carrying some goods while being mounted on Automated Guided Vehicles (AGVs). Reliability, functional safety, latency, and positioning requirements need to be met, even if trajectories are blocked or need to be adapted/modified, while communication resources and capabilities need to be assigned and managed by means of a flexible framework. As further detailed in [HEX-D13], humans can be involved or even required in some of direct machine-to-machine interactions, such interactions being either direct (e.g., through mobile or mounted Human Machine Interfaces (HMIs) or jointly working on the same production item) or indirect (e.g., by approaching them as sensed by the system). Functional aspects of these use cases are also reported in [HEX-D71] as part of the dependability in Industry 4.0 environments.

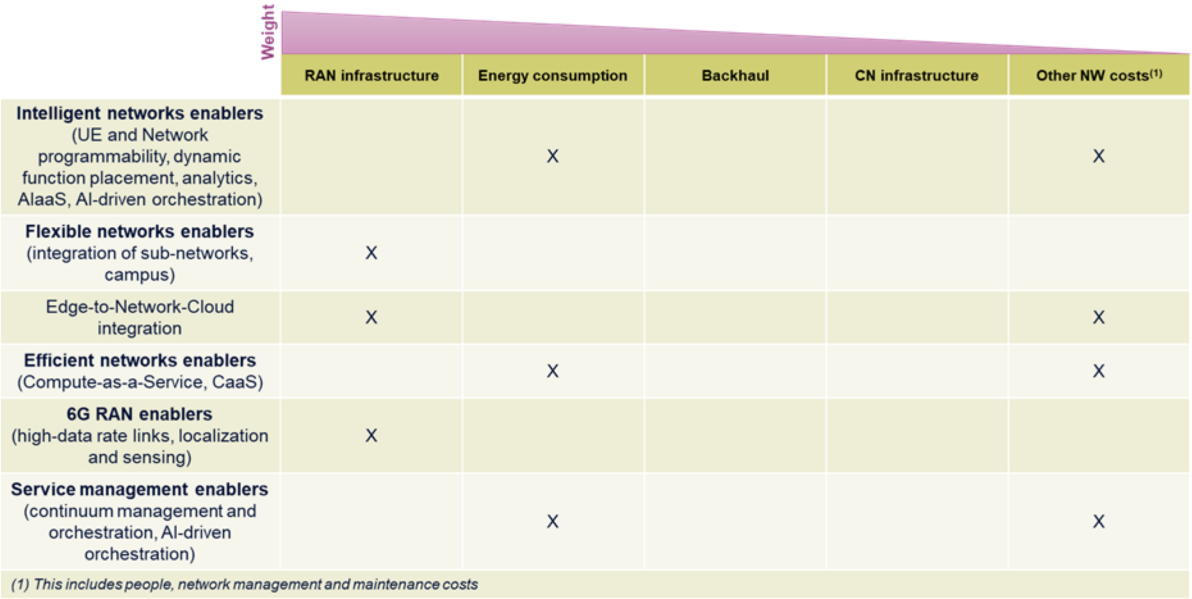
It should be noted that the use case requirements can be partially addressed with 5G, primarily those related to basic communication requirements between robots and the infrastructure and only up to a limited number of robots. The introduction of 6G can bring high added value, with 6G being the enabling platform for efficient AI workload placement in case of impairments as well as for scalable and resilient deployment of distributed/federated AI. Moreover, 6G offers increased computation at Edge-level for lower robot-computation function latencies, joint communication and sensing features for robots’ localization and obstacle detection, increased data rates for massive twinning, as well as increased resilience and trust.

From a TCO perspective, for this use case it is foreseen that the ownership of the network – specifically tailored for the use case itself – can be completely or partially in the hands of another stakeholder in the 6G ecosystem other than the Mobile Network Operator (MNO). As already possible in 5G, also for 6G different private network solutions can be foreseen; for 5G, 3GPP specified solutions in Rel-16 for the so-called Non-Public Networks (NPNs) by introducing two kinds of NPNs (see [21.916]): the Standalone NPN (SNPN) and the Public Network Integrated NPN (PNI-NPN). The former is an end-to-end 5G network isolated from the public network and whose Control Plane (CP) and User Plane (UP) network functions from the Radio Access Network (RAN) to Core Network (CN) are deployed within a private premise and utilizes a dedicated spectrum. The operator of an SNPN – which could be the private enterprise itself or an external third-party operator – has the full control and management of the SNPN network functions. On the other hand, a PNI-NPN is deployed in conjunction with a public network: its infrastructure is integrated with the public network's one based on an agreement reached between the MNO and the enterprise. A PNI-NPN may be provided in different configurations, depending on the degree of the infrastructure sharing between the public network and the private one; the most common configurations are the *NPN Shared RAN* and the *NPN Shared RAN and Control Plane* [HCA+22]. With respect to the first, the configuration termed as *NPN Shared RAN and Control Plane* is achieved by sharing not only the RAN but also the CP network functions (which still reside in the public network), while the User Plane Function (UPF) remains within the private network. Irrespective of the configuration being considered, PNI-NPNs has proved to be a significant cost reduction strategy for MNOs: they allow MNOs to expand their service footprint, coverage and hasten deployment. Savings can accrue from shared equipment, construction, and maintenance costs, hence positively impacting MNOs' CapEx and OpEx cost components. According to the techno-economic analysis in [HCA+22], which provides computations of the percentage of TCO reduction for all 5G NPN scenarios – i.e., SNPN, *NPN Shared RAN* and *NPN Shared RAN and Control Plane* – with respect to the case of legacy 5G network with similar dimensions, a TCO reduction of up to 53% can be achieved for the NPN operator when deploying a PNI-NPN configured as *NPN Shared RAN and Control Plane*, hence showing that TCO (from the NPN operator's perspective) decreases with increase in network integration. Similar benefits in terms of TCO could also be achievable for the "*Interacting & cooperative mobile robots & flexible manufacturing*" use case when realized as a 6G-enabled NPN within the industrial premise.

Table 5-5 indicates which TCO cost items are impacted by a certain 6G technical enabler (or group of technical enablers); it is worth to note that the feasibility of the use case – at least from a technical/functional perspective – has been proven by the Hexa-X project by realizing the "*Handling unexpected situations in industrial contexts*" Proof-of-Concept – refer to [HEX-D72] for details – with several of the technical enablers listed in Table 5-5 being actually used in real-life experiments. From a TCO perspective, it can be observed that enablers for Flexible network significantly impact the RAN infrastructure (RAN subcomponents include passive infrastructure (towers, cabinets), and active infrastructure (radio antennas, as well as baseband processing, and related power and cooling, equipment))– which represents the highest cost component in TCO evaluations, up to 48% for 5G in the "*Strategy #2: enterprise-focused 5G deployment*" [GSM19] – along with the enablers for 6G RAN. The energy consumption is also impacted by different 6G technical enablers, namely intelligent and efficient network, as well as service management enablers, which can increase efficiency in resource allocation, reduce energy consumption and thus OpEx.

At the same time, it should be highlighted that in some cases, extreme performance requirements may also impact a TCO factor such as energy consumption negatively, meaning that not all use case relevant technical enablers contribute to the reduction of TCO. In fact, some may even have adverse impact: for instance, 6G RAN enablers for extreme data rates, including localization and sensing capabilities, may increase the energy consumption; or the addition of extreme-edge domain may lead to increased costs, despite the potential cost savings enabled by AI-driven orchestration.

Table 5-5 Hexa-X technical enablers impact to the TCO cost items for the "Interacting & cooperative mobile robots & flexible manufacturing" use case.



	RAN infrastructure	Energy consumption	Backhaul	CN infrastructure	Other NW costs(1)
Intelligent networks enablers (UE and Network programmability, dynamic function placement, analytics, AIaaS, AI-driven orchestration)		X			X
Flexible networks enablers (integration of sub-networks, campus)	X				
Edge-to-Network-Cloud integration	X				X
Efficient networks enablers (Compute-as-a-Service, CaaS)		X			X
6G RAN enablers (high-data rate links, localization and sensing)	X				
Service management enablers (continuum management and orchestration, AI-driven orchestration)		X			X

(1) This includes people, network management and maintenance costs

However, TCO gains can be achieved, primarily in terms of OpEx, since one of the most important OpEx reduction factors relates to reduction of repair costs. Although CapEx related to the robotic platforms (and related infrastructure) may initially exceed the legacy way of operations in industrial environments, OpEx is reduced in medium/long terms, mainly due to:

i. Greater flexibility in operations (robots are dynamically (re-)programmed/tailored to the changing industrial needs),

ii. highly reduced error rates (robot-powered operations offer much higher accuracy in terms of quality inspection operations),

iii. highly reduced downtimes of impaired robotic platforms (automated identification of impairments and digital twin/virtual reality powered teleoperation radically reduces the time to repair),

iv. AI workload re-allocation to functioning robots ensures that the operations continue seamlessly without stalling the inspection/production processes.

5.5 Developing interfaces for AI/ML driven orchestration and supporting executing agents

Mobile networks generations previous to 5G were designed to work in a "per-domain" way and the interaction was (mostly) limited to peer-to-peer reference points within the very same domain (i.e., S11 interface between 3GPP 4G MME and S-GW functions). The continuous development towards more intelligent networks has driven many standards to adopt the Service Based Architecture (SBA) foundations [23.501] to generate Service Oriented Architectures (SOAs) [GLN+14]. These approaches overcome some of the constraints of the typical classic reference-point-based architectures as they allow the different NFs to communicate and consume services from other NFs. Besides, they open-up the door to the implementation of technologies such as AI/ML, Big Data Analytics, etc. in order to add the capability of managing a larger number of services in future mobile networks e.g., B5G/6G mobile networks. Thereupon, to be able to add features such as dynamic function placement, high degrees of network automation, intent-based networking, services self-optimization and so on, a data-driven approach is required, and new types of endpoints should be created that support multi-domain/multi-stakeholder scenarios (see [HEX-D52] Section 3.3.2).

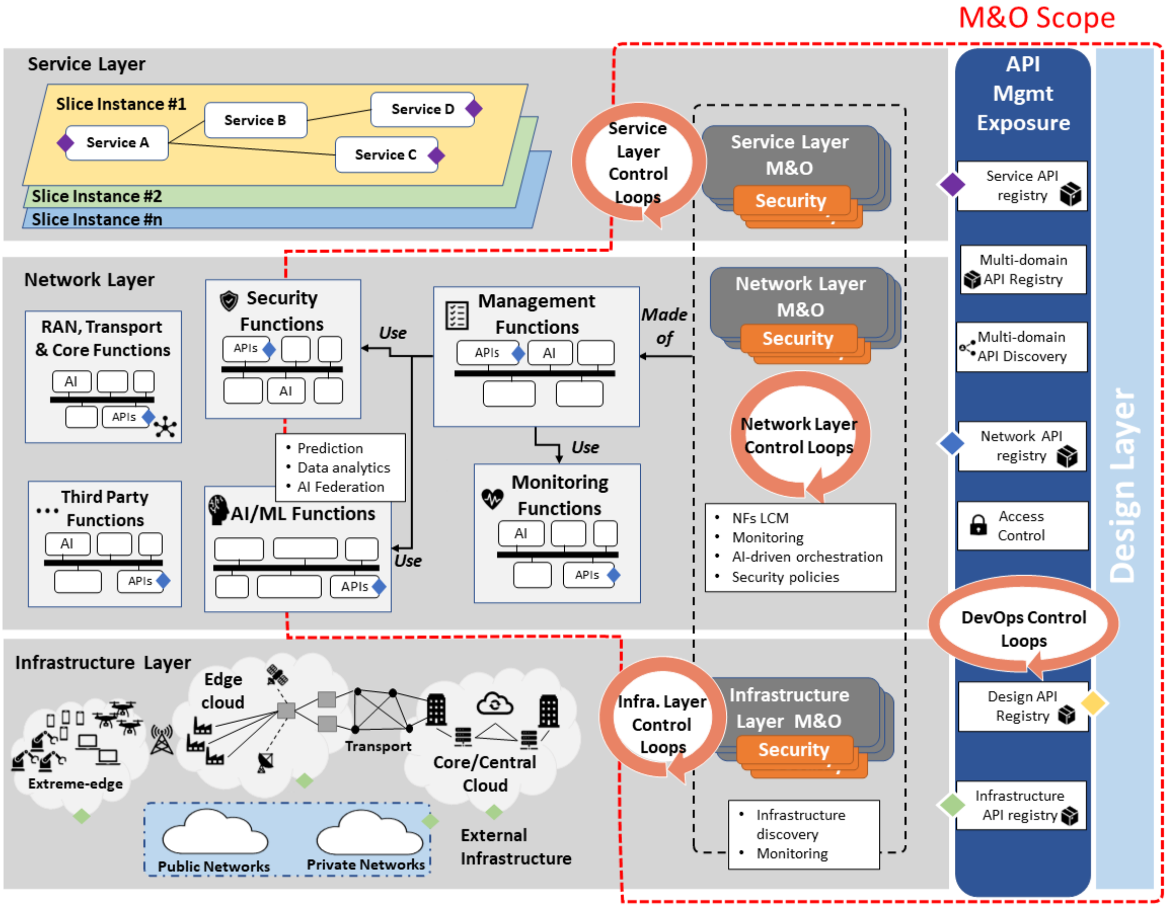


Figure 5-12 WP6 M&O Framework API Management Exposure concept [HEX-D62].

