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 D5.2
Analysis of 6G architectural enablers' applicability and initial technological solutions

Date of delivery:	31/10/2022	Version:	1.0
Start date of project:	01/01/2021	Duration:	30 months

Document properties:

Document Number:	D5.2
Document Title:	Analysis of 6G architectural enablers' applicability and initial technological solutions
Editor(s):	Mårten Ericson (EAB), Hannu Flinck (NOF), Panagiotis Vlacheas (WINGS), Stefan Wänstedt (EAB)
Authors:	Riccardo Bassoli (TUD), Mårten Ericson (EAB), Hannu Flinck (NOK), Hasanin Harkous (NOF), Bahare Masood Khorsandi (NOG), Petteri Pöyhönen (NOF), Janne Tuononen (NOF), Panagiotis Vlacheas (WINGS), Stefan Wänstedt (EAB), Merve Saimler (EBY), Mehdi S. H. Abad (EBY), Damiano Rapone (TIM), Miltiadis Filippou (INT), Markus Dominik Mueck (INT), Thomas Luetzenkirchen (INT), Giacomo Bernini (NXW), Pekka Pirinen (OUL), Giovanni Nardini (UPI), Giovanni Stea (UPI), Slawomir Kuklinski, (ORA), Ricardo Marco (ATO), Adrian Gallego (ATO)
Contractual Date of Delivery:	31/10/2022
Dissemination level:	PU[1]
Status:	Final version
Version:	1.0
File Name:	Hexa-X_D5.2_v1.0

Revision History

Revision	Date	Issued by	Description
0.1	2022-03-28	Hexa-X WP5	Draft ToC of 5.2
0.2	2022-05-01	Hexa-X WP5	Stable version after internal review
0.3	2022-05-31	Hexa-X WP5	Version sent to cross-WP review
0.4	2022-06-24	Hexa-X WP5	After Cross-WP review
0.5	2022-09-01	Hexa-X WP5	Version sent to external review
0.6	2022-09-13	Hexa-X WP5	Addressing the comments
0.7	2022-09-30	Hexa-X WP5	Send to GA
1.0	2022-10-30	Hexa-X WP5	Final version

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

PU = Public

Abstract

This is the second deliverable of Work Package 5 (WP5), “Analysis of 6G architectural enablers’ applicability and initial technological solutions”, denoted D5.2. In this deliverable, we develop enablers for Intelligent networks, with the aim of integrating AI/ML and programmability in the network. Several initial solutions are presented on how to integrate AI/ML functionality to the Hexa-X 6G architecture. Flexible networks intend to enable extreme performance, scalability, and global service coverage. This can be achieved by developing solutions that can incorporate different (sub)network solutions which easily adapt to, e.g., new topologies, different types of spectrum and different traffic demands. The 6G architecture should enable an Efficient networks, meaning that 6G should be more efficient in terms of, e.g., scalability, (signalling) overhead, and resource consumption compared to previous generations.

Keywords

6G architecture, Intelligent networks, AI, AI as a service, Programmability, Network automation, Flexible networks, Mesh networks, NTN, Efficient networks, 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 is the second deliverable of Work Package 5 (WP5), “Analysis of 6G architectural enablers’ applicability and initial technological solutions”, denoted D5.2.

The main objectives of WP5 are to develop architectural components for 6G that support full Artificial Intelligence (AI) integration (a.k.a. AI native system) and network programmability, a new flexible network design, while, at the same time streamline and redesign the architecture for a network of networks. These main objectives are addressed in this document.

We envision that a 6G architecture can be built on top of a multi-domain, multi-cloud environment, where functionalities span over heterogenous, distributed and specialised clouds. In this deliverable, we develop enablers for **Intelligent networks**, with the aim of integrating AI/ML and programmability in the network. Several initial solutions are presented on how to integrate AI/ML functionality to the Hexa-X 6G architecture. **Flexible networks** intend to enable extreme performance, scalability, and global service coverage. This can be achieved by developing solutions that can incorporate different (sub)network solutions which easily adapt to, e.g., new topologies, different types of spectrum and different traffic demands. The 6G architecture should enable an **Efficient networks,** meaning that 6G should be more efficient in terms of, e.g., scalability, (signalling) overhead, and resource consumption compared to previous generations.

Table of Contents

List of Tables

List of Figures

List of Acronyms and Abbreviations

2G	2nd Generation mobile wireless communication system
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
5G-PPP	The 5G Infrastructure Public Private Partnership
ACT	Actuator
AF	Application Function
AGV	Automated Guided Vehicles
AI	Artificial Intelligence
AIaaS	AI-as-a-Service
AI/ML	Artificial Intelligence / Machine Learning
AIOps	Artificial Intelligence Operations
AIS	AI Information Service
AM	Acknowledged Mode
AMF	Access and Mobility management Function
AnLF	Analytics Logical Function
API	Application Programming Interface
AS	Access Stratum
AS-CP	Access Stratum Control Plane
ASIC	Application-Specific Integrated Circuit
AS-UP	Access Stratum User Plane
AUSF	AUthentication Server Function
B5G	Beyond 5G
BBR	Bottleneck Bandwidth and Round-trip propagation time
BBU	BaseBand Unit
BER	Bit Error Rate
BGP	Border Gateway Protocol
BS	Base Station
CA	Carrier Aggregation
CaaS	Compute-as-a-Service
CalREN	California Research and Education Network

CAGR	Compound Annual Growth Rate
CapEx	Capital Expenditures
CFS	Customer Facing Service
CL	Control Loop
CLI	Command Line Interface
CM	Connection Management
CN	Core Network
CNF	Containerised Network Function
COTS	Commercial Off-The-Shelf
COM	Command Execution Confirmation
CP	Control Plane
CPRI	Common Public Radio Interface
CPU	Central Processing Unit
C-RAN	Centralized RAN
CU	Central Unit
CU-CP	Central Unit - Control Plane
CU-UP	Central Unit – User Plane
D2D	Device-to-Device
DARPA	Defense Advanced Research Projects Agency
DC	Dual Connectivity
DCAE	Data Collection and Analytics Engine
DCIM	Data Centre Infrastructure Management
DCON	Domain CONtroller
DE	Decision Element
DetNet	Deterministic Networking
DFP	Dynamic Function Placement
DL	Downlink
DMAN	Domain MANager
D-MIMO	Distributed MIMO
DRB	Data Radio Bearer
DRL	Deep Reinforcement Learning
DTN	Disruption Tolerant Networking
DU	Distributed Unit
E2E	End-to-End
EDA	Event Distribution Agents
EMF	ElectroMagnetic Field

eMBB	Enhanced Mobile Broadband
EN-DC	E-UTRA-NR Dual Connectivity
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
EU	European Union
EuCNC	European Conference on Networks and Communications
E-UTRA	Evolved Universal Terrestrial Radio Access
FANET	Flying Ad hoc NETwork
FCAPS	Fault, Configuration, Accounting, Performance, Security
FD	Functional Domain
FEC	Forward Error Correction
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
FR2	Frequency Range 2
FRR	Fast Retransmission and Recovery
FSP	FL Service Provider
FTP	File Transfer Protocol
GEO	Geostationary Equatorial Orbit
gMURI	generalised Multiradio Interface
GPRS	General Packet Radio Service
GPU	Graphics Processing Unit
GSA	Global mobile Suppliers Association
GSMA	Global System for Mobile Communications Association
GTP	GPRS Tunnelling Protocol
H2020	Horizon 2020
HAPS	High-Altitude Platform Station
HARQ	Hybrid Automatic Repeat reQuest
HNF	Hybrid Network Function

HO	Handover
HOL	Head-Of-Line
HTTP	Hyper Text Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICT	Information and Communication Technology
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things
IP	Internet Protocol
I-RNTI	Inactive Radio Network Temporary Identifier
ISD	Inter-Site Distance
ISL	Inter-Satellite Link
IT	Information Technology
ITS	Intelligent Transportation Systems
ITU	International Telecommunication Union
KPI	Key Performance Indicator
KVI	Key Value Indicator
LCM	Life-Cycle Management
LEO	Low Earth Orbit
LMF	Location Management Function
LSTM	Long Short-Term Memory
LTE	Long Term Evolution
M&O	Management and Orchestration
MA	Moving Average solution
MAC	Medium Access Control
MANET	Mobile Ad hoc NETwork
MAPE	Monitoring-Analysis-Planning-Execution
MC	Multi-connectivity
MCG	Master Cell Group
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

MM	Mobility Management
mMIMO	massive MIMO
mMTC	massive Machine Type Communications
MNO	Mobile Network Operator
MO	Managed Object
MTBF	Mean Time Between Failures
MTLF	Model Training Logical Function
multi-TRP	multiple Transmission and Reception Point
MU-MIMO	Multi User MIMO
NACK	Negative Acknowledgment
NAS	Non-Access Stratum
NE-DC	NR-E-UTRA Dual Connectivity
Near-RTR	Near-Real Time RIC
NEF	Network Exposure Function
NF	Network Function
NFV	Network Function Virtualization
NG-RAN	Next Generation RAN
NGEN-DC	NG-RAN EUTRA-NR Dual Connectivity
NIC	Network Interface Card
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
NLU	Natural Language Understanding
NN	Neural Network
Non-RTR	Non-Real Time RIC
NP	Nondeterministic Polynomial
NPU	Network Processing Unit
NR	New Radio
NRF	Network Repository Function
NS	Network Service
NSA	Non-Standalone (NR (5G) network)
NSM	Network Service Mesh
NSSMF	Network Slice Subnet Management Function
NTN	Non-Terrestrial Network
NWDAF	Network Data Analytics Function
OAM	Operations, Administration and Maintenance
ODA	Open Digital Architecture

ONAP	Open Network Automation Platform
OpEx	Operating Expenditures
O-RAN	Open Radio Access Network
OS	Operating System
OSPF	Open Shortest Path First
P4	Programming Protocol-independent Packet Processors
PCell	Primary Cell
PDCP	Packet Data Convergence Protocol
PDP	Policy Decision Point
PDU	Protocol Data Unit
PEP	Policy Enforcement Point
PHY	PHYsical layer
PNF	Physical Network Function
PoC	Proof-of-Concept
PRNET	Packet Radio NETwork
PSTN	Public Switched Telephone Network
QAM	Quadrature Amplitude Modulation
QoE	Quality of Experience
QoS	Quality of Service
QUIC	Quick UDP Internet Connections
RACK	Recent ACKnowledgement
RAN	Radio Access Network
rApp	Non-Real Time RAN Intelligent Controller Application
RAT	Radio Access Technology
REST	REpresentational State Transfer
RF	Radio Frequency
RFS	Resource Facing Service
RHDB	Reconfiguration History Database
RIC	RAN Intelligent Controller
RIM	Remote Interference Management
RLC	Radio Link Control
RLF	Radio Link Failure
RMSE	Root-Mean-Square Error
RNA	RAN-based Notification Area
RNIS	Radio Network Information Service
RRC	Radio Resource Control

RRH	Remote Radio Head
RRM	Radio Resource Management
RRS	Reconfigurable Radio Systems
RS	Reed-Solomon
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
SDN	Software Defined Networking
SINR	Signal to Interference plus Noise Ratio
SL	Sidelink
SLA	Service Level Agreement
SMF	Session Management Function
SMO	Service Management and Orchestration
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
uMTC	ultra-reliable Machine Type communications

UP	User Plane
UPA	User Plane Adapter
UPF	User Plane Function
URDF	Universal Robotic Description Format
URLLC	Ultra-Reliable Low-Latency Communication
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VANET	Vehicular Ad hoc NETwork
VAR	Vector AutoRegression
VIM	Virtualized Infrastructure Manager
VM	Virtual Machine
VNF	Virtualised Network Function
vRAN	Virtual RAN
WANET	Wireless Ad hoc NETwork
WLAN	Wireless Local Area Network
WMN	Wireless Mesh Network
WP	Work Package
XaaS	Anything-as-a-service
XAI	eXplainable AI
xMBB	eXtreme Mobile BroadBand

1 Introduction

Hexa-X is one of the 5G Infrastructure Public Private Partnership (5G-PPP) projects under the EU Horizon 2020 framework. It is a flagship project that develops a Beyond 5G (B5G)/6G vision and an intelligent fabric of technology enablers connecting human, physical and digital worlds.

The overall objective of Work Package 5 (WP5) is to develop architectural components for 6G that support a new flexible network design, full Artificial Intelligence (AI) integration, and network programmability, while at the same time to streamline and redesign the architecture for a network of networks.

This document is the second deliverable of WP5, D5.2. 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.

1.1 Objective of the document

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 remain to be fulfilled.

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 D5.1
WPO5.2	Develop technical enablers for **Intelligent Networks** 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.	The enablers for this are initiated and discussed in D5.1
WPO5.3	Enable extreme performance and global service coverage within **Flexible Networks**. 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.	The enablers for this are initiated and discussed in D5.1
WPO5.4	The **Efficient Networks** 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	The concept and enablers for this are initiated and discussed in D5.1

	integration and implementation and improve network energy efficiency.	

The objectives WPO5.2, WPO5.3 and WPO5.4 are addressed in this document in chapter 3, 4 and 5, respectively. As can be seen in Table 1-1, in D5.1 we initiated and started a discussion about the architectural enablers. In this D5.2 document the aim is to conceptualize and analyse our architectural enablers. In D5.3 we will aim to also give a better view on how the different enablers integrate with each other in a flexible, efficient, and secure manner.

In addition to this, we also develop a few architectural Key Performance Indicators (KPIs). These can be used to evaluate the WP5 objectives and to ensure a successful 6G architecture. In addition to the architectural KPIs, the document also introduces the so called “quantified targets” for the “Network evolution and expansion towards 6G” objective. The quantified targets are defined in the Hexa-X application [HEXA] and part of the overall project objectives.

1.2 Structure of the document

The document is structured as follows: Chapter 2 gives a short overview and introduction of the different enablers. Chapter 3 includes the enablers for intelligent networks, Chapter 4 presents the flexible network enablers, Chapter 5 presents the efficient networks. Chapter 6 expands the architecture KPIs developed in [HEX-D51], where we discuss if our architecture enablers can fulfil the KPIs. Chapter 7 shows the initial results regarding the Hexa-X quantified targets, and finally, Chapter 8 is the conclusions.

References can be found in Chapter 9 and there is also a list with common terminologies used in the documents in Annex A.1.

2 Architecture overview

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

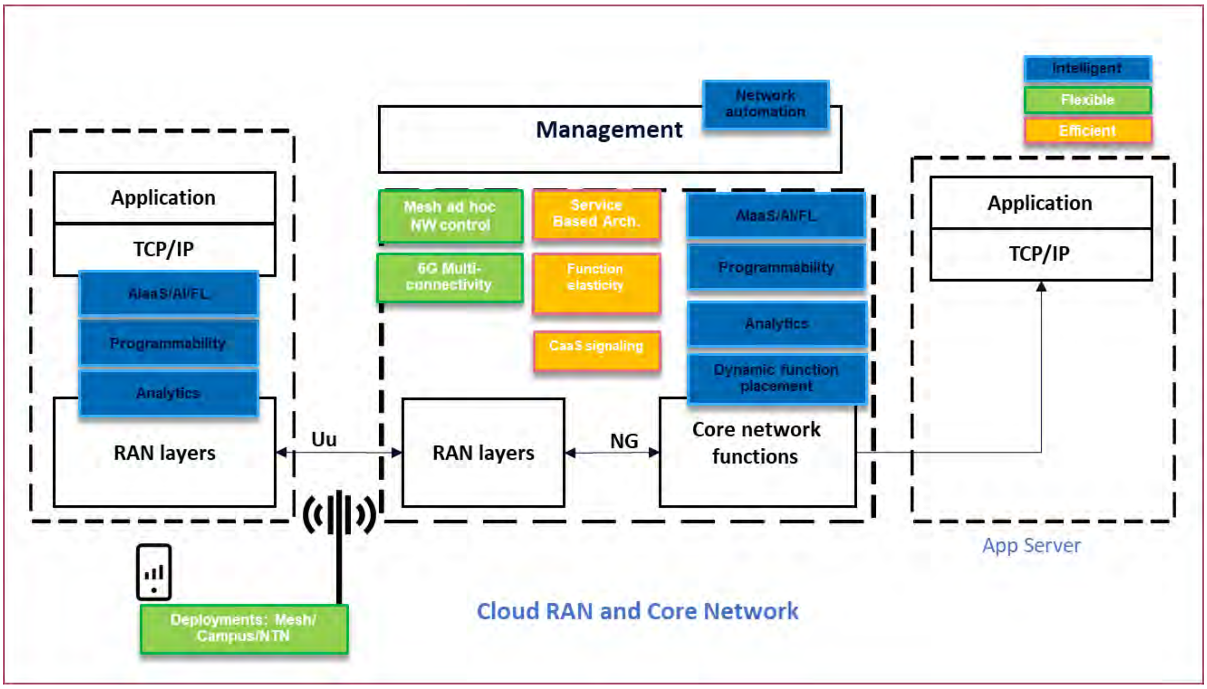


Figure 2-1 Overview of the 6G architecture enablers

The Intelligent networks (blue boxes in Figure 2-1) deals with enablers for developing a fully integrated AI and programmable network. The aim is to develop enablers for AI-as-a-Service (AIaaS) or Federated Learning as-a-service (FLaaS), using data from the analytics framework. 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 Flexible networks 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) 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 networks 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 possible to deploy Network Functions (NFs) in different cloud environments (network refactoring). This also includes a concept for Compute as a service (CaaS).

3 Intelligent networks

Intelligent Networks extend the Hexa-X End-to-End (E2E) architecture [HEX-D13] with Artificial Intelligence / Machine Learning (AI/ML) enabled closed-loop control of NFs and necessary supporting enablers to automate network operations paving the way for AIaaS that extends to AI services beyond orchestration and network management to other network functionality and services provided by the network. Specific User Plane (UP) optimizations and algorithms are described in [HEX-D42] whereas this document provides frameworks to apply AI agents and AI assisted orchestration across multi-cloud continuum.