In order to achieve these M&O capabilities, the following features are required:

- **Flexible Cross-domain data exchange**: In a multi-domain environment, NFs and services from different administrative domains should be able to exchange information and cooperate, with other NFs and services from other domains. Such communication should be carried-out consuming endpoints exposed by these external domains.
- **Capability Exposure**: The capacity of an NF or service to securely expose its M&O capabilities towards an authorised consumer through an endpoint. It is a key aspect in order to allow verticals from other domains or external services/NFs to inter-communicate with a given administrative domain services or Ns.
- **Capability Exposure Levels**: Multi-domain/multi-stakeholder environments will require to stablish different levels of exposure depending on what or who wants to access a service or NF [5GVIN-D31] I.e., a particular service provider (consumer) may have full endpoint configuration capabilities to a Network Slice (producer) while a NF with monitoring capabilities (consumer) is able to access some of this Network Slice API features.

Hexa-X WP6 has proposed a M&O architecture that includes a block called *API Management Exposure* [HEX-D62] which aims at fulfilling the aforementioned features. Figure 5-12 depicts an adaptation of the original M&O architecture with a special focus on the *API Management Exposure* block. With the help of this block, all the network components in the various layers can interact and communicate with one another at different levels (i.e., capability exposure) while still adhering to a common pattern, as described in [HEX-D62]. As it can be seen in Figure 5-12, the *API Management Exposure* supports API registration within an administrative domain (e.g., Service, Network, Infrastructure and Design APIs) but also supports multi-domain API registries and API discovery. Additionally, it is able to provide access control policies that enable the so-called *capability exposure levels*, in order to provide different levels of access depending on the entity that makes requests against the respective API.

It is worth mentioning that this framework is able to integrate a wider scope beyond M&O resources, as it exploits the principles of the 3GPP Common API Framework (CAPIF) [23.222] and has a similar behaviour to the Zero-Touch Service Management (ZSM) cross-domain integration fabric [zsm-002]. Moreover, this framework takes advantage of the *Design Layer* to enable *DevOps*-related capabilities towards API M&O: API automated testing, API automated deployment, API version management, flexible API instantiation, etc. Below, an enumeration of the main features that the *API Management Exposure* should support is given [HEX-D62]:

1. API Discovery
2. API Registration
3. Access Control
4. Traceability between APIs requests
5. Routing across APIs

5.6 CaaS framework

In Hexa-X Deliverable D5.2 [HEX-D52], Section 5.7, a Software Reconfiguration Framework was introduced as defined by the European Telecommunications Standards Institute (ETSI) Technical Committee Reconfigurable Radio Systems (RRS), including a definition of a key Interface, i.e., the generalised Multiradio Interface (gMURI) [303681-1].

In October 2022, ETSI published a Technical Specification TS 103 850 [103850] which defines the format of a Radio Application Package (RAP) that is being used to provide a single or multiple Radio Application (s) and related information to a compute framework.

The existing overall structure is outlined in Figure 5-13.

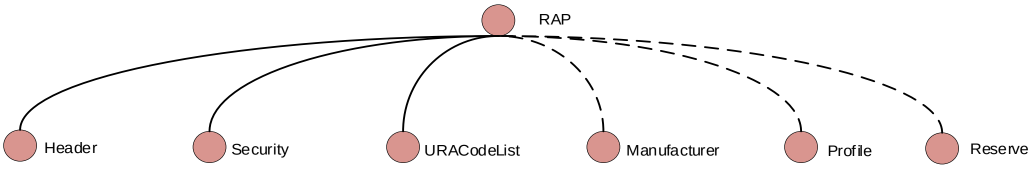


Figure 5-13 Top Level tree structure as defined by ETSI TS 103 850 [103850].

We propose an extension of the upper RAP structure to be applicable for delegating/ offloading generic application related workloads (besides radio signal processing ones) to networked compute nodes based on different computing platforms.

Towards this end, it is proposed to define the content of the "Reserve" information element of the RAP data structure as outlined above. The “Reserve” element is proposed to be transformed into a structure with a number of attributes, as described below (alternatively, the attributes proposed below can be also added to the tree structure above at the same level as the Reserve element, which in that case, can be maintained):

- Element “**Regulation_Conformity**”: Documentation on how requirements on regulations are being met, in particular related to the requirements of the draft AI Act [AIAct+21].

The draft AI Act [AIAct+21] introduced a series of articles among which there are articles introducing technical requirements to be met by AI Systems to be admitted to the European Single Market. We propose that the RAP (within the proposed “Regulation_Conformity” element) documents how the regulation requirements are being met.

The regulation requirements are summarized in Table 5-6:

Table 5-6 Requirements outlined by AI Act [AIAct+21].

Requirements	Summary as defined by the AI Regulation [AIAct+21]
Data and data governance	High-risk AI systems … shall be developed on the basis of training, validation and testing data sets that meet the quality criteria ...
Technical documentation	The technical documentation shall be drawn up in such a way to demonstrate that the high-risk AI system complies with the requirements …
Record keeping	High-risk AI systems shall be designed and developed with capabilities enabling the automatic recording of events ('logs') …
Transparency and information to users	High-risk AI systems shall … ensure that their operation is sufficiently transparent to enable users to interpret the system's output and use it appropriately …
Human oversight	High-risk AI systems shall be designed and developed in such a way, including with appropriate human-machine interface tools, that they can be effectively overseen by natural persons during the period in which the AI system is in use …
Accuracy robustness and cybersecurity	High-risk AI systems shall … achieve, in the light of their intended purpose, an appropriate level of accuracy…
Risk management system	A risk management system shall be established, implemented, documented and maintained in relation to high-risk AI systems …
Quality management system	Providers of high-risk AI systems shall put a quality management system in place that ensures compliance with this Regulation …

For each of the requirements, it is proposed to introduce the following sub-Elements as "children" of the proposed Regulation_Conformity attribute of the RAP "Reserve" attribute (or as additional attributes under the root of the RAP structure).

Table 5-7 Sub-Elements ("children") of the proposed Regulation_Conformity attribute of the RAP "Reserve" attribute.

Requirements	Proposed new sub-elements
Data and data governance	Bit: "Requirement met": 0 = requirement not met (likely no market access allowed in the European Single Market), 1 = requirement met If applicable (if requirements are not met for all cases, but under specific conditions): Bit "Requirements met under specific conditions" = 1, following by a definition of the conditions, e.g., for a specific high risk application (as defined in the annex of [AIAct+21], for example biometric detection applications, critical infrastructure applications, etc,). Further details on how to make sure that the regulation requirements are being met (possibly in text form or in machine readable representation): High-risk AI systems … shall be developed on the basis of training, validation and testing data sets that meet the quality criteria ...
Technical documentation	Bit: "Requirement met": 0 = requirement not met (likely no market access allowed in the European Single Market), 1 = requirement met If applicable (if requirements are not met for all cases, but under specific conditions): Bit "Requirements met under specific conditions" = 1, following by a definition of the conditions, e.g., for a specific high risk application (as defined in the annex of [AIAct+21], for example biometric detection applications, critical infrastructure applications, etc,).

	Further details on how to make sure that the regulation requirements are being met (possibly in text form or in machine readable representation): The technical documentation shall be drawn up in such a way to demonstrate that the high-risk AI system complies with the requirements …
Record keeping	Bit: “Requirement met”: 0 = requirement not met (likely no market access allowed in the European Single Market), 1 = requirement met If applicable (if requirements are not met for all cases, but under specific conditions): Bit “Requirements met under specific conditions” = 1, following by a definition of the conditions, e.g., for a specific high risk application (as defined in the annex of [AIAct+21], for example biometric detection applications, critical infrastructure applications, etc,). Further details on how to make sure that the regulation requirements are being met (possibly in text form or in machine readable representation): High-risk AI systems shall be designed and developed with capabilities enabling the automatic recording of events (‘logs’) …
Transparency and information to users	Bit: “Requirement met”: 0 = requirement not met (likely no market access allowed in the European Single Market), 1 = requirement met If applicable (if requirements are not met for all cases, but under specific conditions): Bit “Requirements met under specific conditions” = 1, following by a definition of the conditions, e.g., for a specific high risk application (as defined in the annex of [AIAct+21], for example biometric detection applications, critical infrastructure applications, etc,). Further details on how to make sure that the regulation requirements are being met (possibly in text form or in machine readable representation): High-risk AI systems shall … ensure that their operation is sufficiently transparent to enable users to interpret the system’s output and use it appropriately …
Human oversight	Bit: “Requirement met”: 0 = requirement not met (likely no market access allowed in the European Single Market), 1 = requirement met If applicable (if requirements are not met for all cases, but under specific conditions): Bit “Requirements met under specific conditions” = 1, following by a definition of the conditions, e.g., for a specific high risk application (as defined in the annex of [AIAct+21], for example biometric detection applications, critical infrastructure applications, etc,). Further details on how to make sure that the regulation requirements are being met (possibly in text form or in machine readable representation): High-risk AI systems shall be designed and developed in such a way, including with appropriate human-machine interface tools, that they can be effectively overseen by natural persons during the period in which the AI system is in use …
Accuracy robustness and cybersecurity	Bit: “Requirement met”: 0 = requirement not met (likely no market access allowed in the European Single Market), 1 = requirement met If applicable (if requirements are not met for all cases, but under specific conditions): Bit “Requirements met under specific conditions” = 1, following by a definition of the conditions, e.g., for a specific high risk application (as defined in the annex of [AIAct+21], for example biometric detection applications, critical infrastructure applications, etc,). Further details on how to make sure that the regulation requirements are being met (possibly in text form or in machine readable representation):

	High-risk AI systems shall … achieve, in the light of their intended purpose, an appropriate level of accuracy…
Risk management system	Bit: “Requirement met”: 0 = requirement not met (likely no market access allowed in the European Single Market), 1 = requirement met If applicable (if requirements are not met for all cases, but under specific conditions): Bit “Requirements met under specific conditions” = 1, following by a definition of the conditions, e.g., for a specific high risk application (as defined in the annex of [AIAct+21], for example biometric detection applications, critical infrastructure applications, etc,). Further details on how to make sure that the regulation requirements are being met (possibly in text form or in machine readable representation): A risk management system shall be established, implemented, documented and maintained in relation to high-risk AI systems …
Quality management system	Bit: “Requirement met”: 0 = requirement not met (likely no market access allowed in the European Single Market), 1 = requirement met If applicable (if requirements are not met for all cases, but under specific conditions): Bit “Requirements met under specific conditions” = 1, following by a definition of the conditions, e.g., for a specific high risk application (as defined in the annex of [AIAct+21], for example biometric detection applications, critical infrastructure applications, etc,). Further details on how to make sure that the regulation requirements are being met (possibly in text form or in machine readable representation): Providers of high-risk AI systems shall put a quality management system in place that ensures compliance with this Regulation …

The proposed extensions are complementary to *Flexible compute workload assignment* related solutions as defined in Hexa-X Deliverable D4.3 [HEX-D43]. Both approaches can be combined as appropriate.

5.7 Handover

Efficient mobility procedures are a main characteristic of cellular systems, i.e., the ability to maintain a call when moving from a cell controlled by one base station to a cell controlled by another base station. One of the most important parts of the process is to decide when control and data streams are exchanged between the two cells. In the following sub-sections, some areas of improvements are discussed.

5.7.1 CaaS handover for 6G

In legacy networks, when a UE is connected to the network it is typically configured to measure the signal quality of the serving and neighbouring cells to determine whether the connection to the current serving cell is sufficiently good or if the UE would be better suited to handover to another cell.

These handover configurations consist both of a measurement configuration and a reporting configuration. The measurement configuration consists of different measurement objects which indicate various parameters e.g., carrier frequency, cell identifiers, offsets, or thresholds. The report configuration describes when and what measurement results the UE shall report to the network. In this regard, there are several events defined that can be configured, that will trigger the UE to send a measurement report to the network [38.331].

Thus, the UE will continuously measure on its serving cell, and it may measure on neighbouring cells, on the same or different frequency and/or same or different radio access technology (RAT), e.g., depending on how good the signal quality is of the serving cell.

If the UE measures on a neighbouring cell and determines the measurement results fulfil one of the configured reporting events (e.g., neighbour cell is better than serving cell by defined threshold), the UE will send a measurement report to the network with e.g., RSPR (Reference Signal Received Power), RSRQ (Reference Signal Received Quality) and/or SINR (Signal plus Interference to Noise Ratio) for the serving and neighbouring cells. Since the network configured the UE to only report measurements if the reporting condition is fulfilled, the network will typically handover the UE to the cell with the best signal quality according to the measurement reports.

In addition, D5.1 [HEX-D51] introduced the concept of CaaS where a UE can offload demanding computations to the network for, e.g., applications such as XR, advanced AI/ML evaluations, sensing/localization, gaming, etc., in order to conserve battery power, or allow more lightweight devices. If the service requirements for the offloaded service stipulates a maximum computational roundtrip time, e.g., in case of streaming XR video processed in the network, the E2E latency to the computational resource may be too large even though the throughput requirements can be moderate.

However, the UE may be connected to a network node which provides quite long E2E latencies, even if the throughput is acceptable (see Figure 5-14). For instance, in case the UE is connected via an IAB-node (i.e., the network node the UE is connected to, is itself connected wirelessly to another network node), the computational offloading resources may be located in the wired network which would require one or more hops between IAB nodes, where each hop would add to the latency.

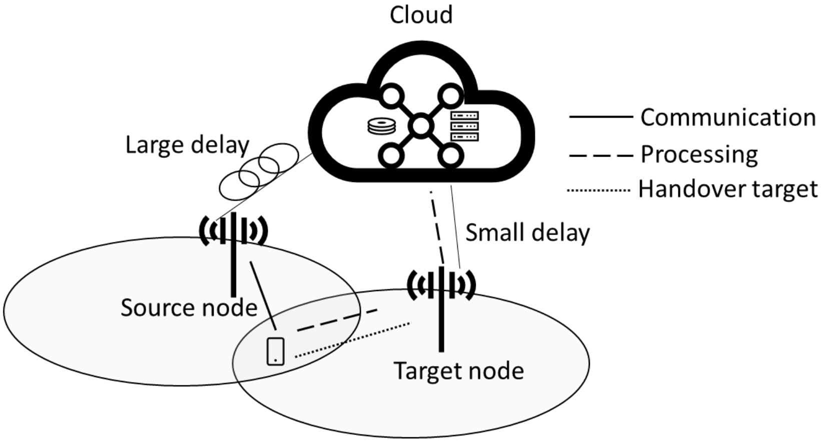


Figure 5-14 Overview of the CaaS handover solution.

If the UE is within coverage of another cell, with a different network path towards the computational resources, it could be beneficial for the UE to handover to that cell, even if the signal quality may be slightly worse.

To enable the network to consider the E2E latency in the handover decision, the network would first need to know the E2E latency towards the preferred offloading resource for the UE. Secondly, the network would have to modify the handover procedure to take this latency into account. For the transport latencies, the network would maintain a mapping of the latency between each RAN node and computational node. For the handover decision, one option to account for the delay would be to introduce a measurement offset, e.g., proportional to the delay when the UE is reporting the signal measurement results to the network.

The network is already able to introduce cell-specific measurement offsets, the so-called q-offset, which are added by the UE to the measurements of each cell before determining whether these fulfil the reporting criteria. However, these current q-offsets are typically static, e.g., differentiating between macro- and pico-base stations and does not account for the E2E latency.

To provide a compromise between the throughput and the computational offloading RTT, the network would first determine which would be the suitable computational resources capable of serving the UE offloading and which computational RTT each of these resources would provide for any cell in the vicinity (e.g., within the same and neighbouring Tracking Areas). Depending on the UEs latency requirements for offloading, the so-called q-offset for each cell could be set to prioritize cells with low latency and down-prioritize cells with high latency.

As an example, consider a UE connected to a serving cell which would provide a computational latency twice as large as UE requirements. A neighbouring cell with slightly worse signal strength that could provide a computational latency at half the UE requirement. The network could then configure the q-offset for the serving cell to -3 dB and to the neighbouring cell to +3 dB. When the UE performs the cell measurements, if the neighbouring cell is only 5 dB (or less) worse than the serving cell, the UE would report the neighbouring cell as the best cell and the network would typically handover the UE to that cell. This method provides both reliability and better compute latency to the user.

A few reasons the serving cell can experience significantly larger computational roundtrip times than a neighbouring cell could be:

- An IAB node, which in turn can be connected to several IAB donors and one IAB parent, and where each IAB hop causes an extra delay
- A cell with high load, which causes substantial scheduling delays (e.g., in the order of 10 ms) but the UE still has relatively good link quality to the said cell
- NTN nodes, which may require multiple hops between NTN nodes and the signal need to traverse long distances until they reach a ground station and the core networks and internet.

The UE can send a “computational offload request” including required resources (CPU, storage etc) and the maximum delay the UE can tolerate to the serving RAN node. The serving node checks if the stored latency (the average computing latency for UEs connection to computing resources) can fulfil the requested maximum delay from the UE. If the serving node determines that it cannot fulfil the request, it sends the computational offload request to adjacent network nodes. This solution enables a compromise between the user throughput (signal quality) and an efficient computational offloading.

5.7.2 Sensing assisted handover

Joint communication and sensing (JCAS) is a technique that relies on the radio resources initially used for communication to also position objects (within the range of the radio). There is a range of different ways to implement sensing. Differences between methods affect complexity to implement and accuracy of any determination. A basic method is to analyse the responses of ordinary measurements, e.g., the sounding reference signal (SRS). When a vehicle crosses an SRS transmission the impact of the scattering channel can be measured by the base station and the scattering from the vehicle can be analysed, e.g., using Doppler filtering. The other end of the range is to design a radar-like pulse that is transmitted towards the target and from the time when the reflection is received a distance to the target can be calculated. There is then a long list of options, e.g., the transmitter and receiver are collocated (i.e., monostatic), transmitter and receiver are in different locations (i.e., bistatic)

As a means to optimize the mobility process, location-based mobility has been discussed for many years. Early papers describe how the location of a GSM UE is used to optimize the process, e.g., by reducing the amount of ping-pong signalling [JLL05]. In more recent studies, location-based handover is studied for high-speed trains. High-speed is one factor that can disturb the normal handover signalling and help of location information the process becomes more robust [CYZ15]. With location-based mobility the actual location and direction are used to initiate the handover process. This is being suggested for NTN scenarios in which the devices report the current position or distance to a reference point, e.g., the centre of the cell, instead of reporting radio measurement as for conventional mobility procedures [JLW+22].

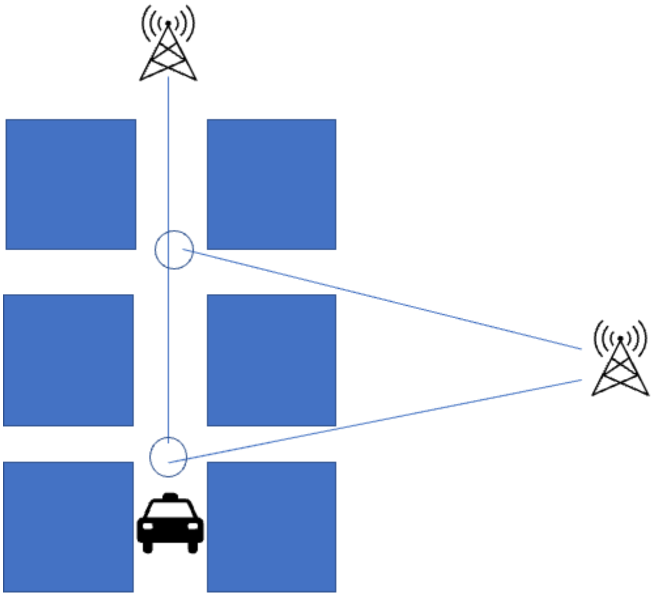


Figure 5-15 Principle for how sensing can help improve handovers.

Although NTN is a special case there may be reasons to adopt a similar process for terrestrial networks. Handover optimizations are frequently studied by 3GPP, e.g., there is a new work item in Rel-18 “Further NR Mobility Enhancements” [RP213565], which suggests that there is room for improvement. An efficient implementation of sensing may help make the handover process more efficient. Figure 5-15 shows how localisation using sensing can be used to improve handovers and possibly reduce signalling. Using sensing, the direction and speed of the vehicle can be measured and even predicted. Based on the prediction of UE movement (assume that the UE moves upwards in the figure) even though measurements show good radio conditions the UE remains connected to the tower at the top of the picture.

5.8 Microservice-based SDN controller

Originally, network softwarization implied the direct translation of network hardware functionalities into software modules. The initial correspondence between the hardware functionalities and resources and their abstractions was 1:1. However, in order to increase the flexibility and adaptivity of the network, network functionalities have started being ‘split' in sub-functions, with the possibility of distributed placement. A well-known example could be the functional split of the softwarized baseband unit of the base station [LCC19]. These sub-functionalities have become more and more microservices that are run and activated on demand according to the needs of the network. The problem of the functional split (microservice-based realisation) of any network softwarized network functionality is the first problem to be solved when the softwarized network continuum consists of microservices and agents. Especially, the former represents the first step to then obtain a set of sub-functions that can subsequently be equipped with intelligence to become agents. In this way, multi-agent systems are created to employ in-network intelligence in the performance of distributed and split softwarized network functions. The problem of splitting a softwarized network function in microservices or agents has not a unique solution. The following text shows a possible solution to this problem for the SDN controller.

Some seminal ongoing efforts [O21] have been proposed with the idea of disaggregating the SDN controller architecture into microservices, each of which can be responsible for a certain task of the controller. These microservices are usually implemented via Docker containers. This can enable the flexible deployment of controller functionalities. Efforts to perform a functional split of the SDN controller are ongoing (e.g., μONOS, ETSI TeraFlow [T23]) so that a full decomposition of the SDN controller can enable efficient and effective microservice-based operations. The ONOS project has proposed μONOS, which is the next-generation architecture for the Open Network Operating System controller [O21]. μONOS adopts a microservices-based architecture splitting the controller and the core itself as an assembly of various microservices. However, μONOS has been specialized mainly for cloud datacentre scenarios by employing a service orchestrator, Kubernetes, to manage microservices that are realized as Docker containers. μONOS' is still ongoing to provide a playground framework [O21]. In addition, this approach has some limitations: first, its limited to certain technologies, not all 5G compliant, for instance, Kubernetes instead of ETSI MANO or containers instead of VNFs. Second, inter-functionalities communication is limited to Google Remote Procedure Call (RPC), which does not give a fair degree of flexibility in certain scenarios. Finally, the implementation is not completed yet. TeraFlow targets a cloud-native architecture, with transport network integration. Moreover, it applies ML for security, and it has a distributed design and approach [T23]. TeraFlow OS principal components are Context Management, Monitoring, Traffic Engineering, Device, SDN Automation, Policy Management, and Slice Management [T23]. The Context Management is responsible for storing the configurations and attributes of the different network elements managed by the TeraFlow OS. It stores the active contexts, topologies, devices, links, and the services created. It does not employ No-SQL database to optimize the concurrent access into the same storage infrastructure to target scalability. Next, the Monitoring component manages monitoring in the controller where subscribers can subscribe for receiving information about metrics or KPIs, coming from different parts of the system. Traffic Engineering is mainly responsible for setting up and optimizing Segment Routing paths in the infrastructure exposed by the Device component. The Device component interacts with the underlying network equipment. The SDN Automation element deals with the design of southbound and northbound interfaces of SDN while the Policy Management consists of a collection of rules that implies the behaviours of network resources. Finally, the Slice Management element uses the Network Slice Controller to realise a transport network slice, using physical and virtual network resources provided by the underlying network controllers.

The following text shows a decomposition of an SDN controller into a set of microservices. Communications issues among the controller's microservices are analysed. The design and performance considerations of microservice-based SDN controller are studied via the implementation of a specific functional split based on the Ryu SDN controller [ASB+22]. This can show the challenges and some main capabilities of the system. Different network communication protocols, such as gRPC, WebSocket, and REST-API are used in the implemented system. The experimental results highlight the robustness and latency of the microservice-based system.