Automation of networks is a long run objective which aims at replacing tasks undertaken by human operator with processes run by machines or pieces of software. The advancements in the application technology (e.g., computing, sensing and actuation) extend the need for automation faster than the human reaction times (e.g., Digital Twins for manufacturing, merged reality game/work, Unmanned Aerial Vehicles (UAVs), [HEX-D12], [FKS20]), leaving the role of human intervention to supervise the training of the suitable AI/ML models and handling of unexpected error cases. Even in those cases AI/ML can be used to assist and extend human decision making. Maybe the most illustrative historical example of automation in the context of telecommunication networks is the transition from manual to electromechanical (cross bar) switching in telephone networks before the emergence of fully electronic switching achieved by software. Automation of switching systems continued by adding specific controllers that analysed signalling messages with the aim to take rule-based decisions. In the context of the Internet, routing algorithms executed by routers in a distributed manner are another example of automation achieving autonomous route calculation instead of computing routes by offline algorithms and then establishing routes via manual configurations.

In this chapter, to further enhance and extend automated network operation, we present several initial solutions and frameworks to embed AI/ML functionality as an integral part of the Hexa-X 6G architecture. As stated in D5.1 [HEX-D51], we envision that 6G architecture can be built on top of a distributed multi-domain, multi-cloud environment, where functionalities span over heterogenous and specialised clouds. We detail the foreseen benefits of applying AI to network operations to deal with the increasing complexity of network configuration and to reduce time to market and cost of deploying new services (Sections 3.1.1 – 3.1.3). We have investigated a multi-agent system that can replace single 'monolithic' softwarized NFs (Section 3.1.4).

We have paid special attention to regulatory aspects of data governance and its impact to the technical requirements of the components of the proposed AI-as-a-Service (AIaaS) framework (Section 3.2.1). We propose a framework in which trust levels of multiple cross-domain AI-service consumers are managed to respond to data privacy needs within each security domain (Section 3.3.1).

We have developed an E2E analytics framework to exchange knowledge across planes and domains to support the AI agents and ML model training (Section 3.3.2). To address the management and training of AI-functionality across the cloud continuum from the central core cloud to the distributed edge and User Equipment (UE), two deployment solutions are presented: first, a fully distributed edge-based solution, where all AI-functions are instantiated at the edge nodes as cloud native applications and, second, a hybrid solution, where computationally expensive AI-functions are executed in the core cloud on behalf of the AI agents located in the edge cloud (Section 3.2.3). As 6G UEs are assumed to consume intelligent services from the network, they can also train their own AI-models collaboratively in a privacy-preserving way according to the Federated Learning (FL) paradigm [AIA21] [RGF+21]. To facilitate this, we propose a FLaaS framework and related protocols to discover and join learning federations of UEs (Sections 3.2.4 – 3.2.5).

We discuss Application Programming Interface (API) implications of the developed frameworks to the orchestrator and explain what is needed protocol wise through an example use case, where AI/ML is used to address mobility related procedures (Section 3.3).

The underlying infrastructure layer of the Hexa-X E2E architecture [HEX-D13] needs to be adaptable to varying NF workloads, new functionality, and dynamic placement of NFs across the multi-cloud continuum. New and varying workloads can be accommodated by using programmable devices. We evaluated performance and cost of Programming Protocol-independent Packet Processors (P4) programmability in multiple configurations (Section 3.4). We also propose to add programmability in the air interface of the UEs to increase 6G network adaptability to new use cases and automated UE repurposing when entering different environments. We discovered that fine-grained telemetry of the UP programmability could further improve network performance and efficiency of service delivery in terms of scalability, availability, and sustainability.

For Dynamic Function Placement (DFP), we propose a 2-level hierarchical orchestration solution, where domain internal dynamicity is not fully exposed externally but still network functions can be executed in a multi-domain multi-cloud environment on-demand basis. A top-level orchestrator decides candidate domains for NFs, to be created or to be moved. Final deployment details are left for domain specific logics (Section 3.5).

We further investigated distributed decision making on less powerful or constrained devices through use cases with AI relevant KPIs and Key Value Indicators (KVIs). This led to a proposed solution that enables distribution of AI applications that use the services offered by AIaaS, Compute-as-a-Service (CaaS), and AI Model Management as a Service in a Robotics as a Service use case (Section 3.6).

3.1 AI assisted network automation

3.1.1 Introduction

Any automation process runs a dialog between the controlled entity (or entities) and one or several controllers (e.g., the end user and the network by means of signalling). Today, controllers often embed AI to run complex tasks and to be as autonomous as possible, for instance, to implement closed Control Loops (CLs). Controllers at the same level (i.e., achieving similar tasks) or at different levels (hierarchies of controllers) can collaborate for achieving a global task [FBB+80]. The introduction of autonomous controllers collaborating to achieve complex tasks introduces additional complexity in networks, which become increasingly hard to be effectively controlled by humans.

This increase in complexity seems to be inevitable with the softwarization of networks and the emergence of Software Defined Networking (SDN), which explicitly relies on the implementation of controllers to operate the network. So far, networks have been operated to some extent by humans, even in the case of the Internet, where routers are frequently manually configured via a Command Line Interface (CLI) and routing algorithms are partially configured by humans (static routes and preferences in Border Gateway Protocol (BGP)). Roughly speaking, classical operation of networks relies on sometimes complex workflows executed by human operators, possibly triggering automatic procedures. But today, networks are evolving to become programmable platforms, in which automation plays an ever-critical role.

The expected benefits of network automation for operators can be classified into four broad categories: reduced management complexity, operation related savings, time to market, and safe operations. It is important to note that management automation is tightly linked to orchestration, which is the coordination of automated tasks to achieve an objective, e.g., the deployment of a network service involving Containerised NFs (CNFs). As tasks become increasingly complex and interdependent, AI provides efficient methods for managing complexity (including algorithmic complexity), dealing with big data, assisting operators (including customer relationships management), etc.

3.1.2 Automation of network operations and the benefit of AI

3.1.2.1 AI for complexity management and data processing

In the past few decades, networks have evolved towards higher levels of automation. However, with the emergence of virtualisation techniques, notably in the context of 5G, the automation of networks has accelerated. More and more NFs can today be virtualised and softwarized, much beyond classical Information Technology (IT) functions, giving rise to the Network Function Virtualization (NFV) framework [NFV21]. Virtualized/Containerised NFs (V/CNFs) are hosted in Virtual Machines (VMs) or containers and deployed by network orchestrators, e.g., Open Networking Automation Platform (ONAP), see [ONA] or [NEP], in relation with Virtual Infrastructure Managers (VIMs), for instance OpenStack for VMs and Kubernetes for containers.

NFV covers a wide range of network functions control functions such as those of the 5G Control Plane (CP) but also those of the data plane such as the User Plane Function (UPF) in 5G and the different components of the Radio Access Network as specified in the Open Radio Access Network (O-RAN) [ORA] architecture, i.e., Central Unit (CU), Distributed Unit (DU) and Remote Unit (RU). Almost all network functions can be virtualized, except those routing and switching functions that are executed by dedicated hardware, because of very high bit rates (more than 100 Gbit/s). While achieving unprecedented agility and customisation of NFs, this trend also introduces a great degree of complexity.

The design of a network service by means of Virtualised NFs (VNFs) can be very complex, as many solutions (in terms of software suites) exist for rendering the same service. Today, operators deploy a restricted number of software suites for a given service but, in the future, it can be envisaged that plenty of software solutions offering a large modularity thanks to microservice based design will exist on a repository to realise a service. Then, to manage this complexity, it will become essential to develop tools for selecting compatible software suites to realise a service and to manage their dependencies. The intent-based approach fits well in this context, where the operator or even a software suite can express needs in general terms (possibly in the context of Natural Language Understanding (NLU) and an AI assistant can translate them in terms of technical solutions. Beyond pure translation, this process can be complemented by ML to select the best solution by considering associated performance metrics.

The deployment of network services appearing as complex Service Function Chaining (SFC) on a shared infrastructure can turn out to be very difficult and, in general, involves Nondeterministic Polynomial (NP)-hard optimisation problems. In this context, ML methods, notably Deep Reinforcement Learning (DRL) techniques, prove to be very efficient for solving this kind of problems. For example, many studies have shown the efficiency of these techniques in the context of network slicing [BGG+20]. To perform placement of V/CNFs, measurements from the physical network are needed. This falls into the domain of network monitoring, which can be part of a safe operation of the network. Measurements from the network give rise to analytics to subsequently compute KPIs. This is for instance the task of the Data Collection and Analytics Engine (DCAE) component of ONAP. The issue related to monitoring will be further addressed in the next section.

One key innovation of 5G networks and beyond is the possibility of offering rich services, for instance slices or private mobile networks tailored to customer needs, as well as interactive services (cloud gaming, virtual reality, Metaverse, etc.). One key issue for the network is then to be able to maintain the negotiated Service Level Agreement (SLA). For this purpose, it is necessary to permanently monitor the services deployed. In classical networks, the monitoring was mostly segmented. For instance, one monitoring platform for Internet Protocol (IP) and another for cellular networks. With the deployment of E2E services spanning over several networks or network segments and the cloud infrastructure, the monitoring requires the collection of analytics coming from different networks and technologies (clouds, mobile and fixed networks). This leads to the creation of huge repository of logically centralized data (data lakes) which can be exploited for different purposes.

Beyond the quality issues and maintenance mentioned above, network analytics can be used for security issues (detection of abnormal traffic patterns, malicious usage, etc.). Data issued from the network can also be used to detect malfunctioning pieces of equipment or software. Measurement data are frequently used by network operators for troubleshooting the network by identifying abnormal patterns. Finally, network data is also used for network planning and provisioning by predicting future traffic patterns. This improves the resilience of the network.

In all these applications, ML and, more generally, all AI tools can be tailored to automate the analysis of data. Instead of requiring tedious analysis of huge amounts of data (notably times series generated by traffic measurements from the network), ML provides network operators with efficient tools for processing data and perform diagnosis in terms of quality, troubleshooting, security, etc. In addition, data are now critical for many businesses. 5G already specifies tools for exposing data to network operators (e.g., Network Data Analytics Function (NWDAF), Network Exposure Function (NEF)) and probably more such tools will be part of 6G. This is potentially a source of new revenues for network operators. Nonetheless, before being exposed, data must be processed by automated tools to be presented to potential consumers according to some formats.

3.1.2.2 AI to shorten time to market and cost reduction

The softwarisation of networks and the merge of networks and IT enable a wide variety of services, much more sophisticated than only offering connectivity via fixed and mobile networks. The services become difficult to expose and the size of service catalogues can be very large, leading to long negotiation between customers and service providers. From this observation, the TeleManagement (TM) Forum has introduced the Open Digital Architecture (ODA) neatly delineating the different responsibilities of the various actors involved in the negotiation, the selection, and quick instantiation of services [TMF21]. In particular, the Engagement Management (EM) block realises the interface between the customer and the network operator. So far, this interface was handled by humans, but with the high complexity of services possible in 5G networks and beyond, this interface could be supported by a machine via a graphical user interface or a chatbot. The advantage of this latter option is the possibility of using NLU. The user describes the desired service by means of intents which are interpreted by specialised software that translates the intents into technical terms, which are used to fill up the so-called Customer Facing Service (CFS). This high-level description of the service is then translated into technical solutions (Resource Facing Service, RFS), which are deployed by the orchestrators involved in service orchestration. The translation of a CFS into RFS by the Service Resolver can be automated with an AI assisted tool.

The automation of the service negotiation seems to be inevitable in the context of new networks, because of the wide range of services and their complexity. An example can be resource allocation in multiple domains to fulfil SLA. AI is essential in this context notably to accelerate time to market.

As illustrated in the previous sections, automation is highly required in emerging networks to manage complexity in terms of technology and services and to meet quality, security, and resilience requirements. Without automation, huge amounts of Operating Expenditures (OpEx) would be necessary to reach the promised objectives. Automation is then instrumental for operation savings.

Because processes involved in the operation of networks have become very complex, automation cannot be realised simply by automating workflows. It is necessary to introduce new learning methods to cope with complexity, to perform optimisation which cannot be solved by using classical algorithms, to better learn needs of customers, while including some network operator objectives (e.g., energy savings, better utilisation of network resources, better dimensioning and planning, etc.), etc. This complexity leads to multi-objective optimisation problems, for which AI proves very powerful. As a consequence, the automation has to greatly rely on AI.

3.1.3 AI-driven network and service orchestration

As described in [HEX-D51], AI and ML are currently considered as the key enablers to achieve full automation in 6G networks. Specifically, 6G network and service orchestration platforms can highly benefit from AI/ML to assist their Life-Cycle Management (LCM) and runtime operations at different phases, including planning, deployment, operating, scaling, and resource sharing. However, current solutions mostly rely on embedded pre-trained algorithms, statically integrated within the network and service orchestration decision logics, with the aim of supporting the LCM operations mentioned above [BMZ+20]. Furthermore, it should be noted that AI and ML solutions are not limited to network and service orchestration (which is the focus of the present section) but can be applied in other contexts including CP and UP, etc.

What is needed to achieve the level of automation and flexibility required by 6G networks and services is a more comprehensive approach, where custom sets of specialised AI functions can be deployed and re-configured on demand, to support the various network optimisation decisions to be taken at the Management and Orchestration (M&O) platforms. This approach allows to enable an agile AI-driven orchestration of 6G networks and services, with concurrent and potentially cooperating AI functions able to address the complexity of different optimisation aspects with data, models and algorithms specifically tailored to the various objectives and constraints of specific network domains, slices and slice subnets, or services. In practice, this requires a novel design approach that includes common services and AI functions for AI/ML algorithms and AI agents' LCM, together with the definition of APIs, data models and metadata for agents and algorithms capability discovery and data source requirements.

The first step is the identification of the relevant AI functions which are required to provide such a comprehensive and seamless AI-driven 6G orchestration approach, and to make them easy to manage and re-configure in a cloud-native environment. In particular, the idea is to define a set of AI functions that can be virtualised, packaged, and therefore orchestrated as AI as a Service (AIaaS) functions to be deployed, activated and re-configured on-demand to assist and complement the regular network and service orchestration logic with additional AI/ML and automation capabilities. Specifically, four main AI functions are identified as required to implement this solution: AI model repository function, AI training function, AI monitoring function and AI agent. Table 3-1 below provides a functional description for each of them.

Table 3-1 AI functions identified as enablers for AI-driven network and service orchestration

AI Function	Functional Description
AI model repository function	It is the function that provides a catalogue of the available AI/ML trained models (including their metadata for capability specification), which are either already deployed or ready to be deployed within new or existing instances of AI agents. Multiple versions of the same model can be stored in the AI model repository function, e.g., related to subsequent training sessions over different datasets.
AI training function	It is the function that performs the training of AI/ML algorithms (including any required data pre-processing) and produces executable models that can be integrated in the AI agents. The AI training function is triggered either autonomously by the AI monitoring function when a performance degradation is detected, or by the M&O layer whenever a new model has to be generated. This may happen when a new dataset is available and the existing models do not meet expected accuracy (e.g., based on pre-defined thresholds), as well as whenever new analytics is requested or a new network or a new use case has to be addressed. The AI training function takes care to store the new models in the AI repository function and can also directly take care of the deployment of the newly generated models within existing or new AI agent instances.

AI monitoring function	It is the function that takes care to evaluate the performance of the AI/ML models and consequently provide the trigger for training and re-training operations in the AI training function. This translates into evaluating the runtime accuracy of deployed models, as well as their performance in terms of action impact. This includes methods for identifying any potential conflict (direct or indirect) in the model inferences. To support a proactive approach, the AI monitoring function may monitor the data used for inference and detect potential data or concept drift that may brin to performance degradations. In addition, the AI monitoring function can also evaluate the accuracy performance of the available models in the AI repository function when a new training dataset is available.
AI agent	It is the function that uses the trained AI/ML models (one or more) to perform the inference process (including any required data pre-processing functionality). According to the specific model(s) it executes, the AI agent requires specific data to be ingested. The outputs of the AI agents are then used to drive the actions and the behaviour of other functions, including 5G NFs and Application Functions (AFs), as well as M&O functions. The AI agent may also embed part of the AI monitoring functionalities, by directly verifying model accuracy at runtime; any potential inaccuracy may trigger AI training by also sending the training dataset.

As said, the aim of having these AI functions as individual and standalone functions (e.g., implemented as edge applications) is to enable their virtualisation and packaging, preferably as cloud-native applications to then be deployed and executed on-demand at either edge cloud and/or core/central cloud locations (e.g., according to their requirements in terms of data to be ingested). Moreover, at the management system, an AI orchestration function is required at the 6G network and service orchestration level to provide the required management functionalities to deploy and configure the AI functions above in the form of AIaaS, considering their deployment constraints (e.g., edge cloud vs. core/central cloud locations). For example, such AI orchestration function can be implemented as a dedicated AI Network Slice Subnet Management Function (NSSMF) [HEX-D62], responsible to instantiate and operate the various cloud-native AI functions.

Two main deployment options are identified for these AI functions: fully distributed and hybrid. First, in a fully distributed scenario, all of the AI functions defined in Table 3-1 are deployed and executed at a single edge (and possibly instantiated at multiple edges as independent deployments) as cloud-native applications provisioned and configured as part of the network and service orchestration procedures. This allows, according to the specific capabilities of the AI/ML algorithms, to execute local automation or optimisation functionalities through the AI agents. As shown in Figure 3-1, this fully distributed scenario requires local data collection/monitoring functionalities to provide training datasets for the AI training function, as well as runtime data to be ingested in the AI agent.

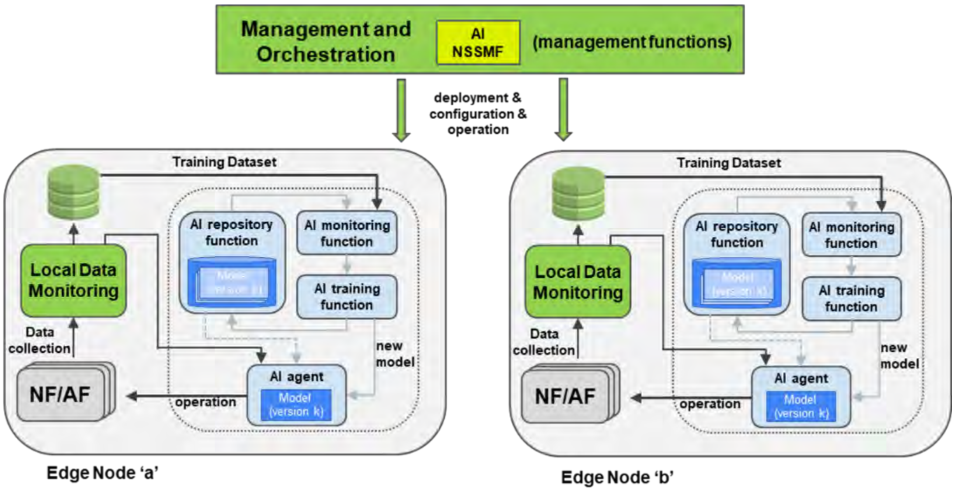


Figure 3-1: Fully distributed AI functions deployment option

In the fully distributed case, the various cloud-native AI functions are configured to implement the proper pipeline to generate and execute new versions of a specific ML model. For example, upon the availability of new training dataset, the AI monitoring function is triggered to evaluate the existing models, and in case no available trained models provide enough accuracy or performance, a new training is triggered in the AI training function, which generates a new version of the model and deploys it in a new or existing AI agent.

On the other hand, in a hybrid edge/core cloud scenario (shown in Figure 3-2), some of the AI functions could be deployed at central/core cloud locations, with the aim of not overloading the edge with computationally intensive tasks. Specifically, the AI training function could be deployed in core/cloud nodes, together with the AI repository function to concentrate all training-related operations where computing resources can be easily available with reduced cost. This hybrid edge/core cloud option requires the definition of dedicated AI functions management interfaces and workflows among the involved AI functions to perform the various operations across the edge/core cloud locations. The definition and implementation of such workflows and operations shall take into account the challenge and communication overhead associated to the sharing of (large) training datasets among the involved AI functions

The selection of which option to deploy (i.e., fully distributed at the edges vs. hybrid at edges/core cloud) can be mandated to the AI orchestration function, which can evaluate the best choice according to specific policies configured in the M&O platform, e.g., which may take different aspects into consideration (e.g., computing resources required to perform AI training operations, network resources required to transfer datasets from edge to core/cloud, real-time constraints for short-term vs. long-term closed-loop operations, etc.), as well as network topology, resource availability and constraints on the edge nodes.

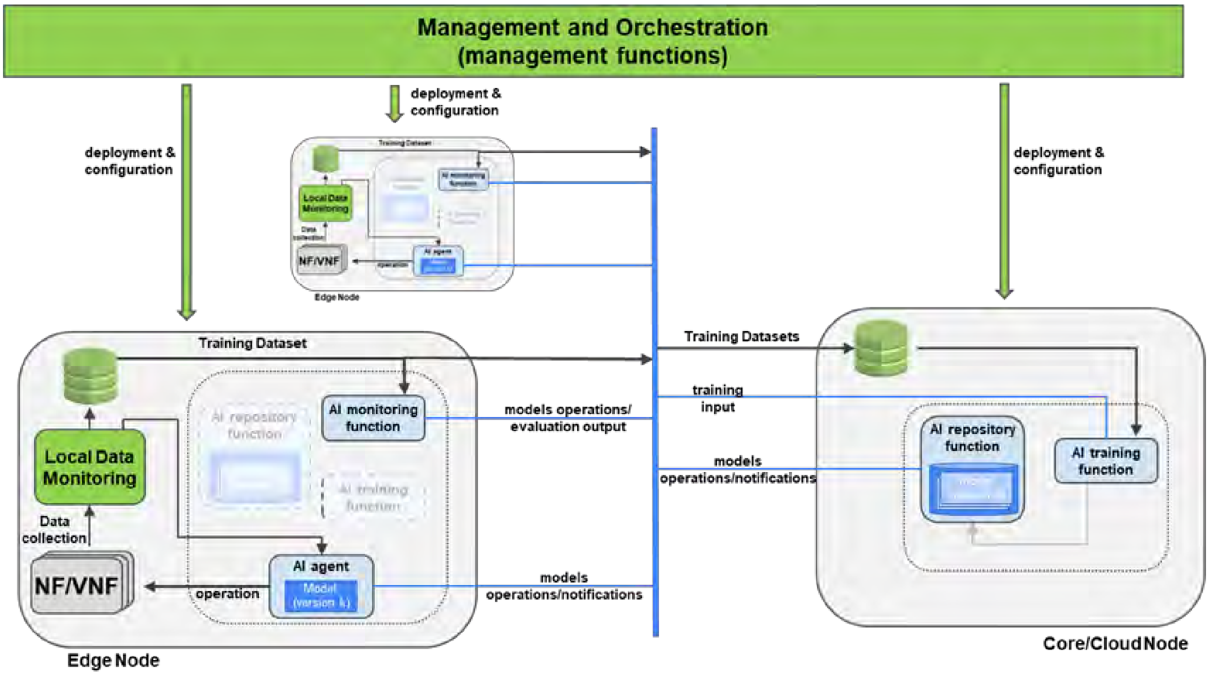


Figure 3-2: Hybrid edge/cloud AI functions deployment option

3.1.4 Multi-agent system in the continuum orchestration

The complete softwarisation of the network started with the software translation of network functionalities. Next, this abstraction has enabled the functional split of virtual network functions into smaller functionalities, which can increase the flexibility of network management and operations. With the advent of in-network intelligence, the vision that has been evolving has focused on the dynamic collaboration of intelligent virtual network functions and subfunctions. From B5G vision, any network functionality could be dynamically placed at any computing network node. By abstracting any network element or protocol in a virtual (softwarized) environment, it is possible to obtain a network that relies on Anything-as-a-service (XaaS), where 'X' is any kind of softwarized entity. By now, the approach has mainly been a one-to-one translation of network functions/protocols. This has been the original perspective of NFV.

An agent is any possible network subfunction, protocol, task, sub-task, etc., which can be performed autonomously, employing intelligence. In respect of virtual network functions and microservices, agents are sub-functions of arbitrary smaller 'size' that collaborate with each other employing intelligence. The collaboration among different agents can recreate the 'macro' network functionalities represented by virtual network function. Agents have the capability to perform decision-making individually or collaboratively. In a multi-agent system, agents are the smallest building blocks. The blocks are specialised in targeting specific goals. Using the appropriate set of autonomic agents, it is possible to build a complete multi-agent system to replace complex and distributed monolithic systems, like network management systems. The constituting autonomic agents can communicate, collaborate, cooperate, and coordinate to accomplish complex tasks. In building a multi-agent system, several challenging aspects, such as autonomy, collaboration, activity, reactivity, communicativeness, inter-agent consensus, and goal setting, have to be addressed.

Agents could be distributed to perform distributed tasks or assigned to handle different sub-functions in a localised and specialised rule. Alternatively, they could be placed in the network computing nodes of a centralised clouds to perform incoming tasks. Agents could also be homogeneous or heterogeneous. Homogeneous agents are agents with identical characteristics. They could be designed or instantiated to perform workload sharing. They could collaboratively perform large tasks that could be sub-divided into sub-tasks to be assigned to different agents. Moreover, agents could also be organised hierarchically. All agents are specialised and dedicated to performing tasks autonomously, while coordinating and communicating with other agents. However, we will limit the description to only network-level functions.

The following are the most typical network-layer functions, that could be defined as agents: Topology Management Agents, Routing Management Agents, Network Slicing Agents, Service Function Chaining Agents, Forwarding Management Agents, Quality of Service (QoS) Management Agents, Mobility Management Agents, Security Management Agents, Fault Management Agents, Resilience and Survivability Agents, Service and Application Management Agents, Monitoring Management Agents, and Generalized Control Plane Management Agents [ABG+21]. Therefore, by using the above 'atomic' units as building blocks, the objective is to realise an autonomic network management system, replacing the existing monolithic management system.

As shown in Figure 3-3, important agents are the Event Distribution Agents (EDA), that respond to incoming service requests or any network-related events. Based on the events, it reacts and dispatches the events (i.e., arrival of tasks, services, packets, network changes, etc.) to other specific agents for event handling and processing. There are also traffic prediction agents, which predict the incoming service workload. Based on the predicted amount of workload, the required number of service processing agents are instantiated, and service function chains (a sequence of service processing agents) are created at the edge data centre. In the numerical simulation results obtained using MATLAB (see the complete mathematical model in [ABG+21]), this is performed proactively, every hour. Based on the predicted workload, additional agents are instantiated if an increase in workload is predicted to come. Next, some agents can be deactivated to reduce and free edge data centre resources, if a decrease in incoming service workload is predicted. Depending on the service requirements, the service agents are sequenced as a service function chain for a given service to pass through as a part of service processing. In the evaluations, four service agents as CP and data plane functions are considered. The service function chain that is assumed consists of two serially connected agents working in parallel with another two serially connected agents, see Figure 3-3 for agents sequencing.

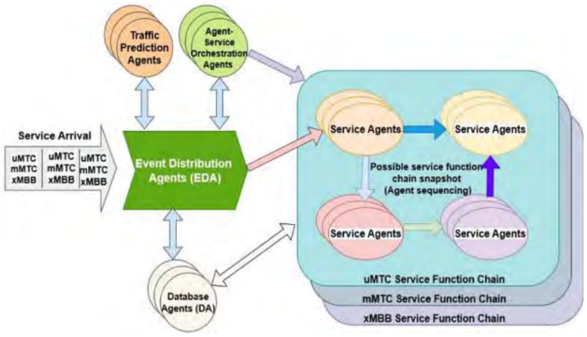

Traffic Prediction Agents
Agent-Service Orchestration Agents
Service Arrival
uMTC mMTC xMBB
uMTC mMTC xMBB
uMTC mMTC xMBB
Event Distribution Agents (EDA)
Database Agents (DA)
Service Agents
Service Agents
Possible service function chain snapshot (Agent sequencing)
Service Agents
Service Agents
uMTC Service Function Chain
mMTC Service Function Chain
xMBB Service Function Chain

Figure 3-3: Agent sequencing and scheduling for arrival service processing.

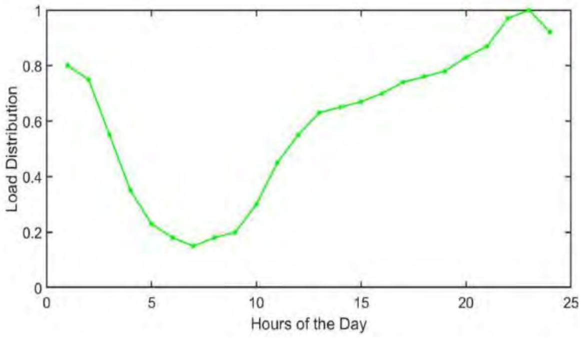


Figure 3-4: Workload distribution during the day.

The network topology considered in the simulation resembles the California Research and Education Network (CalREN), to be used as a backhaul or metro network in the evaluation. The aggregate traffic considered contains multiple types of services such as ultra-reliable Machine Type Communications (uMTC), massive Machine Type Communications (mMTC), and Extreme Mobile BroadBand (xMBB).

The service schedule is based on service priority. It is in the order of uMTC (highest priority) –> mMTC –> xMBB (least priority) in a given service scheduling period. Based on the amount of arriving service workload, resources are allocated to agents before they are instantiated. Based on the total serving capability of agents, a service is injected using the workload percentage shown in Figure 3-4. The evaluation also considers three types of services with a different workload demand to the autonomic network management functions.

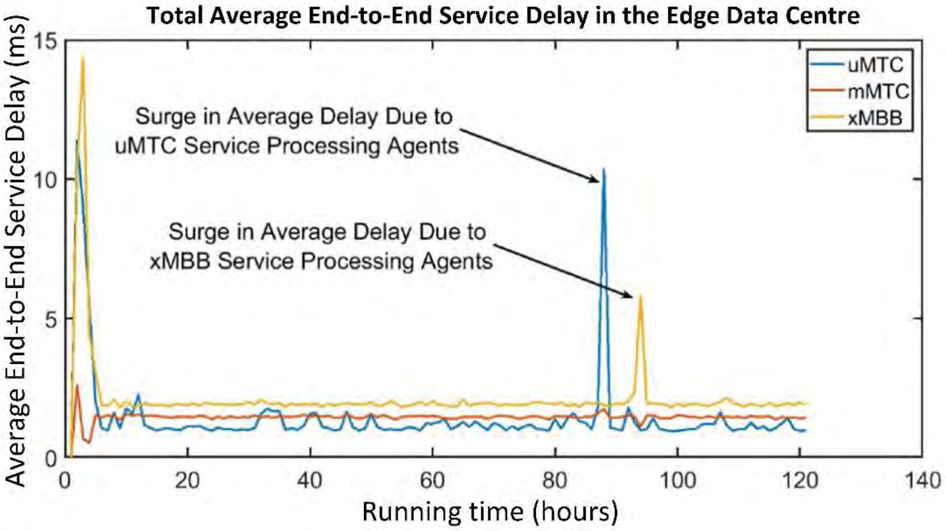


Figure 3-5: Impact of server failure on E2E latency in the edge data centre.

Agents require Central Processing Unit (CPU) resources to process services. The instantiating of multiple agents consumes data centre resources such as memory, storage, CPU, and network bandwidth. In the simulation, we considered only CPU, assuming equivalent requirements would be scaled for memory, storage, and bandwidth (this assumption can be more accurate in case of limited CPU jobs). In the simulations, only the number of agents that are scaling with the workload are considered. The number of active agents instantiated at a given time is directly related to the service prediction. Service workload is predicted by the service workload predicting agents. Figure 3-5 depicts the overall service latency in the edge data centre. This latency includes the decision delay in the agents' chain, the queuing delay, the processing delay at each agent, and the interconnecting links (virtual links) delay. The trend shows there is a significant latency on each type of service processing, depending on the time of failure of the server hosting a given type of service processing agents. This shows the importance of resiliency methods to be designed for edge resource availability to the agents.

3.1.5 Implementation issues of multiple control loops

The network management automation based on CLs provides immediate response to events based on complex algorithms and eliminates human error. In network management, a feedback-based CL approach is typically used to provide network or service self-configuration, performance optimisation, fault management etc. This approach is widely referred to as Monitoring-Analysis-Planning-Execution (MAPE). MAPE is typically single-objective oriented; however, it has to cope with multiple objectives in network management. A list of unresolved issues related to CL-based network management is presented below:

- Numerous mechanisms can be used to achieve the optimisation goal – the reconfiguration may concern the network or application (application-level routing, cache reallocation, content to network adaptation, etc.). The selection of the mechanism used to achieve the required goal (e.g., KPI) can be a problem.

- Different CLs may ask simultaneously for network reconfiguration forcing a change of the same parameters, which may lead to the chaotic behaviour of the system [BKA21] [ISB+14] [MGG+18].
- The reconfiguration requesting CL should be informed if the reconfiguration has been enforced, enforced partially, or not enforced, as it may have an impact on its learning.
- The implementation of the reconfiguration may take some - in general unknown - time. During such transition, no other reconfiguration should be performed, and the monitoring system should be aware of the transition.
- The environment monitoring is typically delayed, as it requires collecting raw monitoring data, processing and analysis. Such delay can be variable due to the jitter of links transferring monitoring data and variable delay of data processing algorithms. The delay in the monitoring may lead to ping-pong reconfigurations (the monitoring information is obsolete).
- Changes in network topology caused by adding new nodes (subnetworks) or failures may cause a strange behaviour of control algorithms Decision Elements (DEs) designed or trained for the previous topology. Significantly, the performance optimisation algorithms should be able to cope with the flexible network topology.

All the above-mentioned issues must be solved to allow commercial CL-based management deployment, as network stability is a primary concern to network operators. It has to be noted that the stability problem in such a complex environment cannot be solved easily. However, some techniques can reduce the probability of unstable system behaviour:

- Decomposing the CL-based architecture into a set of CL-based subsystems coordinated hierarchically. Due to such decomposition, the number of CLs with common parameters is reduced, and the use of distributed AI with online learning can significantly reduce the problem of ‘interfering’ CLs.
- “Separation” of CLs by different operating time scales [QYC+19]. The probability of direct conflicts between other CLs is significantly limited in such a case.
- Implementing operation sequencing of CLs operations of different system layers trying to solve the same problem. Such an approach is already used in fault management in classical management.
- Blocking of new reconfigurations when another reconfiguration is in progress.
- Monitoring the impact of individual CL decisions on the overall system, detecting the ping-pong effect, degradation of KPIs due to CL-based reconfiguration and unstable system behaviour. Implementing mechanisms to restore system stability and proper KPIs using the reconfigurations history.
- Using priorities for CL operations and assigning the highest priority to CLs involved in fault management and the lower to those involved in energy consumption optimisation is less critical which are less important during fault mitigation phase.

Some of the mechanisms mentioned above concern algorithms; however, some impact the management architecture. A high-level proposal of the architecture of the CLs coordination subsystem is presented in Figure 3-6. In the figure, each CL obtains information about the controlled system (network or service) from the Monitoring Subsystem, which is assumed to be shared by all CLs – a publish/subscribe mechanism can be used to obtain information needed for each CL. Using the MAPE approach, each CL can request reconfiguration of the controlled solution (network, service, etc.) based on this information. The reconfiguration requests are queued with priority assigned to each CL by the System Operator. The Reconfiguration Enforcement Engine takes the input from the queue and triggers a change of system parameters. At the same time, it informs all CLs (using N/ACK signal) that system reconfiguration is in progress to avoid their reaction during the transition state. The modification of the parameters is enforced by Actuator (ACT) entities that also provide information about completing the process using the Command Execution Confirmation (COM) signal. When the feedback from all ACTs involved in reconfiguration is obtained, all CLs are informed that the reconfiguration is completed, and the new reconfiguration is stored in Reconfiguration History Database (RHDB).

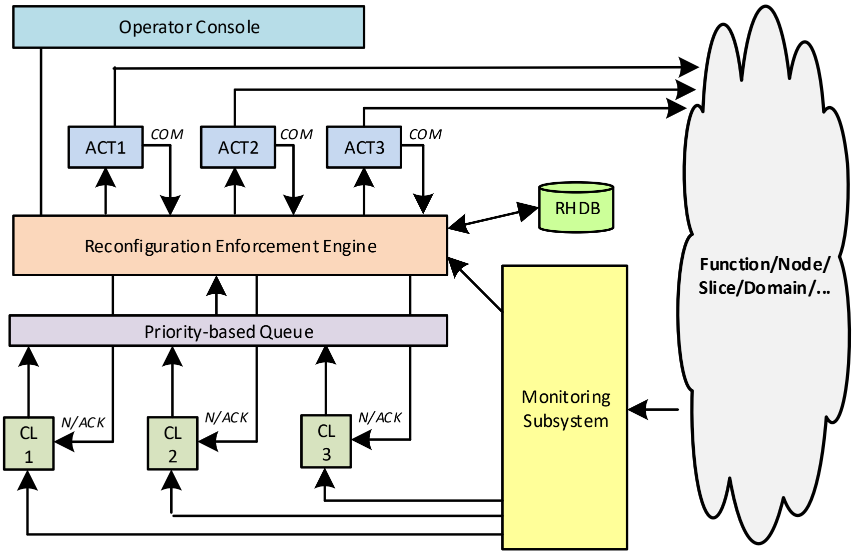


Figure 3-6: A high-level architecture of a subsystem responsible for the coordination of CLs and for providing the controlled system stability

In case when a requested reconfiguration cannot be enforced in a predefined period, the request is cancelled, and the CL is informed about that. Information about the reconfigurations and resulting KPIs degradations below a predefined threshold as well as detected by Reconfiguration Enforcement Engine stability issues are also sent to the Operator console. The Reconfiguration Enforcement Engine can restore the system to the previous stable state using information stored in RHDB with or without the permission of the System operator, which may ask for the execution of an entirely different action. The details of the proposed approach are implementation-dependent; however, the presented ideas could be a part of the CLs-based network or service management solutions.