An initial question could be: why do we need distributed approaches based on microservices and agents? The main issues of a centralized control plane relate to latency constraints to fault tolerance and load balancing. Existing controllers have been implemented as 'monolithic' entities, even in the case of distributed deployments. In particular, in the case of distributed SDN controllers, the distribution consists of replicas of the SDN controller itself, which means all SDN sub functionalities are replicated even if not all of them are necessary. For instance, Ryu SDN Controller, an open-source SDN controller implementation, provides a single piece of code installable on heterogeneous operating systems that enables the machine (or virtual machine) to act as an SDN controller. At the current time, all opensource and proprietary releases of SDN implementations adopt a 'monolithic' software approach, which include ONOS, OpenDayLight, and Floodlight [ASB+22]. The main issue of these implementations is that they do not allow network administrators and developers to choose SDN components and/or functionalities to be (de-)activated. This results in limited flexibility in highly changing network scenarios and creates multiple problems in terms of scalability, fault isolation, and latency. At the same time, the legacy definition of the SDN reference architecture does not mandate the internal composition, implementation, and design of an SDN controller. Thus, the SDN controller can be decomposed and implemented as a set of software components, running in a distributed manner. In this regard, the ONOS project proposed μONOS, which is the next-generation architecture for the Open Network Operating System controller. As previously mentioned, μONOS adopts a microservices-based architecture disaggregating the controller and the core itself as an assembly of various subsystems. Comparison between SDN monolithic architecture and microservices-based SDN architecture:

Scalability	*Monolithic.* It is difficult to scale because it requires replicating overall SDN controller for smaller increase in demands. *Microservice-based.* It can be placed in a container or any virtual environment. Using function sequencing, dynamic instantiation of containerized services, and function orchestration, it is possible to recreate the SDN functionality, which can easily scale by adding only those components that require additional resources.
Cloud Readiness	*Monolithic.* It is bulky to deploy the system in a containerized environment. *Microservice-based.* It can easily be deployed and orchestrated in containerized environment. Therefore, a microservice-based design of SDN controller enables easy deployment in a distributed and dynamic environment such as the softwarized network continuum.
Loose Coupling	*Monolithic.* Its internal modules are tightly coupled which prevents it to be deployed in a distributed environment without replicating the controller. Wherever and/or whenever SDN controller functions are needed, the whole system must be deployed instead of the required functions only. *Microservice-based.* It has loosely coupled components which enable easier deployment in a distributed environment for dynamic scaling along with the dynamic service demand.
Maintenance	*Monolithic.* If an internal component fails, locating the problem to make changes is difficult and may take a lot of time. This is because the SDN controller is a complex set of code that is tightly coupled. *Microservice-based.* It has enough decoupling of the functional components to identify, isolate, and replace them without requiring replacing the whole system.
Component reuse	*Monolithic.* It is difficult to reuse the sub-components as every function or internal modules are part of a single system. *Microservice-based.* Components could be reused and orchestrated to be deployed with other functions for dynamic response to the workload increase. So, the functions are reusable as components loosely coupled, independently implemented and deployed.

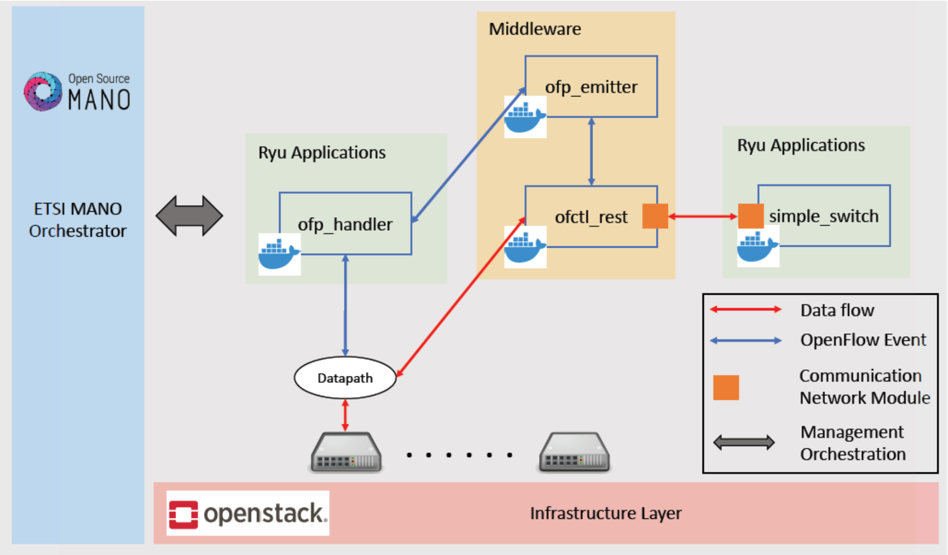


Figure 5-16 Architecture of a Ryu microservice-based implementation of the SDN controller [ASB+22].

In the Ryu implementation in [ASB+22], some of the essential modules can be identified that describe a basic SDN system:

- *Event Handler System Management*: this module is in charge of catching an OpenFlow event and forwarding it to the destination. This module works reactively and may be considered as the core module for a decomposed SDN implementation.
- *Routing System*: this function is used to generate flow rules to allow the network to exchange packets among nodes and switches.
- *South-bound Management*: this module allows the system to interact with the underlying system with several protocols.

The design was driven by the following methodology. First, isolation of the event emitter from the core of Ryu Framework and creation of a support middleware module (the yellow block in Figure 5-16), incorporating the REST APIs block with the emitter to be able to transform events in REST calls. The middleware is the fundamental block for a microservices-based SDN decomposition, precisely because it connects the legacy SDN environment with external microservices. Second, turning each Ryu App in a separate block (i.e., microservices) external to the Ryu Framework, which can communicate with the Framework via REST APIs, through the middleware. In this way, it is possible to transform an SDN functionality into a microservice. For implementation purposes, the already existing REST-based APIs is used in the Ryu framework, precisely the *ofctl_rest* module. Figure 5-16 shows the resulted architecture of the Ryu microservice-based implementation. The described approach can be used for different network technologies, not only REST-based, such as gRPC, WebSocket, RPC, and so on. What is changing is the block internal to the middleware (the ofctl*_rest* block in Figure 5-16 module that connects to external microservices.

Next, this specific solution adopts Docker Container as containerization ecosystem, Open Source MANO for the orchestration and OpenStack as the infrastructure layer [O23]. The middleware is a Docker container that incorporates the *ofp_emitter* and ofctl_rest blocks inside and another Docker container for the event handler functionalities, such as, the *ofp_handler* block. Finally, Ryu Apps are considered as separated Docker Containers that include SDN functionalities comprising routing functionality or Firewall.

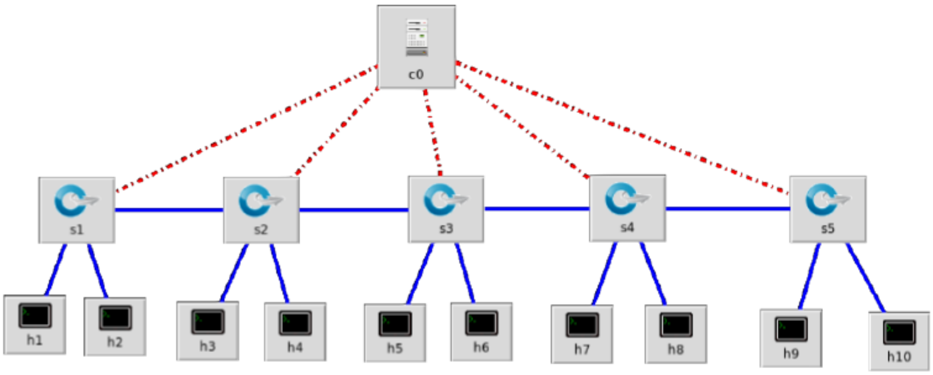


Figure 5-17 Mininet Topology for Experimental Testbed [ASB+22].

Figure 5-17 depicts the experimental scenario. As can be seen, there are two switches between H1 and H3, and there are five switches between H1 and H10. H1. H10 represent the different microservices (Ryu apps). To calculate the end-to-end latency, a video streaming is sent across the network and kept the average round trip time. This is repeated for different nodes of the network (H1 and H3 first, and then H1 and H10).

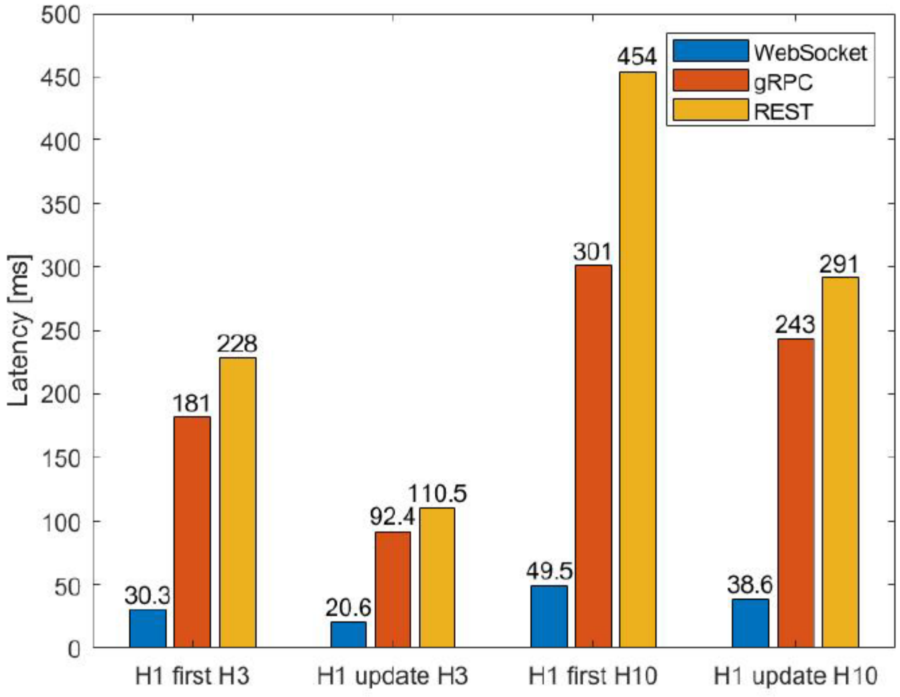


Figure 5-18 Latency measurements [ASB+22].

The measurements in Figure 5-18 show that the major delay is on the first packet latency. The REST protocol is seven times slower than Web-Socket technology, in H1 and H3 scenario, whereas the gRPC protocol provides performance like the REST protocol but a little faster. The performance further degrades in the H1 and H10 scenario for all protocols. In particular, the REST protocol is around ten times slower than WebSocket technology. This is due to the multiple connections between switches. In these experiments, once the exchange of the first packets is completed, switches only introduce delay for the forwarding to the specific selected port. Finally, the performance of all protocols during the normal flow was omitted due to the very low latency time (0.01 ms average around all protocols) but proves that all protocols are consistent and similar to each other. In conclusion, we note that the REST protocol has a high response time for the first packet and rule updating packets compared to WebSocket and also to standard Ryu. However, the response time during the normal flow remains the same for all protocols. Therefore, it is apparent that the benefits of the microservice-based SDN model need to be balanced with any trade-offs incurred. However, despite that the WebSocket protocol proves to be faster, it strongly depends on the Socket concept which relies on the IP address and the Port number of the services. The gRPC protocol could become dominant in the future thanks to the adoption of the HTTP/2 protocol. Furthermore, factors such as scalability and reliability (or availability) should be taken into account when deciding whether to use standard SDN or the microservices-based one. Moreover, the best choice of the right communication protocol depends on many factors including the context. For instance, a heterogeneous and ultrareliable industrial scenario may require REST as a communication protocol to guarantee high connectivity among devices.

6 Quantified targets

One of the five Hexa-X objectives defined by the project [HEXA] is the “Network evolution and expansion towards 6G”. This is the main objective of WP5 and aims to develop architectural components for 6G that support a new flexible network design, full AI integration and network programmability while, at the same time, streamline and redesign the architecture for a network of networks.

However, the objective also has four so called quantified targets:

- Access links supporting simultaneous high data rate and low E2E latency (>0.1 Tbit/s @ <1 ms E2E)
- Supporting (>100 bn) connected devices in the network
- (>99%) of global population reached with (>1 Mbit/s) data rates at sustainable cost levels
- Full coverage (100%) of world area.

In this chapter, we will describe the methodology for how to fulfil these quantified targets.

6.1 Simultaneous high data rate and low E2E latency

This quantified target is about the capability to support high bit rates (above 0.1 Tbit/s) and low latency (below 1 ms) simultaneously. In this analysis, we do not consider session set up times or any other actions preceding the session. Instead, our focus is on the DL of an E2E packet flow. For the feasibility of high bit rates, we refer to data rate analysis of Table 3.2 in [HEX-D21]. In this section we consider how to achieve the E2E latency target. Latency is the time measured from when a data packet leaves a server application to the time when the DL data arrives at the application in the UE assuming the processing time in UE is the radio layer stack processing plus the IP-stack processing. As the data packets vary in length, we consider short non-segmented packets without any retransmissions or packet losses in a limited area.

We want to find out the latency/delay budget of the E2E path. The latency contributing factors on the E2E path depend on the RAN and CN configuration in addition to the distance between the UE and the application server. There are two different scenarios to be considered as depicted in Figure 6-1.

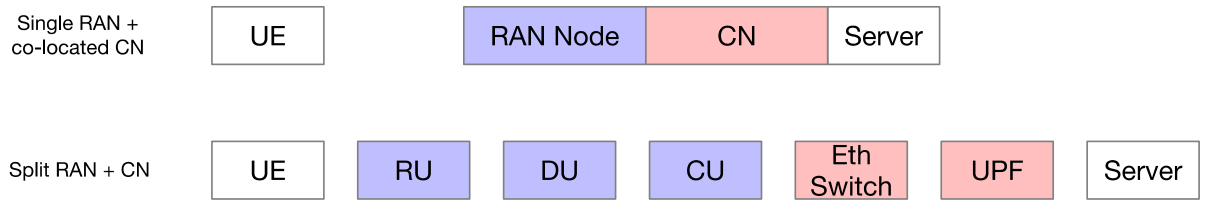


Figure 6-1: RAN and CN configuration scenarios.