3.2 AIaaS and analytics framework

3.2.1 Introduction

An important characteristic of the 6G era is an inclusive and collaborative environment in which a considerable number of AI applications can be readily deployed. Aside from AI-enabled network functions like AI-powered channel prediction and AI-powered mobility management, 6G benefits third-party AI applications from commercial partners and consumers, which require significant space and time flexibility in resource utilization and share several similar attributes. In other words, much like how it provides communication capabilities today, 6G transforms the network into a powerful distributed AI platform and exposes its AI capabilities to consumers. On the other hand, with the 6G networks, analytics capabilities may also need to be utilized to address any requirements coming from 3rd party applications and also be able to consume the data set and trained models from another plane or domain. This is enabled by having a framework that can facilitate the exchange of cross-plane/domain knowledge and resources to improve the accuracy of developing the analytics and training the ML algorithm. This framework can be advantageous not only for seamless transfer of analytics across cross-domains or planes but can also benefit from the Hexa-X concept of AI-as-a-Service (AIaaS).

The AIaaS, consisting of enablers and APIs offering AI functionality to other network functions, AFs, and third parties, arises to support distributed AI services such as analytics, prediction, classification, etc. These services can be virtualized and packaged to be deployed, activated, and reconfigured on demand to support the various network optimization decisions. This approach enables the agile AI-driven orchestration of 6G networks and services, with concurrent and potentially cooperating AI functions able to address the complexity of different optimization aspects with data, models, and algorithms specifically tailored to the various objectives and constraints of specific network domains, slices, and subnets, or services. In practice, this requires a novel design approach that includes shared services and AI functions for AI/ML algorithms and LCM of AI agents, together with the definition of APIs, data models, and metadata for agents and algorithms' capability discovery and data source requirements.

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

This section suggests an AI system architecture (as part of a 6G network architecture) aiming to address the requirements of the future EU AI Regulation, which is currently available as a draft [EUAI21] and incorporates features of the AIaaS paradigm. Key objectives of the EU AI regulation include:

1. Ensure that AI systems placed and used on the European Union (EU) market are safe and respect existing legislation on fundamental rights and EU values;
2. Ensure legal certainty to facilitate investment and innovation in AI;
3. Enhance governance and effective enforcement of existing legislation on fundamental rights (e.g., General Data Protection Regulation) and safety requirements applicable to AI systems;
4. Facilitate the development of a single market for lawful, safe, and trustworthy AI applications and prevent market fragmentation.

As key focus of the EU AI Regulation, essential requirements are being introduced for the case that AI systems are applied in critical fields (“high risk” AI systems), such as biometric identification, critical infrastructure, etc. Recently, the European Council [EUAI21a] added “digital infrastructure” to the list of such “high risk” AI systems including cellular communications infrastructure. With those additions, the EU AI Regulation becomes even more important for future 6G systems. The draft AI Regulation [EUAI21] is introducing a series of articles which are comprising specific technical requirements tailed to “high risk” systems. However, note that the discussion is ongoing on the level of the European Parliament and Council and the final agreement may still change.

To ensure the compliance of the 6G architecture to meet the related essential requirements, it is proposed to translate these requirements into related architectural components, (please refer to Annex A.2 for details). Each proposed component may host a single or multiple functions. This architecture approach reflects the main requirements and further accommodates for user interaction and the connection of the AI system to a database that may be used for the provision of suitable reference training data, logging of user actions, logging of AI system behaviour, etc. The various entities (i.e., components and functions) are summarised below:

AI processing (equiv. to the AI agent defined in Section 3.1.3): It is the core of the AI system and, with respect to supervised and unsupervised learning approaches, it consists of a model, typically being trained using some training data set and optionally some additional data that is being acquired, while the AI system is being operated. Such an AI system can rely on various ML methods/approaches including regression, classification, clustering, dimensionality reduction, ensemble methods, Neural Networks (NNs) and deep learning, transfer learning, Natural Language Processing (NLP), and word embeddings. The entity for AI processing typically uses a model that is trained through suitable training data provided by the attached database. Furthermore, the correct operation of the AI system is monitored and controlled through an authorised supervisor. In case that any undesired behaviour is being detected, several possible steps may be taken for example, the supervisor may trigger a retraining of the model using authorised and error-free training data or may report related behaviour to the manufacturer. The entity for AI processing is typically interacting with all other entities of the AI system as further detailed below. This interaction ensures that all requirements of the AI Regulation are being met – starting with market introduction of the AI system and furthermore including the permanent supervision during the operation of the AI system (e.g., with the objective to identify any introduction of biases into the decision-making processes of the AI system).

Self-verification (partly equiv. to the AI monitoring function defined in Section 3.1.3): It is used to verify the correct operation of the newly trained model, typically in conjunction with an AI monitoring function providing inputs for self-verification. In particular, it is proposed to use a pre-defined test-data set (which is different from the data set used for training) as input to the AI system in order to verify the correct operation of a given model. Only if the correct operation is verified, the AI system is allowed for full usage for its intended purpose. In the opposite case, e.g., in case that biases are detected or any unexpected behaviour, the operation of the AI system is interrupted, until the issues are resolved. Such a verification step is periodically repeated in case of retraining of the AI system with new data. A key requirement of the AI Regulation relates to the avoidance of biases, which requires, among others, that distinct user groups are being treated equally. It is indeed possible that any AI system develops such biases during on-going retraining processes – those need to be detected in the earliest stage possible and suitable counter measures need to be taken. One possibility is to put the system back into a predefined state by applying approved and verified training data to derive the AI model.

Record keeping (related to the AI monitoring function defined in Section 3.1.3): When the system is finally used for its intended purpose (after all successful verification steps), this entity logs all user interactions (commands given by the authorised user, etc.) and records the behaviour of the AI system and stores relevant information in the database.

Risk mitigation (related to the AI monitoring function defined in Section 3.1.3): This function proposes a trade-off between risk and functionality to the user. For example, the entity may propose that the system is constantly retrained using the observed information obtained during the operation. The upside is that this may improve the quality of the AI decision making. The risk is that the new data may introduce biases or other undesired characteristics (e.g., increasing the risk of obtaining malicious training data). Besides application requirements, such as the needed level of inferencing accuracy, etc., the authorised user will need to decide whether corresponding risks are being taken to achieve the expected improvements towards addressing the requirements during service time.

Processing risk (related to the AI monitoring function defined in Section 3.1.3): The entity for Processing Risk takes the results of other entities, such as the “Self-Verification” entity, and processes identified risk related information and unexpected behaviour information, such that it can be presented in a concise way to the authorised user (e.g., by illustrating statistics on the decision making, including outlining of unexpected biases of the statistics, etc.). In case of an issue, the user can use the provided information to take action, e.g., to terminate the AI system operation through the “Human Oversight” entity.

Human oversight (may be mapped to the AI NSSMF entity, introduced in Section 3.1.3): This function allows the authorised user to take action in case the AI system operates in an unexpected or undesired way, e.g., in case that the decision-making processes indicate biases. The user may then take several actions, including termination of the AI system operation, enforce a retraining of the system, choose a different risk trade-off through the “Risk Mitigation” entity, etc.

AI system redundancy (may be mapped to the AI NSSMF entity, introduced in Section 3.1.3): It oversees redundant replacement options for critical entities/components/elements of the AI system. In case some malfunctioning entity/component/element is identified, typically relying on information by the “Self-Verification” entity, then this entity is used to configure a corresponding replacement. After the replacement, the correct operation of the AI system is typically verified, again relying on information by the “Self-Verification” entity. If it is successfully verified, then the operation of the AI system may continue.

Management entity (may be mapped to the AI NSSMF entity, introduced in Section 3.1.3): It orchestrates the interaction between the different building blocks indicated above. For example, when one of the entities of an AI system is dysfunctional or operates in an unexpected way, this entity may detect this behaviour relying on information by the “Self-Verification” entity and may trigger the replacement of concerned components/entities by redundant replacement components/entities through the “AI System Redundancy” entity.

Note that the upper entities may part of a specific AI system or may be provided as services through (remote) access. The inter-dependencies of the entities are an open issue and will be addressed in the process of implementing the regulation.

Besides the possibility of interacting with an integrated database, the AI system may be interacting with external (independent) entities/components operated by 3rd parties. A typical example among the High-Risk list of Annex III of [EUAI21] is “*2. Management and operation of critical infrastructure: (a) AI systems intended to be used as safety components in the management and operation of road traffic and the supply of water, gas, heating and electricity*”.

3.2.3 Analytics framework

In 5G, the network analytic capabilities are enhanced to be able to use AI/ML techniques [23.288]. Although this leads to more accurate statistics, prediction, and analytic reports, the ML models are not shared but limited to the boundaries of a domain or plane (e.g., CP, management domain) in which the analytic providing entity resides.

As stated in Hexa-X D5.1 [HEX-D51], various functional entities exist which provide local analytics, such as NWDAF in the CP [23.288], and Management Data Analytics Service (MDAS) in the management plane [28.809]. These entities have a limited list of analytics that can be provided for an analytics consumer. Furthermore, performing those analytics requires input data sets coming from pre-defined sources. As an example, in order for NWDAF to provide NF load analytics, it is required to get information regarding a load of specific NF instance and NF status from the Network Repository Function (NRF), the usage of assigned virtual resources and the life cycle changes from Operations, Administration and Maintenance (OAM), and the report of UP traffic from UPF. By raising the next generation of mobile networks, analytics capabilities may also need to be utilised to address any requirements coming from 3rd party applications, and also be able to consume the data set and trained models from another plane or domain. This becomes a possibility by having a framework that can facilitate the exchange of the cross-plane/domain knowledge and resources aiming to improve the accuracy of developing the analytics and training the ML algorithm. This framework can be useful not only for seamless transfer of analytics across cross-domains/planes but can also benefit from the Hexa-X concept of AIaaS. It is a new AI-enabled architecture that intends to support distributed AI agents which are providing AI services such as analytics, prediction, classification, etc. These AI agents can be virtualised, packaged to be deployed, activated, and re-configured on-demand. 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 automatic fashion. This leads to opening up domains and planes to seamless transferring of analytics and trained ML models. Model transferring across domains/planes has also had some challenges which required particular attention such as version controls of the models as well as having a mechanism to properly sync the repositories in different domains/plans with the latest updates on the models. The NWDAF defined by the 3rd Generation Partnership Project (3GPP) in TS 23.288 [23.288] plays a key role as a functional entity that collects KPIs and other information about different network domains and uses them to provide analytics-based statistics and predictive insights to network functions in the 5GC. Through the service-based oriented interface NWDAF, NFs can request/cancel the subscription to network analytics service for a particular context. The 5G system architecture allows NWDAF containing Analytics Logical Function (AnLF) to use trained ML model provisioning services from another NWDAF containing Model Training Logical Function (MTLF).

The NWDAF service consumer (e.g., CN NFs, 3rd party applications, etc.) by invoking the relevant NWDAF service can subscribe, modify, or cancel a subscription for an ML model associated with an analytics ID. Depending on the ML model, NWDAF can determine whether an existing trained ML model can be used for the subscription, or it needs to generate an initial ML model for the subscription. For NWDAF consumers to use the ML model provisioning service, it requires providing some input parameters such as a list of analytics ID(s) which can identify the analytics for which the ML model is used, analytics filter information which indicates the conditions to be fulfilled for reporting analytics information, and target of analytics reporting. In return ML model provider NWDAF creates/retrains the requested ML model and provides the consumer the relative result of ML model information.

Hexa-X analytics framework reuses the NWDAF concept by providing APIs for AI agents in different planes to exchange the ML models or the required training data set. One of the uses of an analytics framework is to improve and extend the RAN management and CN communications which are currently limited to request/response through the OAM. Currently no method exists in order to define the ML model sharing across domains since the domain analytics concepts are still evolving and therefore the cross-domain interactions have not yet been considered. Having the analytics framework to enhance the communication between RAN management and the CN can facilitate the query and sharing of the ML models and training data sets. The methods defined in the Hexa-X analytics framework extend the capabilities of functions respectively defined by 3GPP and O-RAN Alliance [ORA21a].

The RAN Intelligent Controller (RIC) is a central software component of the O-RAN architecture. It is a key element in the management of network functions. Both Non-Real Time RIC (Non-RTR) and Near-Real Time RIC (Near-RTR) components manage separate functions of the RAN. The Non-RTR portion manages events and resources with a response time of 1 second or more. The Near-RTR portion manages and events and resources requiring a faster response down to 10 ms. The Non-RTR is described as a function that resides within the Service Management and Orchestration (SMO) framework [ORA21a]. Non-RTR functional extensibility is accomplished through modular applications which can be considered as AI agents (a.k.a. rAPPs) that can be understood as running within the Non-RTR function itself. An AI agent is a network automation tool that can maximise the radio network's operational efficiency and realise different RAN automation and management use cases, with CLs on a time scale of one second and longer. Because these AI agents are associated with the Non-RTR, any framework services exposed to them can be considered as being exposed by the Non-RTR. Hence, an open and standard interface has been defined through which the Non-RTR can expose SMO framework functionalities to these AI agents.

Apart from AI agents, Non-RTR is also including other functionalities such as "AI/ML workflow functions" which is an implementation variability function and can be "AI/ML Model Management Functions", "AI/ML Data Preparation Function", "AI/ML Modelling/Training Function", and "AI Model Repository". The AI/ML training functionality allows the training of AI/ML models of 3rd party applications within the SMO/Non-RTR. The inputs consist of training data and an ML model. The input ML model is described by the AI/ML model descriptor/metadata. This functionality also allows the update/re-training of ML models stored in the ML model repository. The output of the functionality is a trained, validated, and tested ML model, which is ready to be deployed within an AI agent (e.g., in form of a Non-Real Time RAN Intelligent Controller Application (rApp)) or external application and also can be stored by the SMO or by the Non-RTR using the ML model repository functionality [ORA21].

Similar to RAN management architecture, the CN's functionalities can be exposed to an external entity through pre-defined APIs. As an example, the Application Function (AF) in the CN is a conceptual placeholder for all the external applications that need to communicate with CN functions. In order to facilitate the cross-domain ML model exchange, Hexa-X analytics framework can provide APIs for both NWDAF in the CN and Non-RTR in the RAN management domain to be able to share the trained ML models.

Parallel to the Hexa-X FL model sharing, e.g., FLaaS (see section 3.2.4), the analytics framework requires the support of the following in order to enable the cross-domain analytics exposure:

- An extension of an API between the exposure function in the CN and an external AI agent (e.g., in form of rAPPs) devoted to managing the registration, discovery, and transfer of the trained ML models across domains. Moreover, the API enables an entity from one domain to request the training of an ML model to be executed in the other domain.
- A dedicated repository for ML model's metadata which is accessible by both analytics entities on both domains. This metadata can include the model characteristics such as training parameters, evaluation metrics, data set versions, etc., as well as the model ID, model, and data version.
- A procedure to register the trained ML model's metadata in the abovementioned repository: in order to enable the discovery of the available trained ML models in each domain, NWDAF registers a list of available ML models in the repository. Non-RTR can also register its ML models through the new API in the repository. This procedure makes it possible for both NWDAF and Non-RTR to be aware of all available trained models from cross-domain. They can discover and query the ML models similar to the local ML models.
- In addition to the procedure explained above, some other aspects such as scalability in terms of managing the rapidly changing ML models and security and privacy in case the domains are not part of the same trust zones of the analytics framework need to be considered carefully for the future enhancement.

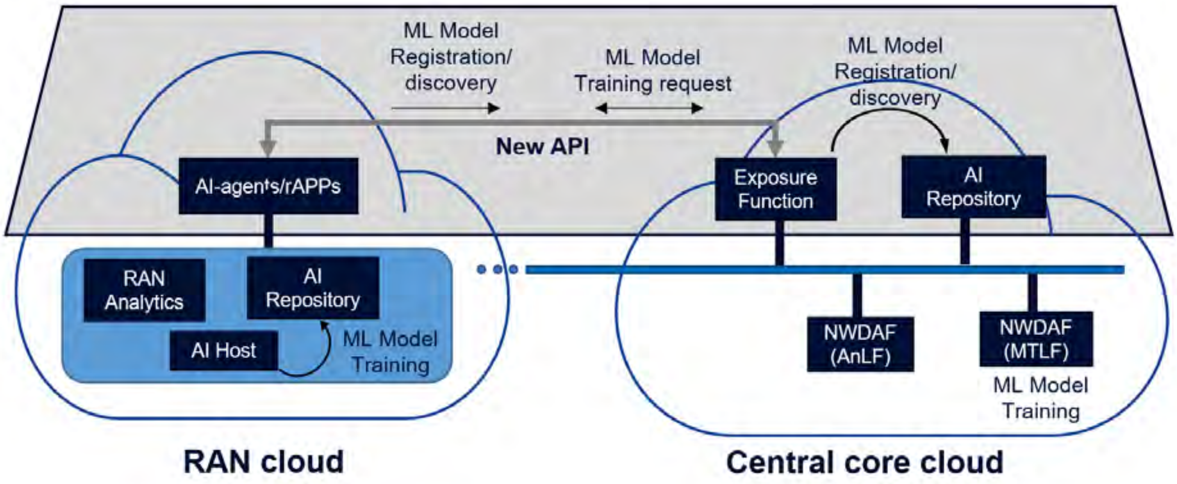


Figure 3-7: Hexa-X analytics framework.