In the classical single RAN configuration, all RAN functionality is integrated into a single RAN node. If CN UPF functionality is co-located with the RAN functionality the number of hops a packet needs to traverse is minimised. Co-locating CN with RAN also hides the impact of possible refactoring of network functionality as this becomes an internal matter of an implementation.

In split RAN configuration, the RAN functionality is split into separate Radio Unit (RU), Distributed Unit (DU), Centralized Unit (CU) functions that can be distributed or combined in multiple ways which means that different deployment options need to be considered depending on how the RAN elements are located and how far the application server is located from the RAN. Respectively, CN functionality like UPF can be moved away from the RAN elements as shown in Figure 6-1.

The radio latency estimate for UE and RAN is common among the studied RAN configuration scenarios of Figure 6-1. The estimate is based on the information from [HEX-D2.1] that states that bandwidth requirements for achieving 100 Gbit/s with a single-stream transmission are quite high, even for higher spectral efficiencies. This implies that a multi-stream (MIMO) transmission with at least two to four parallel streams should be employed, either as point-to-point or distributed fashion (D-MIMO). The choice of numerologies above 480 kHz sub carrier spacing has 90 - 100 μs latency over the radio link (PHY layer) and using multiple streams (2-4 parallel) to achieve 100 Gbit/s (see the latency analysis in Chapter 5.1 of [HEX-D23]). Thus, 100 μs will be used for radio latency in our analysis.

The limiting factor for the latency is not numerology or bandwidth (assuming that they are large enough), but UE and signal processing times in RAN (called BS processing time in [HEX-D23]). From [HEX-D23], we assume that the UE and signal processing time in RAN (for PHY layer) is more or less independent of the slot duration. Further on, in [HEX-D23], α is the expected evolvement of future processing time advances where $0 < \alpha < 1$. The processing times for the UE would become as $t_{UE,tx} = \alpha\, 98.2\, \mu s$ and for RAN as $t_{BS,tx} = \ \alpha\, 80\, \mu s$ (the values are taken from [38.214], Table 6.4.-2 and Table 5.3.-2). In the following estimation, we assume an $\alpha = 0.1$, i.e., a factor 10 times better processing time than used in [38.214] and set the PHY layer processing time to $t_{BS,tx} = \ 8\, \mu s$. The 8 μs value is then used for each layer above PHY layer in this investigation.

Considering 5G case with carrier/cell and assuming a slot size of 2 OFDM symbol, 275 resource blocks with 12 subcarriers each OFDM symbol contains 275 * 12 sub-symbols. With 4 MIMO layers, the PHY can process $2\ \cdot\ 275 \cdot 12 \cdot 8 \cdot 4 = 211200\ bits$ within 8 μs processing time (assuming a 256-QAM modulation to get 8 bits per symbol). There could be multiple parallel carriers ($n \cdot 211200$ bits) limited by spectrum availability.

6.1.1 Single RAN and co-located CN and application server

Considering the single RAN case with co-located and integrated CN and application server, the transport delay in the network is eliminated as shown in Figure 6-2. Assuming that the PHY layer produces 26400 bytes within 8 μs and further assuming that each layer of radio stack above PHY, namely MAC, RLC, PDCP, SDAP and IP can process the same amount of bytes with the same latency, then the radio stack delay would be 5 · 8 μs in UE and RAN. Even without any transport between the RAN and CN an UPF is needed to encapsulate the mobility tunnelling (i.e., GTP-U in 5G). UPF one-way processing latency can be estimated from the state-of-the-art, it corresponds UPF to α 40 μs, where is state of the art delay 40 μs in 5G UPF [Int20]. The E2E delay would thus be *UE radio stack delay* + *RAN stack delay* + *UPF delay* = $40\, \mu s + 40\, \mu s +\ \alpha\, 40\, \mu s = 84\, \mu s$ when using same α=0.1 as for the radio stack processing evolution in previous section. Additionally, there is one extra delay (A μs) covering the leg between CN and Server. For co-located server, ‘A‘ would be relatively small compared to the cases where server is deployed externally.

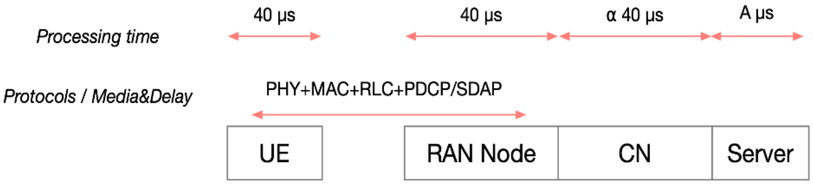


Figure 6-2: Latency components of single RAN with co-located CN.

6.1.2 Split RAN with remote server external to edge

For the RAN functional distribution, we apply a 5G variant of functionality split, as for 6G there does not exist any yet. Starting from the UE and proceed towards the application server beyond the CN (see Figure 6-3), the E2E path consists of a UE including the protocol stack (PHY+MAC+RLC) and the radio layer processing and signal propagation over the air. Thereafter we have distributed RAN functionalities RU, DU and CU that implement the RF, PHY, MAC, RLC layers, PDCP and SDAP processing, fronthaul and backhaul transport latencies (Eth Switch), CN UP element processing (UPF), datacentre switch(es) and the application server (Server) IP-stack.

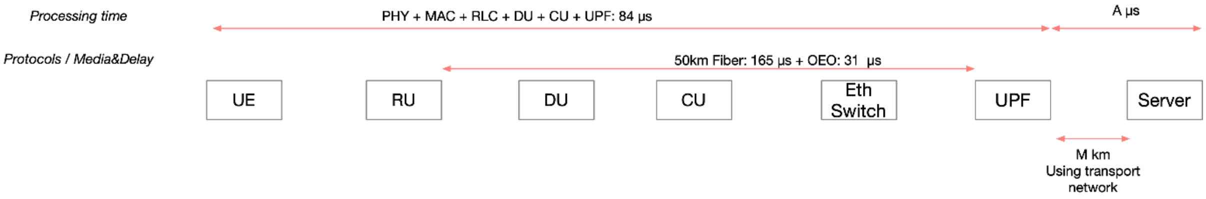


Figure 6-3: Latency components of split RAN.

The distance between the RU and the application server we choose to be a typical 50 (+ distance between the UPF and application server M km, see Figure 6-3). For cable fabric latency we use values of hollow core optical fibres which is ~50 km* 3.3 μs/km = 165 µs latency for 50 km. If the application server is further away, the delay will increase by 3.3 μs/km. The cable fabric is segmented between the distributed RAN elements (RU-DU-CU) and UPF. For opto-electronics (OEO) latency we use average value 7.5 μs [Inf20] and for ether switching 1 µs. The number of components depend on how the RAN is split.

Summing the known latency contributions (radio stack and UPF delays of 84 µs + fibre delay of 165 µs + OEO + ethernet 31µs) up we face 280 μs latency (excluding the delay from an application server A µs and the server side IP-stack processing time). This leaves 720 μs for transport of the packet between the server and UPF and to application server internal processing to stay within the given latency budget of 1 ms.

6.2 >100 bn connected devices

This section describes how to reach the target of 100 billion devices. We have divided this into two methods. The first method is to analyse the connection density results from 3GPP and International Telecommunication Union (ITU) evaluations for NR which more or less only concerns the radio (air interface) capacity. The second method is to discuss how our enablers can contribute to the overall improvement of the E2E capacity for the required connection density.

6.2.1 Air-interface connection density for NR

One method to estimate the number of connections a mobile system can handle is defined in chapter 7.1.3 in [2412-0]. For 5G this method was used to estimate the number of connected devices, i.e., the connection density [37.910]. The connection density in [2412-0] is the total number of devices fulfilling a specific QoS per unit area (per km^2). The QoS is fulfilled if all users have a 99th percentile packet delay that is less than or equal to 10 seconds. This is evaluated by using system simulations and link simulations.

The outputs of the simulations in 3GPP [37.910] are the number of users N supported per transmission point (TRxP) and the basic equation that is used is the resources needed (in average) to support the traffic (W_{user}). When a full buffer simulation is used, there is a need to recalculate the needed resources (or bandwidth) as if there was a packet transmission, i.e., scaling to get the average number of resources (bandwidth) required B_i. This is made by the following equation:

$$B_i = \frac{T}{R_i/W_{user}}, \qquad (6\text{-}1)$$

where $T = PacketSize/T_{\text{inter-arrival}}$., R_i the achievable data rate from the simulations, and W_{user} is the number of users supported. In [37.910] they assume an inter-arrival rate of 1 packet per 2 hour per user and a packet size of 32 bytes.

The connection density is calculated as follows:

$$C = \frac{N}{A} = \frac{N_{mux} \cdot W / mean(B_i)}{ISD^2 \cdot \sqrt{3}/6} \quad , \tag{6-2}$$

Where N_{mux} is the number of users multiplexed on same time and frequency resource (e.g., using Multi-User MIMO, MU-MIMO), W the total bandwidth used, and the B_i is the bandwidth required for the used traffic model and ISD is the inter-site-distance between the cells in a hexagon deployment. The term $W/mean(B_i)$ is giving the number of users that a TRxP can accommodate.

Table 6-1 gives the connection density for NR and LTE for a few selected cases (see [37.910] for a complete set of cases and parameters used). As can be seen the connection density is a rather large number per km^2, more than one million connections per square kilometres can be accommodated for only 180 kHz bandwidth. One reason for this is of course the large interarrival time for the packets, the small size of the packet and the low carrier frequency, but also from an efficient radio interface.

Table 6-1 Connection density for three case [37.910].

Case:	Connection density [Millions per km^2]	Bandwidth (W)	Cell radius	Frequency
NR_FB_500m	36.008	180 kHz	500 m	700 MHz
NR_FB_1732m	1.5034	180 kHz	1732 m	700 MHz
NB-IoT-RRCresume	1.225	180 kHz	500 m	700 MHz

6.2.2 Verifying connection density for NR

To estimate if the connection density can handle more than the target of 100 billion connections, we use two cities with high population density, in this case Paris and Athens. We then compare this with the worst cases in Table 6-1. To get the corresponding city target connection density, we scale the city population with earth population (8 billion) and multiply this with 100 billion connections. Table 6-2 shows that the maximum number of connections achieved in [37.910] exceeds the target connections with 4-5 times.

Table 6-2 Comparing city connection density assuming 100 billion worth wide connections to the estimated maximum connection density.

				Maximum connections [millions per city area] for case:	
City	Population [millions]	Area [km^2]	Target connection density [millions]	NR_FB_1732m	NB-IoT-RRCresume
Paris	2.16	105.4	27.07	158.4	129.1
Athens	0.74	38.96	9.3	58.6	47.7

Since the connection density depends on many of the assumptions, another way to investigate the connection density is to say that it should improve with 6G over 5G. Assuming that the same traffic model and the same QoS requirement is used, what can be done to increase connection density? Parameters that increase the connection density C in eq. (6-2) are:

- Increased Signal to Interference plus Noise Ratio (SINR)
- Increased bandwidth W
- Increased multiplexing

- Decreased inter-site distance.

On the other hand, we can expect an increase in the traffic demand for 6G. For 6G, there will probably be more available bandwidth [HEX-D13], albeit in higher frequency bands. To reach the same QoS as in [37.910] the cell size needs to be reduced. This also increases the connection density (but also the infrastructure cost).

The connection density is not only limited by the radio interface, but also the total E2E capacity. In Hexa-X we develop several enablers that may improve the total E2E capacity of number of connections. These are:

- Improvement of the signalling efficiency (procedures)
- Virtualization and Service based type architecture allows more reuse of functions
- Independent NFs (separation of concerns)

6.3 Full coverage (100%) of world area

The objective with this target is to estimate the global coverage. Note that there is no required minimum data rate here. However, for a meaningful estimation we still assume that 1 Mbps is the minimum wanted data rate per user. We also assume a very low density scenario, as used in Section 4.3.1. To estimate this, we are using the NTN global coverage simulation results from Section 4.3.1. Note that we assume same parameters as in Section 4.3.1, e.g., the scenario is still that the devices are handheld (with 0 dB antenna gain). The assumptions are the following:

- Assume a certain cell area for the LEO satellite
- Vary the number of satellites
- Assume each satellite is equipped with beam forming and the gateway is a dish antenna with a certain antenna gain (see Table 4-2)
- Assume the satellites can relay the data (inter-satellite links)

Thereafter, the methodology involves the following steps:

- Calculate the SINR per cell area for LEO satellites assuming no interference
- Calculate the number of ISL hops and the ISL delay for each cell
- Estimate the cell bitrate from the SINR
- Assign a simple TCP model based on RTT (where the ISL delay is one part) to achieve a more realistic cell throughput
- Assume the feeder and ISL links are not limiting the capacity of the service links

The results from Figure 4-9 shows that around 600 satellites are needed to at least cover the area and Figure 4-10 shows that more than 600 satellites are needed to serve the users with at least 1 Mbps. Note that in this simulation, the number of users over the area had a very low density. Further on, the results here, as described in Section 4.3.1, depend to a large extend on the simulation parameters used, such as the satellite altitude, antenna gain, transmit power, bandwidths etc.