3.2.4 Distributed AI services

The interacting and cooperating robots use case [HEX-D12] is built on various interaction models, which encompasses direct machine-to-machine interaction and human involvement towards a common goal. The cooperating robots ("cobots" in short) with AI processing capabilities, the main actors of interaction models with high-level goals, call for APIs for services. Robotics-as-a-Service, as shown in Figure 3-8, consists of multiple services for cobots (image processing, pattern detection, knowledge gathering, path planning, etc.) to accomplish a particular task (e.g., picking an object). On the other hand, service requirements motivate the need for local computing capabilities as part of a 6G system to ensure reliable execution of AI models and that training data and derived models remain private.

This comprehensive use case covering positioning accuracy and integrity, localization and mapping, flexibility, and trustworthiness aspects and the relevant KPIs and KVIs stated in [HEX-D71] necessitates additional architectural means to enable distributed decision making on less powerful or constrained devices. The challenge is how to perform distributed decision making when different cobots need to collaborate. They need to benefit from AI for the intelligent optimization of cooperative tasks and ensure that changing communication requirements are met when humans and machines dynamically form collaborative groups.

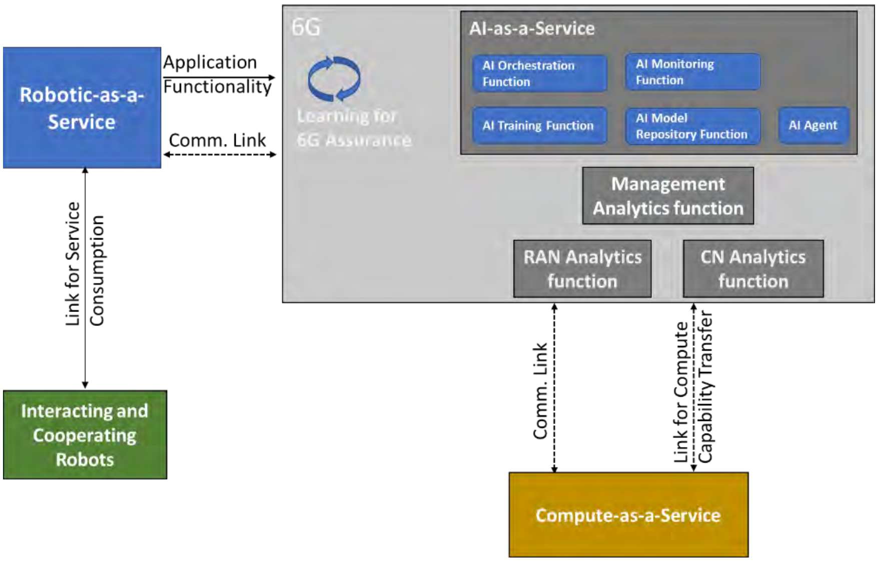

Robotic-as-a-Service
Application Functionality
Comm. Link
Link for Service Consumption
Interacting and Cooperating Robots
6G
Learning for 6G Assurance
AI-as-a-Service
AI Orchestration Function
AI Monitoring Function
AI Training Function
AI Model Repository Function
AI Agent
Management Analytics function
RAN Analytics function
CN Analytics function
Comm. Link
Link for Compute Capability Transfer
Compute-as-a-Service

Figure 3-8: High-level architecture integrating AI to enable Interacting and Cooperating Robots use case.

To couple AI with the 6G architecture, there is a need for a reasoning framework coupled with the architecture that systematically senses data from the environment, analyses collected data, and then applies the discovered knowledge to optimize performance for 6G and AI agents. For this purpose, AIaaS is proposed as an enabler to implement a new AI-enabled architecture that intends to support distributed AI services, AI service chaining, cross-domain AI service consumers, and data producers [HEX- D13].

AIaaS and its functionalities, as discussed in Section 3.1.3 and Section 3.2.2, transforms the network to a powerful distributed AI platform and provides AI capabilities to service consumers. AI orchestration function supporting AI service chaining addresses the matching/ pairing of an AI service consumer to the appropriate AI service provider, underlying communication system, AI model management, and diagnostic services by utilising AI model repository function, AI training function, AI monitoring function and AI agent as described in Section 3.1.3 and in [HEX-D51]. Additionally, AIaaS supports cross-domain AI service consumers by compressing raw data or initialising training at the agent/edge by AI training function so that the continuity of the AI service is ensured in congestion, or resource outage or by revising the distribution pattern or periods of the global model stored in AI model repository function by the AI Orchestration function with the help of monitoring function. Lastly, AI function following AIaaS concept (depicted as AIaaS block in Figure 3-8) has a connection with an additional service that supports, registers, and manages data producer capabilities for AI services. Considering the image processing example of Automated Guided Vehicles (AGV) in the interacting and collaborating robots use case, there will be numerous high-resolution images. Processing them on the AGV can inflate the storage and transmitting them in the air interface can cause latency and intense resource consumption. Therefore, extracting features from raw data on AGV and sharing them with the provider can solve the aforementioned problems. If the AGV has enough processing resources and energy for model training, the first layers of the model can be transferred to the AGV that speeds up the processes and reduces the amount of data to be transferred.

Cobot operations such as object detection, path planning, localization, and knowledge sharing require access to both AI services and computational resources. As specified in [HEX-D12], the computational resources available at the edge/fog/cloud layers or at cobots may be abstracted away within the CaaS paradigm, where the service consumer can issue a request for AI process offloading to the CaaS utilizing AI service. In return, the most appropriate compute node (trusted, available, and with sufficient resources) for process offloading is identified by the CaaS provider via optimisation solvers and delegated to the identified compute node. The obtained processing output is forwarded to the service consumer tagged with the computational latency, energy consumption and trustworthiness requirements [HEX-D51].

The goal is to implement a reasoning service that optimises the placement of services subject to data, AI agents and their tasks, AI models, computing resources, and communication interface constraints. The AIaaS in Figure 3-8 is responsible for acquiring basic information about AI agents to enable Application functionality as a first step. Then, based on the data features, the output of the implemented policy, and KPIs retrieved, it assists the AI Model Repository function and AI training function in the selection and LCM of the AI model implemented on the Robotics as a Service consuming AI agents. Based on the evaluations done by AI Orchestration function, AI Model Repository function and AI training function upload/download/update/delete/store/monitor AI models, transmit AI models to users efficiently per situation and inform AIaaS about the models, perform data analytics for user experience, and generate analysis results to help cloud to train and improve AI models. At the same time, AIaaS retrieves information about AI agents operating in CN Analytics function, RAN Analytics function and Management Analytics function to manage network, i.e., optimisation of slices or provisioning of RAN/CN, congestion control, forecasting of resource usage information in a predefined future time, scaling of resources, admission control, load balancing of traffic, etc., autonomously. It also assists in the selection and LCM of learning models for 6G assurance. 6G service continuity is ensured by providing a communication link – Uplink/Downlink (UL/DL) or Sidelink (SL) – for the Robotic as a Service consuming Robots. Additionally, 6G service continuity is necessary for forming data link for CaaS to transmit information about offloading paths between provider and receiver with sufficient capacity, and data link for UL and DL resources for model transfer between AI Model Management as a Service and AI agents. Last, AIaaS helps to determine the optimal compute node for workload delegation via multi-objective optimisation and itself requires CaaS resources for training and inferencing. AIaaS and CaaS are jointly needed by the robots to perform functions (e.g., object detection, localisation, path planning). AIaaS and CaaS also perform the intelligence distribution based on the comparison between computational latency of the Graphics Processing Unit (GPU), energy consumption, GPU power consumption, and the trustworthiness requirement.

3.2.5 Federated learning as-a-Service

6G network will allow UEs to take advantage of intelligent services (e.g., forecasting of QoS) by exploiting AI models that are built in a collaborative fashion. According to the FL approach, such models can be obtained without potentially disclosing private data of the UEs. Following the AIaaS paradigm, Hexa-X envisions an FLaaS framework that allows UEs to discover FL services made available by the 6G network, obtain the corresponding "federated" AI model and, possibly, participate in building it. To this aim, the 6G network must provide new protocols that handle the interactions among the entities involved in such framework.

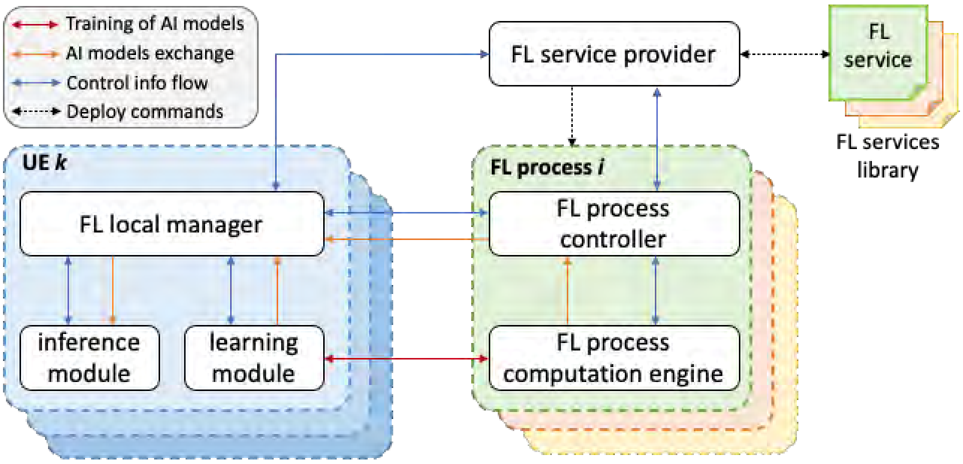


Figure 3-9: High-level architecture of the FLaaS framework.

Figure 3-9 shows the components involved in the FLaaS framework and their interactions. We refer to an *FL service* as a collaborative learning task dedicated to a specific application (e.g., Quality of Experience (QoE) prediction for automotive applications). The FLaaS framework provides a collection of FL services that can be instantiated when needed, i.e., when an AI model for that specific application is requested by some UE. We refer to a running instance of an FL service as *FL process*. Note that multiple FL processes referring to the same FL service may be running simultaneously in the network, for example handling disjoint sets of UEs in different geographical areas. The main component of the framework is the *FL Service Provider* (FSP), which is located in the CN and maintains both the library of available FL services and the list of active (running) FL processes in the system. Each UE is supported by an entity called *FL Local Manager* (FLM) that manages CP interactions with the network-side of the FLaaS framework on behalf of the UE application. Moreover, it manages both the *learning* and *inferencing* modules of the UE, which are the entities that actually train and the exploit the AI models, respectively. The FLM can reside on either the UE device or at the edge of the network, e.g., as an edge cloud application. When the UE wants to discover an FL process, its FLM queries the FSP. The latter is also responsible for orchestrating the entities that will actually execute the FL processes. In fact, each active FL process is composed of two modules, namely the *FL Process Controller* (FPC) and the *FL Process Computation Engine* (FPCE). The former manages CP interactions with the FSP (e.g., authorization grants) and the UEs' FLM, whereas the latter acts as the aggregator of the FL process, i.e., the entity that actually builds the global AI model. To do this, the FPCE must exchange local and global AI model updates with the learning module of the UEs, which, in their turn, act as collaborators in the process of training the global AI model. It is worth mentioning that the deployment of the above entities is immaterial: the FSP may reside either in the core cloud or at the edge of the 6G network. Likewise, the FLM, the learning and the inference modules may reside at either the UE or the network edge, e.g., according to the CaaS paradigm. This last option may be necessary with resource constrained UEs, such as Internet of Things (IoT) devices.

In the following, we describe the features that enable the above framework. For each operation, we describe the necessary interactions among the components and, where applicable, we define alternative options that may present different trade-offs.

Onboard an FL service: The FSP should provide an interface to operators or third parties to allow them to interact with the FLaaS framework. The interface should provide functions to register/unregister an FL service. Such registration message should provide information such as the description of the service (i.e., its objective), the application image and configuration files, possible constraints (e.g., minimum capabilities of the UEs participating in the training) and type of training (see training operation below). The interface should also provide mechanisms to allow the operator or third party to instantiate an FL process. In this case, the FSP deploys the entities (e.g., VMs or containers) acting as FPC and FPCE starting from the application image that was previously onboarded within the FSP.

Discovery: The FLM queries the FSP to obtain the list of available FL services. The response may include the conditions that need to be met for the FSP to start a new instance of that FL service, i.e., an FL process. Alternatively, the FLM can obtain the list of the active FL processes in the system. The request messages may have fields for filtering the required FL service the UE may be interested in (e.g., services available in a given geographical area, or services relevant for specific use cases only).

Join an FL process: Once the UE is aware of the available FL services and processes, it may be interested in obtaining the corresponding AI model. Note that it is not strictly required that the UE participates in the construction of the model itself. The FLM contacts the FLSP, which checks whether to grant the authorization to the UE and, in the affirmative case, returns the endpoint of the FPC (e.g., IP address/port) to the FLM. Then, the FLM can do the following actions:

- obtain the global AI model available for the given FL process, if any;
- take part in the construction of the global AI model.

Obtain the global AI model: The FLM can request the global model from the FPC according to either a request-response or a subscribe-notification paradigm. Such paradigms allow the entities to interact directly without the need of an intermediary (i.e., a broker). The first option is useful to allow the FLM to obtain the model whenever it wants. However, some timestamping mechanism may be needed to prevent sending global models update too fast (e.g., before a new one is available) and flooding the network. With the subscribe-notification pattern, instead, the FPC sends global model updates as soon as the FPCE produced a new model. The subscription message from the FLM may specify filter conditions that regulate the number of updates it expects to receive (e.g., one model update per day).

Join the training of the global AI model: The FLM notifies the FPC its intention to participate in the building of the global AI model. The FPC must either accept or reject the request. Motivations for rejection may include, for instance, insufficient computational resources available at the UE for training a local model, or position of the UE outside the geographical area of interest for the given FL process. Note that the FLM is not allowed to join the training without first completing the join operation described above, since the FSP must first authorise the FLM. This can prevent some type of training poisoning done by malicious users that bypass the authorisation of the FSP.

Training phase: This phase can start when the UE has successfully joined the training process. On one hand, the FPC and the UE's FLM take care of exchanging CP information like the status of the training process and changes in the availability of UE's computational resources. On the other hand, data-plane information is exchanged between the FPCE and the UE's learning module. This entails sending the global model from the FPCE to the local learning module and sending the local model in the opposite direction. Both models can be sent as a whole, or by incremental updates. The FPCE can adopt both a *synchronous* and an *asynchronous* strategy to perform the training:

- In synchronous mode, training takes place in multiple *rounds*. For each round, the FPCE selects the participants, i.e., a subset of the UEs that are available to train the model and sends them the notification of round start to their learning module. If some UE became unavailable after the selection phase, it sends a Negative Acknowledgment (NACK) back to the FPCE, which may select other UEs before starting the round. Upon reception of an ACK from the selected participants, the FPCE sends them the global model. In turn, the learning module of each participant starts its local training and sends its updated local model to the FPCE. When all the participants (or a predefined percentage of them) have sent their update, or after a predefined deadline, the FPCE aggregates the received local models and builds a new version of the global one.

- In the asynchronous mode, whenever a UE joins the training, its learning module receives the global model from the FPCE, performs the local training and sends the model back to the FPCE. The latter aggregates such models and builds a new global model as soon as it receives an update from the UE.

Once the global AI model has been produced, the FPCE sends it to the FPC, which stores it and provides it to FLMs that request it.

Leaving the training of the global AI model: A UE may decide to stop participating in the construction of the global model. For instance, this may occur when its computational resources become too scarce, or when it has not been selected for participating in a round for a long time. In this case, the FLM sends a leave request to the FPC, which revokes previous authorisations and responds with an ACK. However, the FPC itself may decide to remove a UE from the training process. For instance, this may occur because the UE moved outside the geographical area of interest for the FL process, or because the UE became too slow to perform its local training. When such an event occurs, the FPC sends a notification to the UE's FLM, which in turn replies with an ACK.

Leaving an FL process: A UE may decide to leave an FL process, e.g., because it is no longer interested in using a global AI model for the given FL service. In this case, a message is sent from the UE's FLM to the FSP, which takes care of withdrawing the authorisation to contact the corresponding FPC. If the operation is done before the UE left the training (see above), the FSP is also responsible to inform the FPC to remove the UE from the set of potential participants to the training of the global AI model.

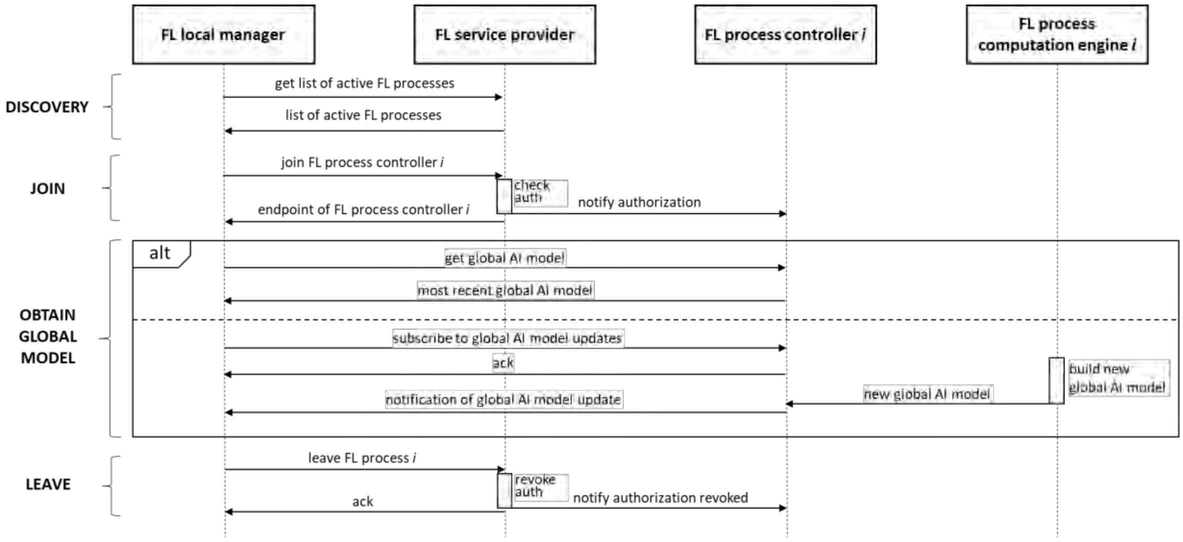

FL local manager
FL service provider
FL process controller *i*
FL process computation engine *i*
DISCOVERY
get list of active FL processes
list of active FL processes
JOIN
join FL process controller *i*
check auth
notify authorization
endpoint of FL process controller *i*
alt
OBTAIN GLOBAL MODEL
get global AI model
most recent global AI model
subscribe to global AI model updates
ack
build new global AI model
new global AI model
notification of global AI model update
LEAVE
leave FL process *i*
revoke auth
ack
notify authorization revoked

Figure 3-10: Procedure to discover and join an FL process, obtain the global model, leave the FL process.

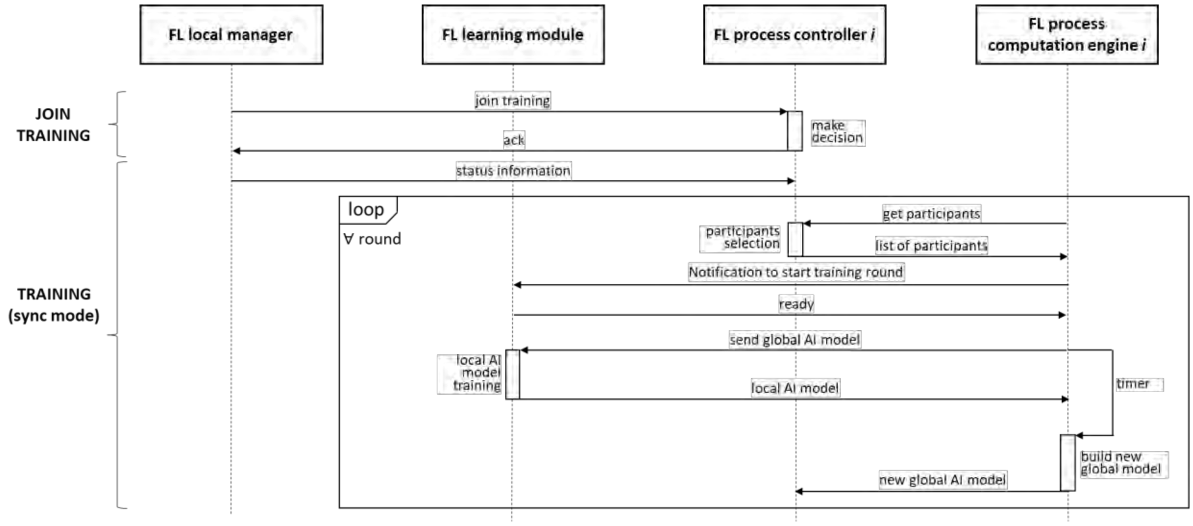


Figure 3-11: Procedure to join the training of a global AI model, and to train it.

Figure 3-10 depicts an exemplary sequence diagram of the interactions occurring in the FLaaS framework when a UE wants to discover and join one active FL process, so as to obtain a global AI model that its inferencing module can exploit. The figure reports both the cases of request-response and subscribe-notification patterns. The leave operation is also shown in the figure. Figure 3-11 reports the operations required to join the training phase of an FL process, as well as the models exchange between FPCE and the UE's learning module. For simplicity, only synchronised mode is visualized.

3.2.6 FED-XAI demo

The FEDerated eXplainable AI (FED-XAI) Proof-of-Concept (PoC) will demonstrate the benefits of AI models that are built collaboratively according to the FL paradigm, in the context of QoE prediction in an automotive use case. The demo involves activities carried out in both in Work Packages 4 and 5. On one hand, AI models considered in the PoC will be built according to the FL algorithms developed in WP4. Such AI models will be inherently explainable [HEX-D42]; hence, we refer to eXplainable AI (XAI) models. On the other hand, the PoC will include a realisation of the FLaaS framework described in the previous section.

We consider an automotive scenario, where the prediction of the perceived quality of videos streamed via the 6G network is a relevant factor to determine the availability of advanced driving assistance systems, such as see-through and tele-operated driving. The QoE prediction can take advantage of XAI models trained locally by UEs and aggregated into a global, federated XAI (FED-XAI) model.

The PoC will be deployed as a real-time testbed, as shown in Figure 3-12. The mobile network will be realised using Simu5G [NSS+20], an open-source 5G network simulator that can also work as a real-time emulator, i.e., it can exchange real packets with real devices and applications [NSV+20], while the simulation time follows the same pace as the wall-clock time. Video traffic will be generated by a video-server application running in an end-device and sent to the PC hosting Simu5G, via Ethernet connection. The traffic will be then injected into a running instance of Simu5G – specifically at the UE side (e.g., the blue car in the figure) – and will traverse the (emulated) mobile network until it reaches its destination, i.e., the Multi-access Edge Computing (MEC) host in the figure. From there, traffic will be delivered to the corresponding destination end-device, where a video-player application will be running and playing out the video. In parallel, live QoS and QoE metrics are sent to the FLaaS, namely to its FLM (, see Section 3.2.5). Such data will be used to predict the QoE that the UE will perceive in the future, by exploiting a FED-XAI model produced by the FPCE. The prediction, as well as its related root cause(s), will be sent to an XAI dashboard, which will visualise them in real-time. Training operations for producing the FED-XAI model will be carried out using OpenFL [RGF+21], an open-source framework to train AI models according to the FL paradigm.

In order to test the FED-XAI approach in realistic scenarios in terms of traffic load and interference, we will dimension the mobile network simulated by Simu5G according to data gathered in a live RAN. In particular, Simu5G will be fed with data from the Minimization of Drive Test (MDT) functionality implemented in TIM's RAN. MDT is defined by 3GPP [37.320] and allows UEs to collect anonymised, geolocated, layer-2 measurements and report them to the network periodically (e.g., every second). Relevant metrics are Reference Signal Received Power (RSRP), Reference Signal Received Quality (RSRQ) and UL/DL throughput. Since not all UEs support or activate MDT, we will integrate such data with those coming from cell-wise statistical counters, which provides metrics (such as, e.g., throughput and number of active UEs) averaged over 15 minutes time windows. The above data will be used to simulate realistic background traffic within Simu5G, in addition to the video-streaming flow generated by real, external applications.

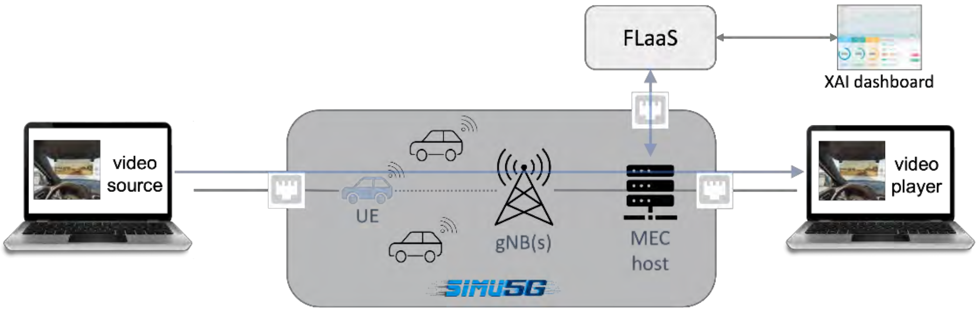


Figure 3-12: High-level representation of the testbed deployment for a tele-operated driving use case: a vehicle sends a real-time video stream to a recipient located on the MEC.

3.3 Interfaces, APIs, and protocols

5G network capabilities are functions which can extract information from a given network and, also, to configure the network. The equivalent network capabilities in 6G networks should be able to face the multi-domain requirements of these future networks, as described in the Hexa-X concept of *network-of-networks* [HEX-D13], to facilitate the inclusion and transformation of the different 6G-stakeholder networks to achieve a clear application domain to network domain integration. This integration is of paramount importance in order to be able to cope with the dynamicity, multi-domain, multi-stakeholder and heterogeneous requirements inherent to future 6G networks. The following subsections explore how to fulfil these objectives by describing various proposals related to domain interfaces, application/network APIs and protocols evolution.

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

The aim of the solution proposal in this section is to enable fine-grained, privacy-preserving user (or any other data-contributing entity) data LCM across security domains of a network, where the data is aimed to be used for either AI/ML-model training/ updating purposes or for inferencing purposes. In other words, it proposes a way with which different networks (with different data privacy policies) can best collaborate for ML model training on the basis of the widest possible pool of relevant data across these networks without breaching data privacy policies, either device-specific or network wide. Focusing on learning data, a device user (or any other data contribution entity in the network) can indicate data attributes of a specific client application instantiated to the device/ UE, or machine as private/ confidential or publicly shareable. Learning data LCM will then take place following strictly the user data privacy preferences either for the whole data set life cycle (e.g., till time of data deletion) or for specific timeframes. This management framework is also beneficial to AI agents carrying models, as it reduces the surface of data poisoning attacks; only possible attackers in the same trust domain as the AI agent will be able to change the value of all involved learning data attributes, while potential attackers in other trust domains will only be able to influence a limited set of features, in the lack of data poisoning countermeasures.

In further detail, we propose a framework of managing multiple cross-domain (e.g., cross- Mobile Network Operator (MNO), cross-geographical region) AI service consumer (e.g., data contributors, AI agents) trust levels (i.e., from globally trusted in all network deployments to globally untrusted) taking into account the AI service consumer's data privacy limitations set within each security domain. Solution components of the proposed framework are the following:

1. Each trust level is proposed to be separated from others through a specific “NEF of Level X” see Figure 3-13 below. Each AI agent, regardless of being deployed at a device or at network infrastructure, can access private/ secure user device data with no additional authorisation only in case these user devices providing such data are of the same trust level as the AI agent.
2. An interoperable AI Information Service (AIS) with an AI API consumed over an open network interface - when consumed by, e.g., an AI agent needing to train/ update its ML model, advertises the request and, based on responses, it prioritises data acquisition coming from providers (UEs, machines, other AI agents) of widest cross-domain trust level and most relaxed data privacy constraints within the concerned domains to mitigate biasing of AI/ML-based decisions that would be otherwise taken based only on data originating from providers of the same trust level.
3. To avoid cases of, for example, insufficient data acquisition, leading to AI agent unavailability (or, poor inferencing performance), it is proposed to introduce a “generalisation score” of the trained/ updated ML model indicating how many trust domains a learning data point has traversed to be considered for ML model training. This score can be calculated by introducing a simple counter for each data point, increased each time the data point enters a different trust domain. This score value per (re)-trained model may, for example, be first calculated per input feature and then, as a compound value for the trained ML model, possibly with different weighting factors across the input features. Based on this score, it will, thus, be up to a calling AI service consumer to use or not the trained (originally or updated) model for needed decisions; for example, an AI service consumer interested obtaining e.g., QoS predictions focused on a limited topology and specific times of day may be sufficient to consume a model trained using only local data. This introduces a level of training bias, which may be, however, acceptable, depending on the task. Of course, this score is limited by the capabilities different trust domains have in discovering an AI agent and receive a call for ML model updating, in other words, a "perimeter of trust" defined by means of an authorisation policy.

Figure 3-13 illustrates the solution proposal. For example, within a "Level-1 Trusted Domain", an AF, such as an AI agent instantiated within this domain, is allowed to request and acquire all available data by user devices being also local to the same domain, even those indicated as "private" or "confidential" without the need to provide additional authorisation credentials. Acquisition of private data external to a given trust level the AI agent is part of will only be possible upon providing additional authorisation credentials. This means that, for a learning data point to traverse across N trust domains, N-1 authentication and authorisation procedures will need to be implemented, including the filtering out of data attributes (referring to specific features) that are considered as domain-private and, therefore, non-shareable. This approach, although involving heavier device control signalling may be better applicable to cases where a device frequently changes its network attachment.

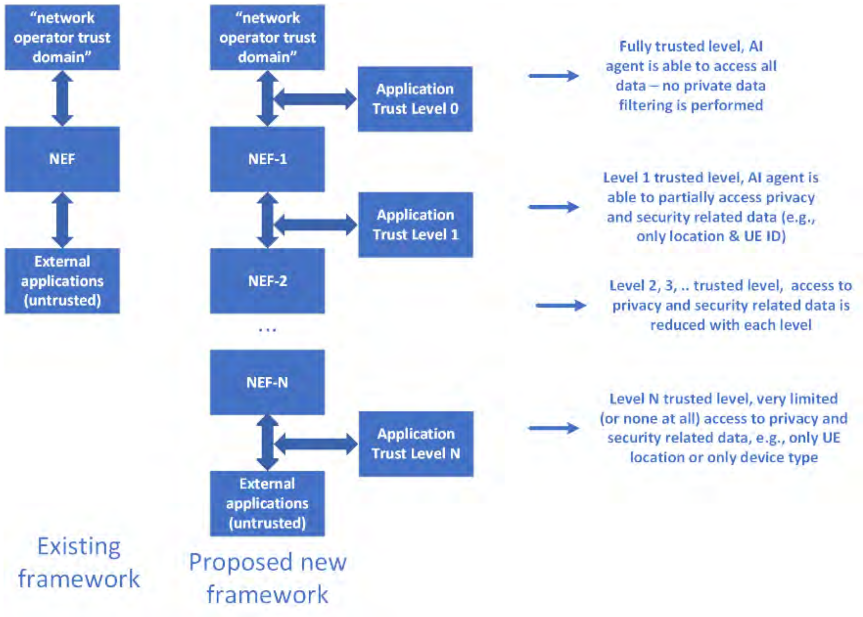

"network operator trust domain"
NEF
External applications (untrusted)
Existing framework
"network operator trust domain"
NEF-1
NEF-2
...
NEF-N
External applications (untrusted)
Proposed new framework
Application Trust Level 0
Application Trust Level 1
Application Trust Level N
Fully trusted level, AI agent is able to access all data – no private data filtering is performed
Level 1 trusted level, AI agent is able to partially access privacy and security related data (e.g., only location & UE ID)
Level 2, 3, .. trusted level, access to privacy and security related data is reduced with each level
Level N trusted level, very limited (or none at all) access to privacy and security related data, e.g., only UE location or only device type

Figure 3-13: Defining a "NEF of Level X" per trust level.

Figure 3-14 explains the principle of the proposed cross-domain trust level management approach when data/ model resource (i.e., data contributor and AI agent containing a model) location is considered as a contextual criterion of implementing data ingress/ egress filtering per a predefined policy at each trust level. In this case, a given NEF is considered as a "gatekeeper" entity implementing that policy. The Policy Decision Point (PDP) could either be an over-the-top entity, such as the NRF, collecting data privacy policy updates by all corresponding NEFs and authorizing incoming cross-domain data transfer requests or each NEF on its own. In the second case, this would call for NEF-to-NEF exchange of data privacy policy updates in a "partner-to-partner" network fashion, thereby queuing up cross-domain data transfer requests during policy synchronisation phase. In upcoming deliverable D5.3, more details can be provided on how the proposed approach translates to the AI Service and its API.

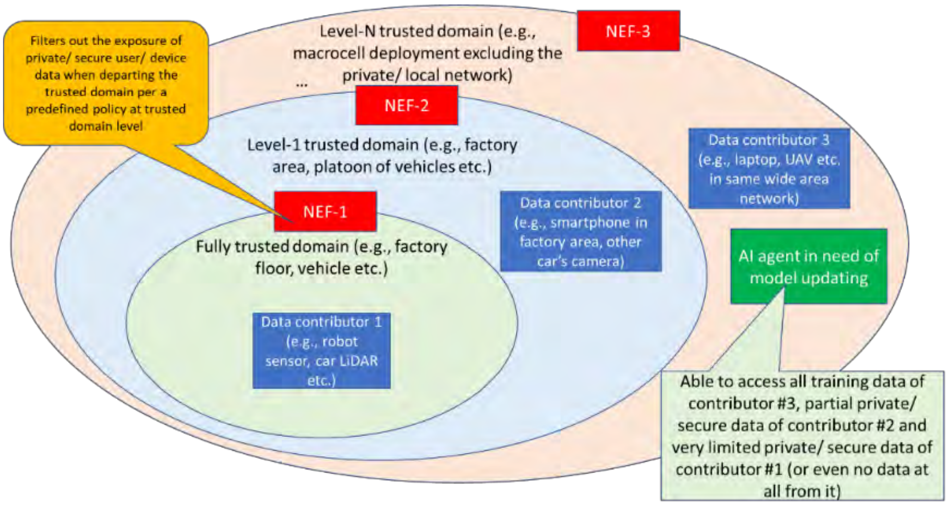

Filters out the exposure of private/ secure user/ device data when departing the trusted domain per a predefined policy at trusted domain level
Level-N trusted domain (e.g., macrocell deployment excluding the private/ local network)
...
NEF-3
NEF-2
Level-1 trusted domain (e.g., factory area, platoon of vehicles etc.)
NEF-1
Fully trusted domain (e.g., factory floor, vehicle etc.)
Data contributor 1 (e.g., robot sensor, car LiDAR etc.)
Data contributor 2 (e.g., smartphone in factory area, other car's camera)
Data contributor 3 (e.g., laptop, UAV etc. in same wide area network)
AI agent in need of model updating
Able to access all training data of contributor #3, partial private/ secure data of contributor #2 and very limited private/ secure data of contributor #1 (or even no data at all from it)

Figure 3-14: Learning data acquisition in a cross-domain, multi-trust level network environment.

A possible alternative solution component is based on Zero-Trust Architecture, as defined by the National Institute of Standards and Technology (NIST) SP800-207 [SP800-207]. In this solution, communication partners (e.g., network operators) authenticate each other, and all communication is secured regardless of the network location. This approach may be applicable to some cases where inter-network data consumption for ML model training purposes is expected to be occurring frequently with the data contributing devices not moving frequently across networks. A Policy Enforcement Point (PEP) performs AI service consumer request access control and proposes a trust level per AI task request based on policies provided by a PDP, observable state of client identity, AI task type, requested AI resources, location information, other behavioural and environmental attributes. Data anonymisation is performed by the data producer itself based on the proposed trust level. An overview of the alternative solution appears in Figure 9-2.

3.3.2 Developing interfaces for predictive orchestration and supporting execution agents/modules

New APIs will be developed to communicate within and across different administrative domains, in order to expose services from network elements in different architectural layers. Therefore, following a unified pattern, the interaction and communication between layers and network elements can be enabled, e.g., using access control policies.