6.4 (>99%) of global population reached with (>1 Mbit/s) data rates

The objective with this target is to estimate the global coverage with a minimum data rate of 1 Mbit/s for 99% of the population. To show this, we perform a satellite capacity estimation for a rural area with low population density. The goal here is not to show that the satellite system can handle *all* traffic. Instead, the goal is to show that NTN can support rural areas (i.e., areas not easily covered by terrestrial network) with low traffic density and enabling 99% of the global population with 1 Mbps traffic service. Note that we assume same parameters as in Section 4.3.1, e.g., the scenario is still that the devices are handheld (with 0 dB antenna gain and 33 dBm transmit power).

From the NTN simulations in Section 4.3.1.2, the throughput is summed for the whole satellite for all beams (using non-TCP throughput). The number of satellites in orbit is thereafter varied, from 500 to 7000. The area to be covered for each satellite is then the earth area divided by the number of satellites in orbit.

The satellite area is populated with the population density from two rural areas in Sweden with low population density, namely Kiruna and Jokkmokk (with roughly 1.2 and 0.25 persons per square km^2, respectively). With few satellites in orbit, the area to cover per satellite becomes large and the number of users to service for each satellite is also rather large. Figure 6-4 left shows the throughput per user for different number of satellites. As said above, two different population densities are used (Kiruna and Jokkmokk). The Figure 6-4 left shows that is feasible to support 1 Mbps for the Jokkmokk density in downlink (DL) when the number of satellites exceeds 3000. However, the uplink (UL) remains challenging and for the Jokkmokk density, around 14000 satellites are needed. Figure 6-4 left shows the same calculations but now with the traffic per area. The traffic area throughput is similar to the results achieved in [FJT+20] for the hand-held scenario.

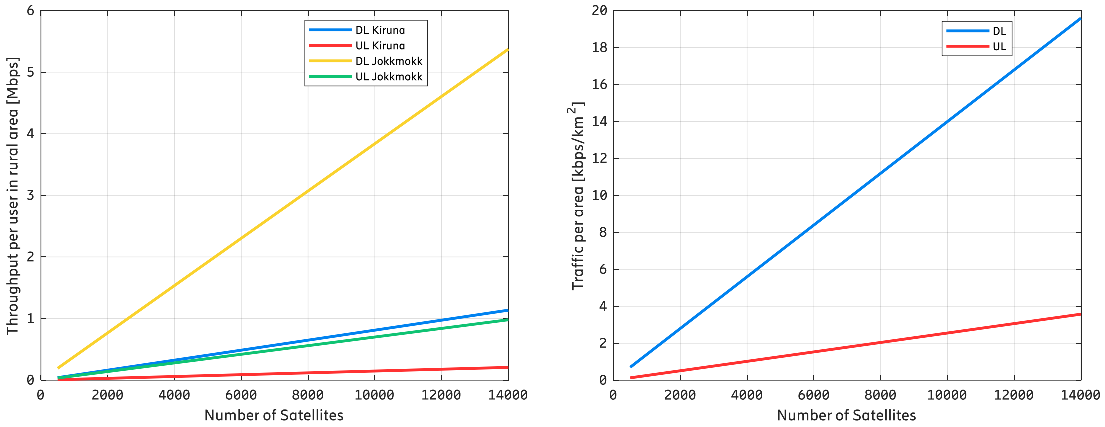


Figure 6-4 Estimate of the downlink throughput per active user for two rural areas in Sweden (Kiruna and Jokkmokk, left figure) and the downlink traffic per area (right).

The conclusion is that it is feasible to support very low density areas with 1 Mbps/users assuming there is a terrestrial network for areas with higher population density. This means that we can with high probability support 99% of the population with at least 1 Mbps, assuming both NTN (for rural areas) and TN (for all other areas). Note that the results here, as in Section 4.3.1, depend to a large extent on the simulation parameters used, such as the satellite altitude, antenna gain, transmit power, bandwidths etc.

Another possibility to using NTN is to use tower based base stations, which can cover wide areas and therefore can sparsely deployed, i.e., a Sparse Terrestrial Networks (STN) using high towers and large antenna arrays. In [FJT+20], it is concluded that STN has been showed to be equal or better than an NTN network in terms of user throughput.

7 Conclusions

The **main objectives of WP5** are to develop architectural components for 6G and to enable:

- *Intelligent network* to support full **AI integration** and support of network **programmability** (WP5.2),
- *Flexible network* design including ad hoc networks and global coverage (WP5.3),
- *Efficient network* for a **streamlined architecture** for a network of networks (WP5.4).

To enable an efficient introduction of the new components (or enablers) in the 6G architecture, the first deliverable [HEX-D51] proposed several architecture principles. These principles stipulated for example self-sustained functions, network scalability, efficient exposure of network capabilities, automation, flexibility to new deployments and simplification of the architecture. Further, these principles have guided the development of our enablers during the project life-time. In [HEX-D52] the initial enablers and concepts were developed for the main WP5 objectives, including some evaluations of the concepts. [HEX-D52] also initiated the work on several so-called frameworks, i.e., a collection of several technical components. This deliverable continues developing the various frameworks and enablers, providing more details and evaluations.

For WP5.2 (Intelligent network), an AIaaS framework with required services and functions are developed, together with the analytics (data collection) framework needed. A complete programmability framework is also developed. The framework enables the network to reprogram certain functionality over all nodes and functions in the network (UE, RAN CN etc), controlled via the management and orchestration. The deliverable also includes evaluations of the concepts and frameworks, for example AI to assist control of a remote robot, evaluation of efficiency of FLaaS in a system simulation AI and an application on how to utilize UE programmability. Security and trust issues regarding AI are addressed with a framework that supports a secure exchange of AI related information.

The flexible network design (Flexible network, WP5.3) includes a new framework for mesh ad hoc device networks to enable increased coverage and capacity on a demand basis. The ad hoc network is created and controlled by a management network that gives a detailed control of the mesh network. The ad hoc mesh network is evaluated to illustrate the ability of the ad hoc network to optimize cost, throughput energy etc. Global service coverage is shown to be possible assuming an NTN architecture that allows inter-satellite-link (ISL) hops.

To enable a more streamlined architecture (Efficient network, WP5.4), a possible 6G SBA architecture with fewer interfaces and processing points are evaluated in terms of latency for a handover procedure. Another important aspect of efficient networks is the total cost of ownership (TCO). In this deliverable, a method is developed on how to perform a qualitative TCO analysis for some exemplary Hexa-X use cases. Further, a Compute as a service (CaaS) framework is proposed that allow delegating/offloading generic application-related workloads.

In addition to the main objectives, this document/deliverable also introduces aspects on how the components and frameworks mentioned above are meant to integrate with each other in the 6G architecture. These frameworks are for example the AIaaS, FLaaS, analytics, programmability, CaaS and ad hoc mesh networks management. The intention is to enable that these frameworks can leverage each other's services, even though we do not mandate that all of them must be deployed in a given network configuration. Therefore, an Exposure and Coordination Framework (ECF) for integrating different frameworks is proposed in the document. The ECF facilitates integration of different frameworks into a functional system to meet the needs of a specific deployment scenario, so that the individual frameworks can discover, use, and share services and resources among themselves. Two main methods to implement the ECF are identified. The first is to use current SBA to enable a tightly integrated approach between the frameworks. For looser integration between the frameworks, an API management framework of [HEX-D6.2] can be applied. In the latter case, each framework is considered as its own domain interacting with other frameworks over CAPIF APIs and Data Mesh for streaming data. The ECF should contain cross framework governance and control functions.

This deliverable concludes the **quantified targets** for the Hexa-x "Network evolution and expansion towards 6G" objective, see also [HEX-D73]. For the "full (100%) global service coverage" target, the conclusion is that it is feasible to support assuming a LEO constellation that allows efficient inter-satellite-link hops (in order to achieve coverage over ocean areas) with at least 600 satellites. Further, for the target of "99% of global population reached with more than 1 Mbps", the investigation in this document shows that it is possible to serve very low population density areas (where terrestrial networks are not the main viable solution) with 1 Mbps/users assuming at least 14000 satellites in orbit. The underlaying assumption here is that there is a terrestrial network for areas with higher population density. Note that the results depend to a large extent on the simulation parameters used, such as the antenna gain, transmit power, bandwidths, etc. The target "Simultaneous high data rate and low E2E latency" is estimated for a cloud RAN/CN scenario with a lower layer split for the radio unit. Assuming fibre and a server not further away than 50 km, it is possible to achieve a user plane latency lower than 1 ms for high data rates.

7.1 More detailed conclusions

For the **Intelligent network,** an **AIaaS framework** that provides the core functionality for applying AI/ML across the cloud continuum is developed. The framework defines the common services and functions for consumption of an in-network AI. These services and functions can manage and train AI/ML models as well as deploy and monitor the accuracy and impact of the decision of the AI agents in a consistent manner. Distributed AI/ML techniques come with a wide range of advantages when facing the ecosystem of 6G networks, however it is important to consider that they also come with various challenges: (i) unbalanced data size, (ii) communication constraints, (iii) Privacy & security requirements and regulation. Therefore, a trust framework that supports a secure (trusted) exchange of AI related information is developed. The main advantage here, besides security and privacy, is that the "AI communication and computing overhead" architecture KPI can be positively impacted, as the scope of operation of a given AI agent can be extended to different privacy domains. Further, the proposed trust framework reflects the main EU AI Regulation requirements.

Within the Intelligent network, there has also been work in how to evaluate and implement the different intelligent network frameworks. For this reason, a proof of concept (PoC) for the FLaaS is developed, called "Federated Learning of eXplainable AI models (FED-XAI)". The PoC shows the benefits with FL for a Tele-operated Driving (ToD) use case and shows a real-time video streaming using commercial video server/player. It includes an offline training of a global FED-XAI model and online forecasting of the video quality implemented using Intel OpenFL (following the FLaaS framework). The network scenarios are configured according to data from TIM's live RAN.

To increase the reliability of the network at application level, a predictive control loop using ML is developed. The control loop infers delayed or lost commands and feeds them to the robot control loop. Performance has been assessed under simulated environment with wireless interference and shows that the solution provides higher precision by reducing the trajectory error.

A framework for supporting **programmability** in the network's infrastructure are developed. The framework enables the flexible reconfiguration of the behaviour of this infrastructure over time. A framework for local, domain-specific programmability managers has been defined for different network domains to discover programming capabilities of underlying infrastructure.

Flexible network aim is to enable extreme performance, scalability, and global service coverage. This can be achieved by developing solutions that can both incorporate different (sub)network solutions that can easily adapt to new topologies and spectrum as well as different traffic demands in a flexible way.

In order to overcome challenges imposed by static infrastructure solutions, a flexible D2D mesh ad hoc topology of access points, e.g., with the aid of unmanned aerial nodes, can be used. A proper selection of these nodes is achieved, in order to account not only for the maximization of the system's trust but also for the minimization of the deployment cost.

The deliverable also develops and evaluates different NTN architectures. Global service coverage is possible assuming an architecture that allows inter-satellite-link (ISL) hops. To achieve 100% availability, more than 600 satellites (with ISL) in LEO are needed. For a very low population density, the 600 satellites are able to serve roughly 95% of the users with more than 1 Mbps. The simulation results shows that device throughput then depends on the number of satellites. With more available satellites per UE there is also an increase in available resources per UE. The deliverable also investigates a 3D scenario where the devices first connect to an UAV, which in turn connects to a fast moving satellite or HAP (in low orbit). This can provide a more robust mobility solution since devices can connect via an almost stationary UAV unit. Another important topic for a flexible network is the ability of the network to utilize the available spectrum and achieve a reliable connection. This is achieved with a new 6G multi-connectivity solution for 6G, which combines the best features from CA and DC.

The 6G architecture should be **streamlined** and enable an **Efficient network**. With this we mean that 6G should be more efficient in terms of (signalling) overhead, scalability, flexibility as well as resource and power consumption compared to previous generations. The efficient network rely on the following principles:

- Exposed interfaces are service based and designed for cloud use.
- Network functions are designed with minimal dependencies to simplify interaction among services.
- Network simplification compared to previous generations.