In a single administrative domain, the resource consumption is regulated as in a service mesh topology, spanning across layers. A service mesh can be defined as, a dedicated infrastructure layer for managing service-to-service communication over microservices, without imposing any modification on how the service is implemented [LLG+19]. In a multi-domain scenario, the resources and functions are not anymore belonging to just one individual; instead, cooperation between different administrative domains is required to consume APIs exposed by other domains. Therefore, there is a need to unify the exposure and management of a variety of APIs from multiple domains, this view is aligned with the open-source project "CAMARA – The Telco Global API Alliance" [CAMARA22] launched by the Global System for Mobile Communications Association (GSMA) association and the Linux Foundation, to hide telco complexity and develop open, global, and accessible API solution to operator capabilities, regardless the customer's domain is in [CAMARA22a]. Also, depending on what is required, the granularity provided by the APIs needs to be fine-tuned depending on the entity and, dynamic discovery APIs are required since, functions and resources can be added or removed at runtime.

Therefore, 6G orchestration frameworks must be able to optimally select the most suitable interface in the architecture in a dynamic manner considering each technology's infrastructure topology, capability, and availability. This is important because the current interface selection frameworks (through network slice templates and service descriptors) are static and infrastructure agnostic. Hence, the instantiation of the services should be customisable, allowing the automatic adaptation to the resources, functions and network capabilities, as well as, triggered by client/consumer demand.

In addition, 6G orchestration must be able to handle combinations of service definitions using diverse software modules, including also VMs or containers and serverless functions, which is important to be able to provide fine-grained services within the network continuum. Also, extreme edge devices need to be considered as part of the 6G network slicing mechanisms to ensure that E2E orchestration is provided, meaning that, the 6G orchestration will be able to aggregate the required resources from the different verticals, maintaining at the same time a proper isolation between them.

Regarding the extreme edge previously mentioned, the integration of it is not trivial, because the environment includes a variety of devices, which would be very volatile and heterogeneous. One of the challenges is to decide which orchestration actions must be performed and when to perform them, for example, AI/ML techniques can be useful to predict scaling actions in order to meet the networks' traffic demands or proactively take function placement decisions in any domain (core, edge or extreme edge). Additionally, AI/ML techniques could interface with other functions to provide a proactive alerting system. Such proactive orchestration actions must be supported by metrics gathered from infrastructure and application, e.g., control and data plane metrics, correlating data from different domains.

3.3.3 Integrated and distributed AI with supporting protocols

The usage of AI/ML in 6G networks could be present in every segment along the whole infrastructure [LSL+22], therefore, the inclusion of AI in Managed and Managing Objects could be required. A Managed Object (MO) can be a NF, Network Service (NS) or Network Slice among others, it is explained in detail in [HEX-D62]. For instance, an AI software element interacting as MO in the deployment of a NS, would involve actions at RAN and CN side, as well as management functions (Managing Objects). When deploying a NS, different phases are involved (preparation, instantiation, configuration and activation), during the configuration phase, the orchestrator has to allocate as efficient as possible the available network resources.

In a distributed AI, multiple parties are involved, and security must be guaranteed by means of protocols [YLC+19], e.g., the three-phase commit protocol (3PC) is a distributed algorithm that can be applied in a distributed system letting all nodes agree to commit a transaction or secure multi-party computation (MPC) computing a function over their inputs while keeping those inputs private.

The AI Orchestration function mentioned in Section 3.1.3 and 3.2.3 is aligned with the AI/ML Function Block described in [HEX-D62], are providing the mechanisms to build the intelligence to optimise, manage and control the services to be or already deployed and to take decisions regarding which actions to perform at different architectural layers. AI/ML functions can be applied also in other Functions Blocks described in [HEX-D62], having AI/ML components distributed across the network and be managed from the AI/ML Functions Blocks. In WP4 [HEX-D42], the AI/ML components and their distributed functionalities are discussed, i.e., AI agents can be distributed across the network in large deployments, therefore AI/ML techniques such as FL [AIA21] can be considered to handle such number of AI agents' population.

An advanced monitoring system is required for the AI/ML components distributed over the network to, enabling the aggregation, collection and dispatch of data, to provide telemetry, monitory and manage of data ingestion from every managed network segment (from extreme edge up to the central cloud). Such monitoring system will allow the integration of, infrastructure, data and CP from different sources to applications.

Application-based monitoring makes possible to perform advanced M&O actions in vertical services by making use of custom metrics defined by verticals or NS supplier, also integrated with the standard fixed set of metrics. This integration infers, to define data aggregators to collect data from any source and format to manage the flow stream from data collection to data consumption.

This heterogeneous mixing of application and infrastructure metrics should be supported by AI/ML techniques, such as forecasting (time series), finding complex hidden relationships among the large amount of data [XHH+21] [KRV+22].

Artificial Intelligence Operations (AIOps) [DLH19], e.g., event noise reduction, applying ML to high volumes of operational data to spot patterns, suppress and identify events that come near to ordinary, ensuring just critical alerts are spotted, or intent based networking [VBT+22] [GKT21] to allow end-users to configure services by using a formal language which will be later on transformed in actions. Larger datasets result in better decisions for AI/ML models.

The AI/ML components deployed and distributed across the network, are able to take better decisions based on the data or metrics collected by the advanced monitoring system, e.g., the deployment of a NS, is a decision that implies actions at RAN, Transport Network and Core segments, this decision is taken upon changes in the network (i.e., losing QoS), or to achieve specific latency or throughput for specific use cases (e.g., automotive or manufacturing).

The deployment of a network slice implies a collection of actions at different layers and domains, some of these actions are not only related to network performance, but also energy efficiency related among others. This can lead to a hidden cross-relationship where AI/ML protocols can help. Besides, forecasting to deploy a NS can be applied [ZZC20] [SSC+17].

3.3.4 Protocol considerations of AI for mobility management

The utilisation of AI/ML has been accelerated to enhance radio access networks. In fact, in release 17, one of the study items was to identify suitable use cases and corresponding AI/ML-based solutions for RAN [37.817]. The release 18 also has an AI/ML scope [Qua21] to investigate its utilisation for air interface for beam management, Channel State Information feedback, and positioning to identify defining stages of AI/ML, performance, and overhead trade off and specification impact to standardisation. A design principle in going forward with AI/ML topics in 5G evolution is that any transfer of information between two separate nodes is very limited and, for instance, AI/ML models are not transferred among different entities. Although this principle simplifies the introduction of AI/ML into RAN, it also limits the impact of AI/ML. To take full advantage of AI for 6G, a different approach can be designed to enable the AI/ML concept over the air interface to be transferrable. The transferrable AI/ML will allow to take full advantage of AI/ML functionality at both gNB and UE by enabling transfer of AI/ML related information (e.g., training/inference models and algorithms) across UE and gNB.

We explain what is needed protocol wise through an example use case, where AI/ML is used to address mobility related procedures by taking advantage of an AI/ML engine at UE side, as shown in Figure 3-15. Specifically, we assume that the UE is provided with AI/ML models from the network and performs layer-3 related quantity predictions. The predictions are used to make mobility related procedures more intelligent both at network and UE side. The AI/ML models use locally available information at UE side, such as measurements, various timers, and counters to produce the predictions. In this example use case, we focus on the protocol considerations for transferrable AI, rather than on the actual analysis of how to realise such use case, needed AI models, etc.

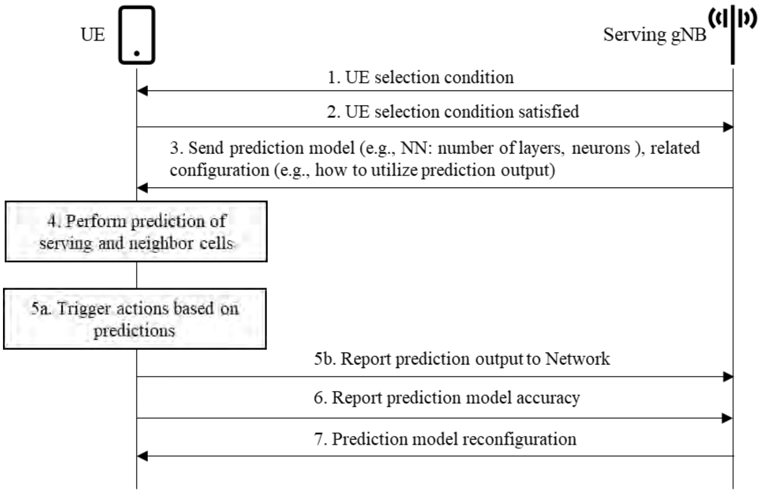

UE
Serving gNB
1. UE selection condition
2. UE selection condition satisfied
3. Send prediction model (e.g., NN: number of layers, neurons), related configuration (e.g., how to utilize prediction output)
4. Perform prediction of serving and neighbor cells
5a. Trigger actions based on predictions
5b. Report prediction output to Network
6. Report prediction model accuracy
7. Prediction model reconfiguration

Figure 3-15: Example signalling diagram of the AI concept for mobility procedures.

Aspect 1: UE selection feature

For efficient utilisation of AI across the UE and gNB, a protocol feature is required so that most suitable UEs can be identified to be used in an AI procedure. The suitability depends on several aspects, including, e.g., UE hardware such as processing and memory, battery status, as well as aspects, such as the quality of the data that a UE possesses, or it can produce. For example, the network could be interested in training a mobility prediction model specific to Radio Link Failure (RLF). Then it can construct a test condition for the RLF (such as a minimum number of observed RLF events in a given period) and send it to a UE where the UE tests the condition and reports the outcome, as shown in Figure 3-15 (steps 1, 2). Based on this feature, the network will select only those UEs that have the potential of being used in training or inferencing RLF prediction models and hence increase their efficiency (e.g., UEs which didn’t observe RLF events will not be selected).

Aspect 2: Convey information related to AI

Another aspect is to provide a means to convey information that is needed or produced by the AI to/from the intended entity. In general, one would require specific procedures to be able to send information regarding the structure of the AI model (e.g., a deep neural network with certain number of layers and neuron) the associated inputs/outputs of the models at UE side to the network and vice versa. In the mobility use case this aspect is emphasised by UE receiving a prediction model from the gNB (step 3 Figure 3-15) and reporting the predictions to the gNB (step 5b in Figure 3-15). The network, by receiving the measurement result as well as corresponding measurement predictions can make intelligent decision regarding mobility procedures. A typical measurement example is shown in Figure 3-16 (right), wherein neighbour cell becomes better than serving cell in terms of RSRP. This triggers a measurement report to the network (referred to as an A3 event in 3GPP [38.331]). Although at time of measurement report the neighbour cell is more favourable, in future instance this may not hold true impacting the success of the network decision regarding Handover (HO). By including the measurement predictions, the network can identify the best course of action (Figure 3-16, left). The predictions of RSRP values indicate at time t0 (current time) the event A3 will be satisfied. This information triggers a measurement report to the network to inform it about the radio conditions in an earlier time.

When the network receives the measurement report triggered by A3, typically a HO command is sent to the UE to change the connection to the better cell. However, based on various reasons, such as threshold mistuning, and UE mobility, the reported neighbour cell may not be the best and too soon/late HO, RLF and ping pong HO may occur. The predictions included (e.g., in the measurement report in Radio Resource Control (RRC) protocol) serve to help identify those scenarios as explained in Figure 3-16.

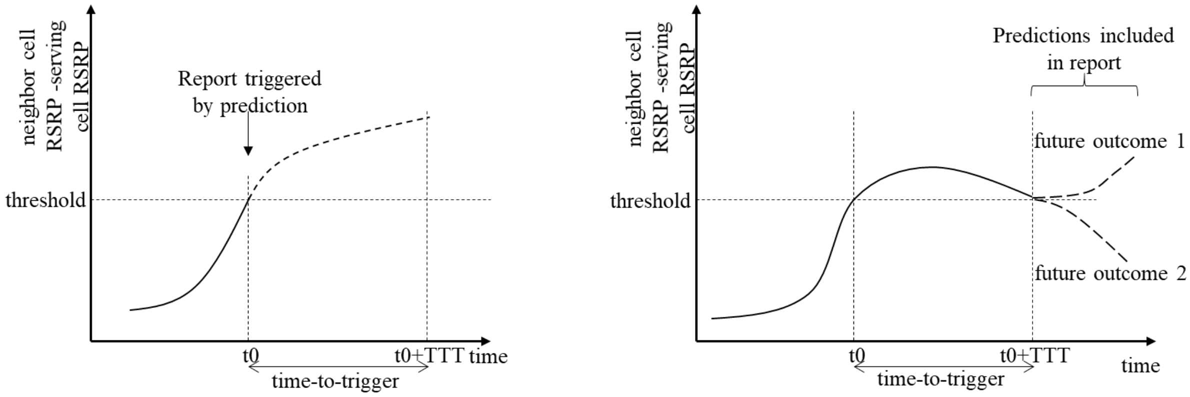


Figure 3-16: RSRP measurement predictions for identifying course of action (left), report outcome to network (right)

Aspect 3: Executing actions based on AI

Beyond transfer of AI models, input/output information, etc., another aspect is required to address how the AI should be used to execute a certain task. The AI model can implement a functionality on its own or be used to generate input to the existing procedures. In the mobility use case, the output of AI/ML models is utilised as an additional input to trigger various procedures. Various measurement events such as A1, …, A6 [38.331] have been standardised to trigger measurement reports when certain conditions regarding serving/neighbour cells are satisfied. Those events are parameterised to optimise the reporting for various scenarios. Here, the prediction can be used as an additional input feature to further optimise the measurement report. For example, as shown in Figure 3-16 (right) the predictions of radio condition of serving and neighbour cells can identify whether the conditions of an event will be satisfied in the future and if so then measurement report can be triggered without any delay, as introduced by time-to-trigger, resulting in swift actions by the network. By defining the corresponding procedures in the RRC protocol (e.g., triggers based on outputs of AI models) this protocol aspect can be addressed for this specific use case.

Aspect 4: Life cycle management

The AI code implementing the training/inference part is only a small part of the AI system which contains various components such as data collection/validation/processing, model training/validation and monitoring to operate. Specific protocol features are required to properly address the deployment and management of AI components. Demonstrating, as an example, in the mobility use case, the UE monitors the predictions (e.g., as specified by RRC) and calculates accuracy of the model and reports this to gNB. Consequently, if the gNB detects deteriorated performance then it can send a new prediction model to the UE as shown in Figure 3-15 (step 6 and 7, respectively).

The widespread use of AI for the air interface is expected in 6G. Hence other components of the air interface will need protocol level features to efficiently utilize the AI. Although some protocol level needs such as data collection/processing/report are common among all air interface functions there could be specific requirements unique to certain functions (such as strict latency). Thus, to enable utilization of the AI to the air interface, a general protocol framework is needed to collect common features in a single entity to eliminate redundancy while offer tailored features for specific/unique requirements of individual functionalities.

3.4 Programmability

Programmability is the capability of a device or a network to accept a new set of instructions that may alter the device or network behaviour [BJ15]. It could be applicable at the control and management planes, as well as at the data plane. For example, we can change the function running on a programmable device from performing IPv4 routing to any other function simply by loading a new NF program on the programmable networking device. This enables agile updating of functionalities executed on the UE and network side when programmability on these two ends is enabled. Currently, there are different technologies and programming abstractions that enable programming networks and end-hosts. In the following, we elaborate on network programmability advancements and then, we discuss UE programmability concepts.

3.4.1 Network programmability

We select the P4 language [BDG+14] as an exemplary programming abstraction to demonstrate the gains that can be achieved when adopting programmability into networks.

P4 language is one of the initiatives that abstracts the packet processing pipeline of networking devices, while offering great flexibility in defining its behaviour. P4 is also platform-independent so that it can be used to program different classes of packet processors such as software switches, SmartNICs (NIC stands for Network Interface Card), Field Programmable Gate Arrays (FPGAs), or Application-Specific

Integrated Circuit (ASIC) programmable switches, which can be used to form the substrate of cellular infrastructure data centres.

3.4.1.1 Performance-Aware Management of P4-based Programmable Networks

It is important to understand and guarantee certain forwarding performance levels when using programmable networks. This can be achieved using models that can predict the packet forwarding latency when running arbitrary P4 programs on any P4 device as we previously proposed in [HJH+21]. This model makes use of an extensive measurement campaign that identifies the base processing delay of different P4 devices, as well as the marginal delay of executing atomic P4 operations on these devices. These measurements are used to create performance profiles for different types of packet processors. On the other hand, any given P4 program can be decomposed to its atomic operations whose processing latency can be retrieved from the predeveloped performance profiles. For example, these atomic operations can be parsing and manipulating Ethernet and IPv4 headers in the case of IPv4 Forwarding NF. This way, the performance model predicts the packet forwarding latency of running arbitrary NFs on any P4 device as the summation of the base processing delay and the marginal latency cost of executing the constituting atomic P4 constructs. The model is tested for different NFs and showed sub-microsecond precision.

Towards optimising the management of the programmable networks, the optimal placement of NF workload into a heterogeneous programmable substrate should be investigated. In [HHH+21], we propose a mathematical formulation for an optimisation problem that targets placing different NFs into P4-based cloud environments. The orchestrator can use the previously described performance model to conduct the placement with **performance awareness.** It permits achieving the highest performance levels in managing these programmable networks and satisfying QoS requirements.

The problem formulation recognises the different capabilities and characteristics of the hosting P4 devices. These device characteristics include the supported P4 architecture, the supported extern functions (i.e., non-native P4 functions such as encryption), the throughput capacity, the base process delay, the marginal delay for executing different P4 constructs, the delay to execute extern functions, the available processing resources, and the cost.

In addition, the problem formulation considers the requirements of the NF workload in terms of required throughput, required P4 architecture, required extern functions, and the constituting P4 constructs.

For a given NF workload, the objective function searches for the optimal set of P4 devices and the optimal placement of NFs into these devices, while minimising both the total forwarding delay in the system, as well as the capital expenditures cost for building the system. The predeveloped performance model is used to calculate a priori the resultant delay of different placement options as shown in the following equation:

$$\Delta_d^f = \delta_d^{BP} + \sum_{c \in C_f} \delta_d^c + \sum_{e \in E_f} \delta_d^e$$

Where Δ_d^f stand for the forwarding delay when running NF f on P4 device d. This is equal to the summation of three components: (i) the base processing delay on P4 device d, denoted as δ_d^{BP}; (ii) the summation of the marginal delay of running the atomic P4 constructs $c \in C_f$ that constitute NF f on P4 device d denoted by δ_d^c; (iii) the summation of the delays for running extern functions $e \in E_f$ required by NF f denoted by δ_d^e.