In this final deliverable some of the principles are revisited to further demonstrate how the principles help in the process of designing independent and self-sustained network functions. With this in mind, a possible 6G SBA architecture with fewer interfaces and processing points is evaluated in terms of latency for a handover procedure. The results show that the latency of the control signalling of the handover procedure may be reduced. One aspect of an efficient network is the ability to dynamically deploy functions over the network, depending on the wanted performance. In this deliverable, we develop a concept on how to optimize placement of NFs for latency for an NTN scenario. Latency aware NF function placement can reduce the control plane latency introduced in satellite backhaul and fronthaul scenarios for Edge computing.

Another important aspect of efficient networks is the total cost of ownership. In this deliverable, a method is developed for how to perform a qualitative TCO analysis for some exemplary Hexa-X use cases. The network’s cost structure in terms of RAN infrastructure, energy consumption, backhaul, CN infrastructure, and other network costs (people, network management and maintenance, etc.) as well as the “weight” of each cost item have been defined based on the analysis performed by GSMA in [GSM19].

With CaaS, devices can choose to delegate resource-intensive processing tasks to other parts of the network providing more powerful compute nodes. This may both lead to a more efficient use of the computing resources but may also lead to a more complex architecture. The proposed Compute Federation architecture is a generic architecture that can be applied to offloading a workload of a device to networked compute nodes. Also, a new method is proposed where CaaS is used for device mobility where the main idea is to incorporate compute latency requirements in the handover decision, to prioritize cells with low compute latency and down-prioritize cells with high latency. Joint communication and sensing (JCAS) can potentially also improve the 6G mobility. It is a technique that relies on the radio resources initially used for communication to also position objects (within the range of the radio). A number of potential scenarios where sensing information can help improve handovers have been identified.

8 References

[23.222] 3GPP TS 23.222, “Functional architecture and information flows to support Common API Framework for 3GPP Northbound APIs; Stage 2 (Release 17)”, June 2021.

[23.288] 3GPP TS 23.288, "Architecture enhancements for 5G System (5GS) to support network data analytics services (Release 17)", v17.2.0, Sep. 2020.

[38.214] 3GPP TS 38.214 V17.4.0 (2022-12), “NR; Physical layer procedures for data (Release 17)”, December 2022.

[23.501] 3GPP TS 23.501, “System architecture for the 5G System (5GS); Stage 2 (Release 16)”, December 2021.

[23.700-27] 3GPP TR 23.700-27, “Study on 5G System with Satellite Backhaul; (Release 18), December 2022.

[23.737] 3GPP TR 23.737, “Study on architecture aspects for using satellite access in 5G; (Release 17)”, March 2021.

[2412-0] ITU-R M.2412-0, “Guidelines for evaluation of radio interface technologies for IMT-2020”, Oct. 2017.

[29.520] 3GPP TS 29.520 Network Data Analytics Services; Stage 3, (Release 15)”, April 2019.

[29.522] 5G System; Network Exposure Function Northbound APIs; Stage 3 (Release 15), April 2019.

[33.501] 3GPP TS 33.501 Security architecture and procedures for 5G system (Release 17).

[37.910] 3GPP TR 37.910, “Technical Specification Group Radio Access Network; Study on self-evaluation towards IMT-2020 submission”, v17.0.0, March 2022.

[38.801] 3GPP TR 38.801, “Study on new radio access technology: Radio access architecture and interfaces (Release 14)”, March 2017.

[38.811] 3GPP TR 38.811, “Study on New Radio (NR) to support non-terrestrial networks”; (Release 15) September 2020.

[38.901] 3GPP TR 38.901, “Study on channel model for frequencies from 0.5 to 100 GHz (Release 16)”, v16.1.0, January 2020.

[303681-1] ETSI, “EN 303 681-1 V1.1.2, Reconfigurable Radio Systems (RRS); Radio Equipment (RE) information models and protocols for generalized software reconfiguration architecture; Part 1: generalized Multiradio Interface (gMURI),” June 2020.

[103850] ETSI, “ETSI TS 103 850 V1.1.1, Reconfigurable Radio Systems (RRS); Definition of Radio Application Package”, October 2022.

[3GPP18] 3rd generation Partnership Project:, “User Equipment (UE) Radio Access Capabilities,” 3GPP, Technical Report (TR) 38.306, Sep. 2018, version 15.2.0.

[3GPP-22] 3GPP, “Specifications & Technologies / Release / Release 18”, [Online] Available at https://www.3gpp.org/specifications-technologies/releases/release-18

[5GVIN-D31] 5G VINNI D3.1: Specification of services delivered by each of the 5G-VINNI facilities. [Online] Available at: https://zenodo.org/record/3345612 [Accessed 27 December 2022]. June 2019.

[ACG+16] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi (eds.), Proceedings of the 2016 ACM SIGSAC Conference on Computer and

Communications Security, Vienna, Austria, October 24-28, 2016, pp. 308–318. ACM, 2016

[AIAct+21] Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS, April 2021

[AM18] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey". *Ieee Access*, vol. 6, p. 14410-14430, 2018.

[ASB+22] S. Tadesse Arzo, D. Scotece, R. Bassoli, D. Barattini, F. Granelli, L. Foschini, F. H. P. Fitzek, MSN: A Playground Framework for Design and Evaluation of MicroServices-Based sdN Controller, Journal of Network and Systems Management (2022) 30:19, https://doi.org/10.1007/s10922-021-09631-7.

[Bia00] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," IEEE Journal on selected areas in communications, vol. 18, no. 3, pp. 535–547, 2000.

[BAK+21] Bayazeed, Adnan, Khaldoun Khorzom, and Mohamad Aljnidi. "A survey of self-coordination in self-organizing network." *Computer Networks* 196 (2021): 108222.

[BCC+20] F. Babich, M. Comisso, A. Cuttin, M. Marchese, and F. Patrone, "Nanosatellite-5G Integration in the Millimeter Wave Domain: A Full Top-Down Approach," IEEE Transactions on Mobile Computing, vol. 19, no. 2, pp. 390–404, 2020.

[BIK+17] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning." Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2017.

[BLB+20] P. Bosch, S. Latre, and C. Blondia, "An analytical model for IEEE ´ 802.11 with non-IEEE 802.11 interfering source," Computer Networks, vol. 172, p. 107154, 2020.

[CA16] M.E. Celebi and K. Aydin, "*Unsupervised learning algorithms*." Springer International Publishing, 2016.

[CCM21] M. Corici, P. Chakraborty, T. Magedanz, Fraunhofer FOKUS Institute, "A Study of 5G Edge-Central Core Network Split Options", Network 2021, 1(3), 354-368;

[CHA23] Archit Chauhan (2023) Microservice Architecture vs Service Based Architecture. https://medium.com/@architchauhan/microservice-architecture-vs-service-based-architecture-278ccbec32ba

[CSC+12] S. Chitta, I. Sucan, and S. Cousins, "Moveit![ros topics]," IEEE Robotics & Automation Magazine, vol. 19, no. 1, pp. 18–19, 2012.

[CYZ15] Chen, Ming-ming & Yang, Yan & Zhong, Zhang-dui. (2015). Location-Based Handover Decision Algorithm in LTE Networks under High-Speed Mobility Scenario. 2015. 10.1109/VTCSpring.2014.7022977.

[Deh20] Z. Dehghani, "Data Mesh Principles and Logical Architecture" https://martinfowler.com/articles/data-mesh-principles.html, December 2020.

[EBS+22] N. H. Essing, A. Bucaille, P. Sanguinho, and P. Tavares, "5G's promised land finally arrives: 5G standalone networks can transform enterprise connectivity", November 2022, [Online]. Available: Standalone 5G: Predictions for 2023 | Deloitte Insights.

[ECR+21] M. Ericson, M. Condoluci, P. Rugeland, et. al, "6G Architectural Trends and Enablers", 2021 IEEE 4th 5G World Forum (5GWF)

[FJT+20] L. Feltrin, N. Jaldén, E. Trojer, G. Wikström, “Potential for deep rural broadband coverage with terrestrial and non-1terrestrial radio networks” Front. Comms. Net., 05 July 2021, Volume 2 - 2021 https://doi.org/10.3389/frcmn.2021.691625

[GLN+14] P.L. Gorski, L. Lo Iacono, H.V. Nguyen and D.B. Torkian, ”SOA-readiness of REST.” *European Conference on Service-Oriented and Cloud Computing*, pp. 81-92, 2014.

[Glo00] P. Y. Glorennec, "Reinforcement learning: An overview." Proceedings European Symposium on Intelligent Techniques, 2000.

[GMW19] O. Goldreich, S. Micali, and A. Wigderson, “How to play any mental game, or a completeness theorem for protocols with honest majority.” Providing Sound Foundations for Cryptography, 2019.

[GSM19] Global System for Mobile Communications Association (GSMA), “5G-era Mobile Network Cost Evolution”, August 2019, [Online]. Available: https://www.gsma.com/futurenetworks/wiki/5g-era-mobile-network-cost-evolution/.

[GSR+21] J. Gallego-Madrid, R. Sanchez-Iborra, P. M. Ruiz, and A. F. Skarmeta, ‘Machine learning-based zero-touch network and service management: A survey’, Digital Communications and Networks, 2021.

[HCA+22] H. Hilary, C. Colman-Meixner, K.D.R. Assis, S. Yan and D. Simeonidou “Techno-economic analysis of 5G non-public network architectures”, IEEE Access, vol. 10, pp. 70204-70218, 2022, doi 10.1109/ACCESS.2022.3187727.

[HEXA] Hexa-X website, https://hexa-x.eu/objectives/.

[HEX-D12] Hexa-X Deliverable D1.2, “Expanded 6G vision, use cases and societal values – including aspects of sustainability, security and spectrum”, Apr. 2021, Online: Hexa-X_D1.2.pdf.

[HEX-D13] Hexa-X Deliverable D1.3, “Targets and requirements for 6G – initial E2E architecture”, Mar. 2022, Online: Hexa-X_D1.3.pdf.

[HEX-D21] Hexa-X Deliverable D2.1, “Towards Tbps Communications in 6G: Use Cases and Gap Analysis” Jun. 2021, [Online]. Available: https://hexa-x.eu/wp-content/uploads/2021/06/Hexa-X_D2.1.pdf.

[HEX-D23] Hexa-X Deliverable D2.3, “Radio models and enabling techniques towards ultra-high data rate links and capacity in 6G”, to be submitted April 30.

[HEX-D43] Hexa-X Deliverable D4.3, “AI-driven communication & computation co-design: final solutions”, Mar. 2023, Online: to be published: https://hexa-x.eu/

[HEX-D51] Hexa-X Deliverable D5.1, “Initial 6G Architectural Components and Enablers”, Dec. 2021, Online: Hexa-X_D5.1_v1.1.pdf

[HEX-D52] Hexa-X Deliverable D5.2, “Analysis of 6G architectural enablers’ applicability and initial technological solutions”, Oct. 2022, Online:Hexa-X_D5.2_v1.0.pdf

[HEX-D62] Hexa-X Deliverable D6.2, “Design of service management and orchestration functionalities”, Apr. 2022, Online: Hexa-X D6.2_v.1.1.

[HEX-D63] Hexa-X Deliverable D6.3, “Final evaluation of service management and orchestration mechanisms”, Apr. 2023, Online: to be published: https://hexa-x.eu/

[HEX-D71] Hexa-X Deliverable D7.1, “Gap analysis and technical work plan for special-purpose functionality”, Jun. 2021, Online: Hexa-X D7.1.

[HEX-D72] Hexa-X Deliverable D7.2, “Special-purpose functionalities: intermediate solutions”, Apr. 2022, Online: Hexa-X D7.2.

[HEX-D73] Hexa-X Deliverable D7.3, "Special-purpose functionalities: final solutions", May 2023, Online: to be published: https://hexa-x.eu/

[HTF09] T. Hastie, R. Tibshirani, and J. Friedman, "Overview of supervised learning". Springer, 2009.

[INFL] InfluxDB, https://www.influxdata.com/

[Int20] Intel, "Low Latency 5G UPF Using Priority Based 5G Packet Classification," White paper, Jan. 2020, [Online]. Available: https://builders.intel.com/docs/networkbuilders/low-latency-5g-upf-using-priority-based-5g-packet-classification.pdf.

[INTP4] In-band Network Telemetry (INT) Dataplane Specification Version 2.1; The P4.org Applications Working Group. Contributions from Alibaba, Arista, CableLabs, Cisco Systems, Dell, Intel, Marvell, Netronome, VMware

[IP22] Intel and Penn, "Intel and Penn Medicine Announce Results of Largest Medical Federated Learning Study." [Online]: Intel and Penn Medicine Announce Results of Largest Medical Federated... [Last Accessed: 27 of December 2022].

[JLL05] Juang, R., Lin, H., & Lin, D. (2005). An improved location-based handover algorithm for GSM systems. IEEE Wireless Communications and Networking Conference, 2005, 3, 1371-1376 Vol. 3.

[JLW+22] Chen, Ming-ming & Yang, Yan & Zhong, Zhang-dui. (2015). Location-Based Handover Decision Algorithm in LTE Networks under High-Speed Mobility Scenario. 2015. 10.1109/VTCSpring.2014.7022977.

[KMA+21] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings et al., "Advances and open problems in federated learning". Foundations and Trends in Machine Learning, vol. 14, no. 1–2, pp. 1–210, 2021.

[KUB] KubeFlow, https://www.kubeflow.org/

[Kuk22a] S. Kukliński, In-Slice Management Decomposition and Implementation Issues, IEEE Future Networks World Forum 2022 (IEEE FNWF 22), Montreal, 12-14 October 2022.

[LCC19] L. M. P. Larsen, A. Checko and H. L. Christiansen, "A Survey of the Functional Splits Proposed for 5G Mobile Crosshaul Networks," in IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 146-172, Firstquarter 2019, doi: 10.1109/COMST.2018.2868805.

[MEC003] ETSI GS MEC 003: "Multi-access Edge Computing (MEC); Framework and Reference Architecture", March 2022.

[MER+177] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data". *Artificial intelligence and statistics*, pp. 1273-1282, PMLR, 2017.

[MIN] MinIO, https://min.io/

[MRT+18] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in 6th International Conference on Learning Representations, ICLR, 2018.

[MSM97] M. Mathis, J. Semke, J. Mahdavi "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm", Computer Communication Review, a publication of ACM SIGCOMM, volume 27, number 3, July 1997

[EBS+22] N. H. Essing, A. Bucaille, P. Sanguinho, and P. Tavares, "5G's promised land finally arrives: 5G standalone networks can transform enterprise connectivity", November

2022, [Online]. Available: Standalone 5G: Predictions for 2023 | Deloitte Insights.[NSH19] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning". IEEE symposium on security and privacy (SP), pp. 739–753, 2019.

[NSS+20] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis "Simu5G – An OMNeT++ library for end-to-end performance evaluation of 5G networks", IEEE Access, vol. 8 pp 181176-181191, 2020, DOI: 10.1109/ACCESS.2020.3028550.

[OPENFL] Intel OpenFL: https://openfl.readthedocs.io/en/latest/index.html

[O21] μONOS: the next-generation architecture for the Open Network Operating System Controller. 2021. https:// docs. onosp roject. Org/.

[O23] Open Source MANO, https://osm.etsi.org/, 2023.

[OSF19] T. Orekondy, B. Schiele, and M. Fritz, "Prediction poisoning: Towards defenses against dnn model stealing attacks," arXiv preprint arXiv:1906.10908, 2019.

[SPK+20] D. Scano, F. Paolucc, K. Kondepu, A. Sgambelluri, L. Valcarenghi, P. Castoldi, F. Cugini "Augmented In-Band Telemetry to the User Equipment for Beyond 5G Converged Packet-Optical Networks". IEEE 2020 European Conference on Optical Communications (ECOC), 6-10 December 2020, DOI: 10.1109/ ECOC48923.2020.9333353.

[Pha05] P. P. Pham, "Comprehensive analysis of the IEEE 802.11," Mobile Networks and Applications, vol. 10, no. 5, pp. 691–703, 2005.

[PAE17] N. Papernot, M. Abadi, Ú. Erlingsson, I. J. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data." 5th International Conference on Learning Representations, ICLR, 2017.

[RP213565] New WI: Further NR Mobility Enhancements 3GPP RP-213565, 3GPP TSG RAN Meeting #94e, Dec. 6 – 17, 2021

[SBE+19] E. U. Soykan, Z. Bilgin, M. A. Ersoy, and E. Tomur, "Differentially private deep learning for load forecasting on smart grid." IEEE Globecom Workshops, pp. 1–6, 2019.

[Sha48] Shannon, C. E., "The Mathematical Theory of Communication" Bell System Tech. 27 (July and October 1948)

[SKK+22] E. U. Soykan, L. Karaçay, F. Karakoç and E. Tomur, "A Survey and Guideline on Privacy Enhancing Technologies for Collaborative Machine Learning". *IEEE Access*, *10*, 97495-97519, 2022.

[SRF+16] D. Sabella, D. Rapone, M. Fodrini et al. (2016). Energy Management in Mobile Networks Towards 5G. In: Shakir, M.Z., Imran, M.A., A. Qaraqe, K., Alouini, MS., V. Vasilakos, A. (eds) Energy Management in Wireless Cellular and Ad-hoc Networks. Studies in Systems, Decision and Control, vol 50. Springer, Cham. https://doi.org/10.1007/978-3-319-27568-0_17

[T23] TeraFlow, https://tfs.etsi.org/, Ferbuary 2023.

[TAZ+13] A. Tzanakaki, M. P. Anastasopoulos, G. S. Zervas, B. R. Rofoee, R. Nejabati and D. Simeonidou, "Virtualization of heterogeneous wireless-optical network and IT infrastructures in support of cloud and mobile cloud services". IEEE Communications Magazine, vol. 51, no. 8, pp. 155-161, August 2013.

[TCC+20] C. Thapa, M. A. P. Chamikara, S. Camtepe, and L. Sun, "Splitfed: When federated learning meets split learning," arXiv preprint arXiv:2004.12088, 2020

[TFS] TensorFlow Serving, https://www.tensorflow.org/tfx/serving/architecture

[TFX] TensorFlow Extended, https://www.tensorflow.org/tfx

[VTS13] S. Vitturi, F. Tramarin, and L. Seno, “Industrial wireless networks: The significance of timeliness in communication systems,” IEEE Industrial Electronics Magazine, vol. 7, no. 2, pp. 40–51, 2013.

[PKH+19] G. I. Palmer, V. A. Knight, P. R. Harper, and A. L. Hawa, “Ciw: An open-source discrete event simulation library,” Journal of Simulation, vol. 13, no. 1, pp. 68–82, 2019

[PPC+21] N. Pachler, I. Portillo, E. Crawley, B. Cameron, “An Updated Comparison of Four Low Earth Orbit Satellite Constellation Systems to Provide Global Broadband”, 2021 IEEE International Conference on Communications Workshops, June 2021

[WRS+20] C. X. Wang, M. Di Renzo, S. Stańczak, S. Wang, and E. G. Larsson, “Artificial Intelligence Enabled Wireless Networking for 5G and Beyond: Recent Advances and Future Challenges”, 2020

[Yao86] A. C.-C. Yao, “How to generate and exchange secrets,” 27th Annual Symposium on Foundations of Computer Science (sfcs 1986), pp. 162– 167, 1986.

[YHL+19] Y. Yang, X. Huang, X. Liu, H. Cheng, J. Weng, X. Luo, and V. Chang, “A comprehensive survey on secure outsourced computation and its applications.” IEEE Access, vol. 7, pp. 159 426–159 465, 2019.

[ZLL+18] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, “Federated learning with non-iid data,” arXiv preprint arXiv:1806.00582, 2018.

[ZSM-002] ETSI GS ZSM 002, “Zero-touch network and Service Management; Reference Architecture”, August 2019.

[ZXB+21] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, Y. Gao, “A survey on federated learning”. Knowledge-Based Systems, Vol. 216, 106775, 2021.

[ZSM] ETSI Zero Touch Network and Service Management (ZSM) Industry Specification Group specifications https://www.etsi.org/technologies/zero-touch-network-service-management, Accessed Feb 2023

Annex A: Additional information

A.1 Terminology

Table A-1 Terminology.

Term	Abbreviations	Term description
Service Based Architecture	SBA	A modular, cloud compatible, architecture introduced for 5G for the first time in which the CP functionality and common data repositories of a 5G network are delivered by way of a set of interconnected NFs, each with authorization to access each other's services.
Access and Mobility management Function	AMF	A CN function/node that handles authentication of user's access and mobility.
Artificial Intelligence agent	AI agent	An Artificial Intelligence agent is anything which perceives its environment, takes actions autonomously in order to given achieve goals, and may improve its performance with learning or may use of knowledge. AI agents use the trained AI/ML models (one or more) to perform the inference process (including any required data pre-processing functionality). In Hexa-X AI agents use services of AIaaS.
Artificial Intelligence as a Service	AIaaS	A concept developed in Hexa-X that consist of a set of enablers and APIs offering AI functionality to other network functions, AFs and 3rd parties. Internally it contains AI repositories, a set of AI agents for inference, AI process enforcer and AI monitoring function. See more in [HEX-D13] and [HEX-D51].
Artificial Intelligence Function	AI function	Artificial Intelligence function implements on part of an AI operation such as model creation, training, learning, inference, etc. AI agents and AIaaS implement AI functions.
Dynamic Function Placement	DFP	The act of dynamically place network functions within and across clouds. This is done by deploying intelligent algorithms to orchestrate differentiated services optimally across multiple sites and clouds, based on diverse intents and policy constraints of dynamically changing environments.
Subnetwork		An operator's network may consist of one or more subnetwork, where each subnetwork is one way to deliver services over a certain area. Subnetworks can for example be a normal macro network, pico networks using sub-terahertz spectrum (i.e., 100-300 GHz, see [HEX-D21]), mmW street micro network, high-speed railway network, Satellite network etc.
Flexibility to different topologies	Not Applicable (N/A)	The ability of the network to adapt to various scenarios subnetworks such as new non-public networks, autonomous networks, mesh networks, new spectrum, etc., without loss of performance and easy deployment. Addition of service capabilities and new services endpoints require no changes to existing E2Eeervices.
Network Function	NF	Network Function is a functional building block within a network architecture, which has well-defined external interfaces and a well-defined functional behaviour. It can be a software based or a physical function (PNF) or node. Cloud native NF is a NF that is designed to natively use services offered by a cloud execution environment (e.g., registration, discovery, etc.)
Network of networks	N/A	Defined as a network that can both incorporate different subnetwork solutions as well as a network that easily (flexibly) can adapt to new topologies (same thing as Flexibility to different topologies also)

Network Service Meshes	N/A	Network service mesh is intended to support application-to-application and function-to-function communications in 6G networks and scenarios through dynamic and automated virtual network services, to be allocated on-demand, based on application requirements.
Full Network Automation	N/A	Full Network Automation is driven by high-level policies and rules without minimal human intervention. Networks will be capable of self-configuration, self-monitoring, self-healing, and self-optimisation
Non-Terrestrial Network	NTN	Satellites and other flying objects such as HAPS and UAVs.
Programmability	N/A	UE and network programmability, a framework that gives the possibility to update the program for specific features in a network entity
Scalability	N/A	The network architecture needs to be scalable both in terms of supporting very small to very large-scale deployments, by scaling up and down network resources based on needs, e.g., varying traffic, utilizing underlying shared cloud platform
Resilience and availability	N/A	This means that the network (architecture) shall be resilient in terms of service and infrastructure provisioning using MC, and separation of CP and UP, support of local network survivability if a subnetwork loses connectivity with another network, removing single point of failures
Dependability	N/A	Dependability is the “ability to perform as and when required”. Dependability consists of the attributes: availability, reliability, safety, integrity, and maintainability. E2E dependability refers to dependability from the application perspective, encompassing multiple services (c.f. Productivity)
Reliability	N/A	Reliability is the probability to perform as required for a given time interval, under given conditions

Annex B: Architecture KPIs and enablers addressing the KPIs

In [HEX-D52] and [EWS+22] we defined a set of architecture KPIs. These KPIs are then connected with the Hexa-x architecture enablers that aim to fulfil the KPI. Table B-2 shows a summary of the architecture KPIs (explained in [HEX-D52]). The Table B-2 includes a short definition and a target value (if possible). The last column also lists some of the developed enablers that may fulfil the architecture KPIs.

Table B-2 Summary of the architecture KPIs and their targets

KPIs	Definition	Target	Enablers to fulfil KPI
Convergence time	$T_{convergence} = T_{detection} + T_{decison} + T_{reconfig} + T_{stabilization}$	Improved compared to previous generations	DFP, AI based orchestration
AI overhead	All overhead over any RAN and CN interface concerning AI compared to the case without AI	<10%	AI framework, FL framework
Network reliability	Downtime of a connection and the 5th percentile data rate of a single user in a cell. The KPI measures both the mobility within networks, between networks and the global coverage.	<0.1% RLFs, and >1 Mbit/s full global coverage	Mesh, NTN, DFP and network programmability
Separation of concerns and Ease of adding new functions in future	Number of nodes/NFs/interfaces used for a procedure or number of specifications that need to be updated. The target value for this KPI varies for different procedures, the KPI should be used to compare different solutions.	Minimize compared to previous generations	Independent NFs, Efficient signalling, programmability, cloudification of RAN and CN
TCO	6G-specific costs evaluation – in relative terms (i.e., *x*% cost savings) with respect to the baseline architecture (5G NR SA) – of the items as per the GSMA study: RAN infrastructure, backhaul, CN infrastructure, energy, and other network costs	30% reduction	It depends on the use case being considered, it could be efficient signalling or DFP.