A wide set of constraints should be satisfied. These include: (i) an NF should only run on a P4 device that support a compatible P4 architecture and include all the required extern functions by the NF; (ii) the cumulative throughput required by different NFs to be placed on a device should never exceed the limited throughput of that device; (iii) the processing resources capacity of a P4 device should never be exceeded; (iv) some NFs like the Layer 3 Forwarding NF must be placed on every used packet processor to guarantee proper packet forwarding between devices, etc.

For evaluating the proposed workflow, we consider a use case, where a mixture of distinct types of P4 devices is allocated to build the programmable substrate of a cloud environment. These devices can belong to CPU, Network Processing Unit (NPU), FPGA, or ASIC processing platforms. The performance and characteristics of these devices are surveyed from literature, and they are summarised in [HHH+21]. Figure 3-17 depicts a web chart that presents the comparative advantage of different P4 device types in terms of different characteristics. A set of realistic NFs such as IPv4 forwarding, load balancer, firewall, etc. are used as the workload to be processed by the network. More details related to the requirements of the NFs and the characteristics of the P4 devices can be found in [HHH+21].

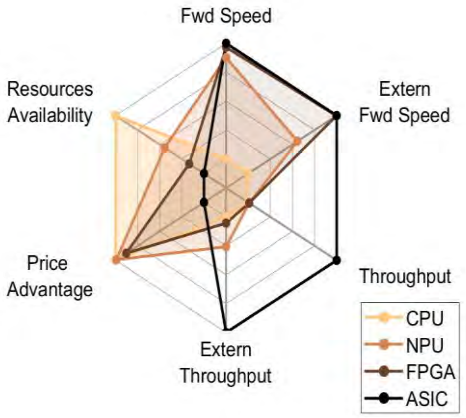


Figure 3-17: Comparative advantage of P4 device types.

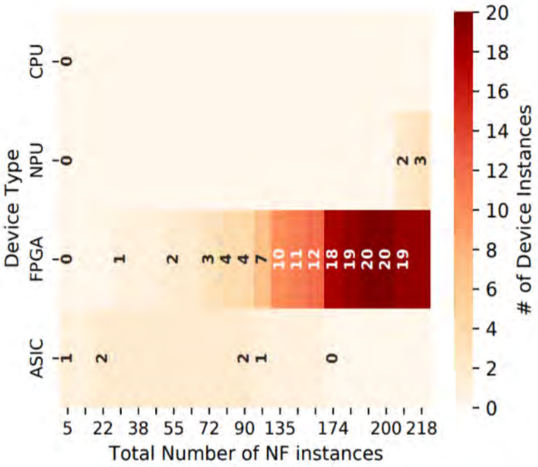


Figure 3-18: Used P4 devices in Scenario 4 [HHH+21].

Four scenarios are evaluated wherein the weights of the two objective functions defined earlier (i.e., the forwarding delay in the system and the cost for building the system) are varied. Scenario 1 targets achieving the best performance without worrying about costs, while Scenario 2 targets finding the cheapest solution where Best-Effort performance is tolerable. Scenario 3 targets achieving a balanced solution with the best performance and minimum costs. Finally, Scenario 4 targets achieving the best performance with a predefined limit on the available budget not to exceed $100k.

The placement solution for Scenarios 1, 2, and 3 are trivial where the solution favoured to use an increasing number of the same type of devices when the workload increases. In Scenario 1, the ASIC devices that have the highest performance were selected, while in Scenario 2 the CPU devices were selected since they are the cheapest. In Scenario 3, the NPU-based devices were selected as they achieve the best balance between performance and cost. The placement solution of Scenario 4, where a cost limit of $100k is pre-set, is more interesting to analyse. The placement result of this scenario is depicted in Figure 3-18 along with the number of instances of each device type required for an increasing number of NFs to be placed. It can be seen that for low workload, one ASIC device is chosen because it provides the best performance, while remaining affordable given the cost limit. Then, another ASIC device is used when up to 22 NFs must be placed, where the second device is required because the first device's processing resources are exhausted. After this point, no more ASIC devices could be used because the remaining budget only allows for the second-best performing device, which is an FPGA. In this case, up to four FPGA devices are required to process the additional NFs until the total number of NFs reaches 90. Following this point, one ASIC device is sacrificed to afford more FPGA devices capable of handling the increased workload. At 174 NFs, the second ASIC device is also replaced with more FPGAs whose number increases to 20 FPGAs when the NF workload reaches 200. Following this, the optimal solution sacrifices performance even further by replacing FPGA devices with the next best performant device, i.e., NPUs, to support processing all NFs within the available budget.

Figure 3-19 and Figure 3-20 depict the delay and cost functions of the optimal solution for various scenarios as a function of the number of NFs to be placed. As expected, the overall delay in Scenario 1 is the shortest, while the overall cost in Scenario 2 is the smallest. The results for Scenario 3 show the trade-off between the two objectives, where the overall delay and cost are both minimised. The Scenario 4 results show that the system's delay is as low as that of Scenario 1 (when only the delay is optimised) until the budget constraint is reached after 22 NFs. After this point, the system's delay begins to increase in comparison to Scenario 1, while the cost is always less than the pre-set budget of $100k.

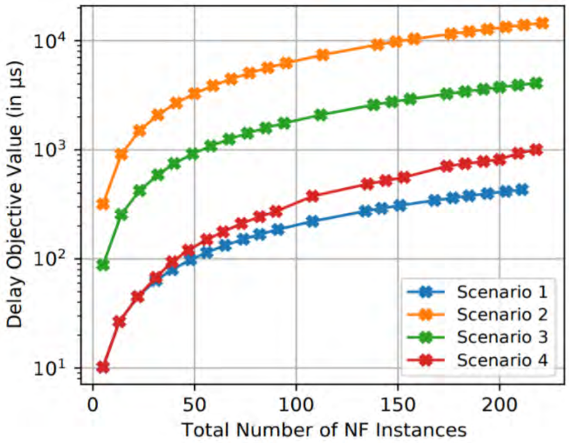


Figure 3-19: Total forwarding delay in the system [HHH+21].

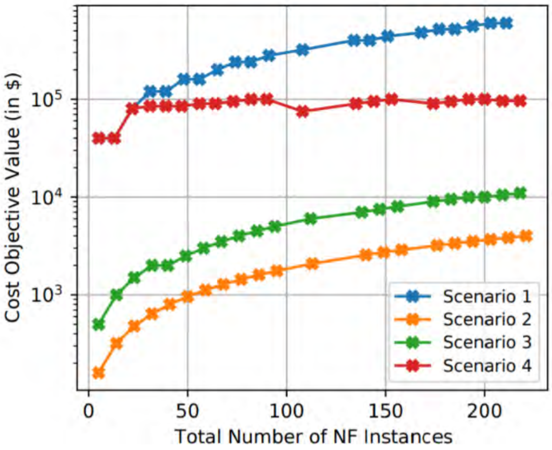


Figure 3-20: Total cost of the system [HHH+21].

In conclusion, the adoption of accurate performance models for programmable networking devices and intelligent management schemes paves the way toward more flexible networks without compromising performance. Moreover, properly considering the capabilities and costs of the network substrate enables cost-efficient planning of the infrastructure according to the expected processing workload towards reducing the Total Cost of Ownership (TCO) of the system.

3.4.1.2 E2E Network programmability and Closed Loop (CL) control

CL control (see [FMR+20]) allows for more efficient resource usage and operation resilience. E2E network programmability at runtime is one building block for achieving this.

E2E network programmability (e.g., based on P4) does not only address packet forwarding policies, but also fine-grained telemetry by modifying packets to include information about the packet provenance (e.g., the path they took, the rules they followed, the delays they encountered, etc.).

By monitoring the network state and validation against packet telemetry it is possible to detect errors and perform needed corrections within milliseconds.

In Figure 3-21 an architecture is depicted supporting programmability with fine-grained telemetry and Intent-based CL control allowing the network owner to specify top-level behaviour as a program that is compiled to code for the SDN controller, switch and host OSs, and data planes. Telemetry data is collected and used for runtime verification and closed-loop control:

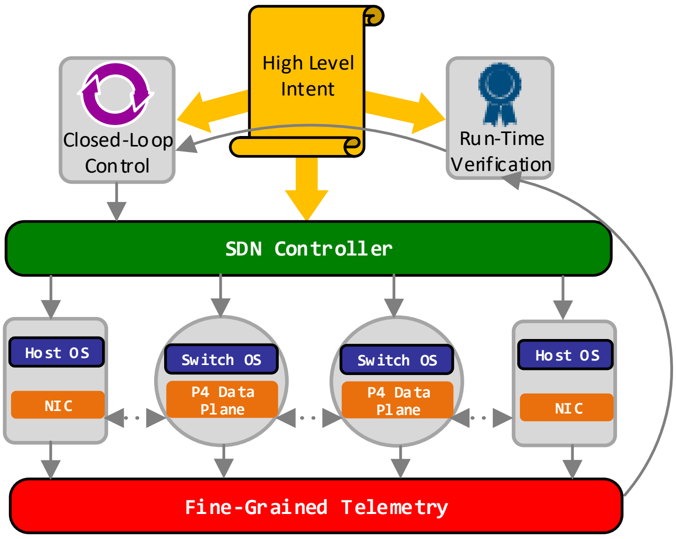


Figure 3-21: Programmability with fine-grained telemetry and Intent-based CL control [FMR+20].

The architecture illustrated in Figure 3-21 and defined in [FMR+20] consists of the following elements:

- **Data Plane** consists of programmable switches and/or SmartNICs typically implemented in the form of an FPGA supporting packet processing at line rate with the ability to react to changes in local state and to update the forwarding behaviour instantaneously.
- **Switch & Host Operating System (OS)** implements the supported routing protocols for IP networks (e.g., BGP, Open Shortest Path First (OSPF), P4Runtime, etc.), the related algorithms, and the aggregation of measurement data collected from the data plane.
- **SDN controller** maintains a networkwide view of the current topology and operating conditions and runs sophisticated algorithms that would be difficult to express as a distributed computation.

Already today's 5G System (5GS) supports several CL control use cases in the data plane. UP programmability with fine-grained telemetry as introduced above could further improve the network performance and efficiency. Use cases of interest are:

- Traffic Load-Balancing control when Multi-Access Protocol Data Unit (PDU) Sessions are used supporting different access networks, including untrusted and trusted non-3GPP access networks, wireline 5G access networks, etc.
- Traffic Load-Balancing control when redundant UP paths with Dual Connectivity (DC) is used.
- Traffic Load-Balancing control when redundant transmission on N3/N9 interfaces is used.
- Control of Ultra-Reliable Low-Latency Communication (URLLC) Services.
- Decision for relocation of the UPF when acting as the PDU Session Anchor.

In 6G we are proposing to support E2E programmability of the data plane with the involved UP nodes (e.g., based on P4) and the UE (controlled via Non-Access Stratum (NAS) protocol).

3.4.2 UE programmability

The standardization process in 3GPP has proven its paramount value for numerous successful generations of cellular networks. However, the 3GPP process can be rather time consuming in some areas. One way to decrease the time for innovation can be to allow programmability in 6G. The concept here concerns defining API(s) for the UE associated to actions/routines/sub-routines that can be exposed to a programmer entity, so that a programmer entity is able to modify/add one or multiple behaviour(s) at the UE associated to the air interface protocols. A high-level signalling diagram and architecture is shown in Figure 3-22.

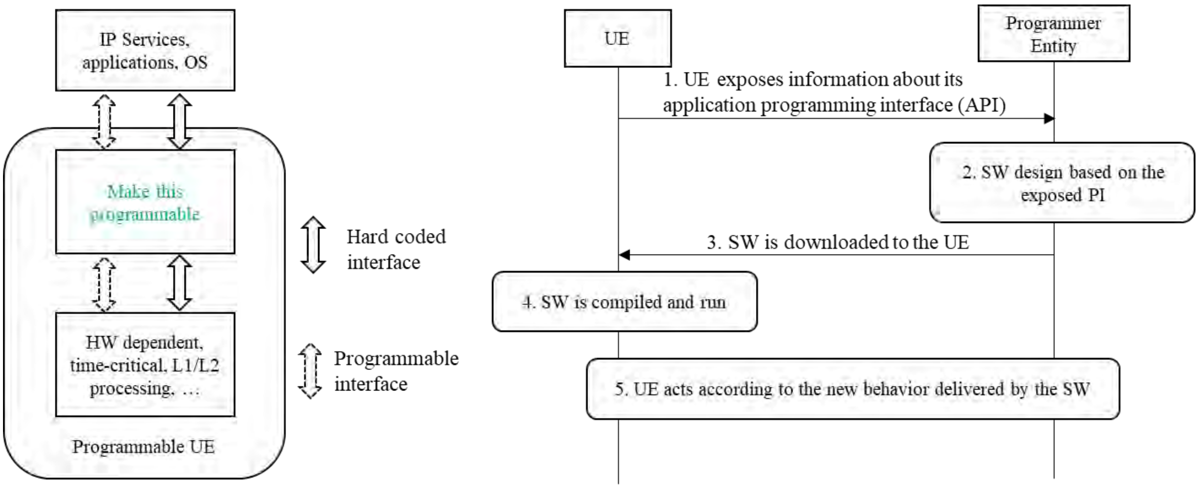

IP Services, applications, OS
Make this programmable
HW dependent, time-critical, L1/L2 processing, ...
Programmable UE
Hard coded interface
Programmable interface
UE
Programmer Entity
1. UE exposes information about its application programming interface (API)
2. SW design based on the exposed PI
3. SW is downloaded to the UE
4. SW is compiled and run
5. UE acts according to the new behavior delivered by the SW

Figure 3-22: High level architecture of a programmable UE (left) and signalling diagram for programmability (right).

As a result, with a reduced amount of 3GPP standardisation, the programmer entity should be able to define new behaviours/ features for the programmed UE, such as a new message received or transmitted, new information elements and associated interpretation, new reports, new trigger for that message or report, and new/additional triggers for existing messages.

At a high level, some initial components are essential to enable UE programmability. Those components can be considered as high-level solutions that are needed initially to realise the concept. Here, we provide three of those which are API exposure towards network, initial access mechanism for programmable UEs, and software version management.

The API exposure mechanism is needed so that the programmer entity knows the specific capabilities of programmable UEs in order to utilize their capability. Hence, there needs to be a signalling procedure to convey the programmable capabilities and their specifics to the programmer entity. Such capability exposure can be initiated at specific events such as UE registering with the network or can be on-demand using dedicated signalling towards specific UEs. The capabilities regarding programmability can be standardised to include specific APIs and UEs can indicate which subset of standardised APIs are supported. Therefore, the programmer entity receiving the capability can design suitable SW for specific UEs based on available APIs.

Upon initial access, programmable UEs may have a SW version for a specific behaviour which is different from the one at network side. Hence the UE may not be able to connect to the network because of SW incompatibility. A bootstrapping broadcast channel can potentially solve this problem by providing information on the current version of the SW at network side and how to acquire it. A programmable UE upon initial access first checks the bootstrap channel to acquire information of the active SWs and how to acquire them. Based on such information the UE can download the proper SW version and hence connect to the network.

Finally, SW management mechanisms are needed because for specific behaviour implemented by a SW there could be various versions corresponding to new updates or different interests of operators/vendors. Hence, SW management solution is required to enable synchronised operation of the UE and network with respect to SW versions. Such a solution will serve to store and manage multiple SW versions including adding, removing, and updating SW. Moreover, the solution will allow to select a specific SW version to be initialised upon request from the network. However, there are many challenges to overcome for the successful adoption of the concept which faces manifold open research questions.

One aspect is to ensure that the programmability solution does not disrupt 3GPP. As mentioned before, the 3GPP way of working is vital to success of cellular network evolution and trusted by every player in the ecosystem. The concept must be developed in harmony to the 3GPP and complement it rather than disrupt an already successful framework. This can be achieved by defining an overall framework for UE programmability by defining a bare minimum for the concept in 3GPP, such as methods of downloading a software and defining and exposing APIs.

Another challenge is that there are potentially many flavours of programmability, each with advantages and disadvantages. Each flavour balances a trade-off between its capability and pragmatism. At one extreme edge, one can envision a downloadable UE stack paradigm potentially offering full programmability. However, developing this approach from concept to reality will face many difficulties ranging from technical issues, split of responsibilities and concerns among different entities/vendors, trust and privacy issues and acceptance from 3GPP. On the other hand, the programmability could be introduced in a limited way by allowing only specific features, e.g., Radio Resource Management (RRM) measurement, to be programmed. Such an approach will be more acceptable from the point of pragmatism at the risk of being very limited and potentially leading to introduction of multiple APIs per protocol stack.

Another challenge is related to a typical UE hardware. The UE hardware typically has a small footprint and packed with optimised codes to achieve extreme efficiency in contrast to more flexible hardware such as a VM. Thus, any framework should make an extra effort to accommodate this constraint.

Privacy and security aspects should be considered as fundamental parts of the concept. A security/privacy functionality needs to ensure that UE always receive safe program, the received program does not introduce security concerns, privacy of the UE is never compromised, and UE is implemented with mechanism to ensure trusted computing.

Another challenge is considering the mobility aspects. The programmability framework should not restrict the UE in moving freely in the network. When a new behaviour is programmed to the UE, e.g., by a SW patch, then the framework should ensure that the UE is not interrupted when moving to another part of the network because either that specific SW is not available or have non-compatible versions. This also raises the multivendor issues and the need to properly handle trustworthiness aspects when it comes to code exposure.

Beyond the high-level solutions presented here, enabling the realisation of the concept in a general framework, a next step is to develop a concrete architecture for the UE programmability and specific use cases that it can realise for a programmable configuration of the air interface.

3.5 Dynamic Function Placement (DFP)

In [HEX-D51] DFP was described as the concept of deploying functions to orchestrate differentiated services optimally across multiple sites and clouds based on diverse intents and policy constraints of dynamically changing environments. Following subsections contain high-level proposal for hierarchical 2-layer orchestration and DFP solution principles aiming to tackle the challenges deriving from multi-domain environment.

3.5.1 Orchestration in multi-domain environment

SBA based multi-domain orchestration is depicted in Figure 3-23, where each domain has its own SBA functionality, such as NRF and Service Communication Proxy (SCP), and Network Service Mesh (NSM) enabled inter-domain connectivity with networking policy support. These functionalities together with NSM are used in the creation of a uniform resourcing space, where NFs are managed over domain boundaries in harmonised fashion. This management can be seen as hierarchical orchestration with two separate levels, where domain internal dynamicity is not fully exposed externally, i.e., multi-domain level, to ensure sufficient information stability required for scalability properties. For LCM, this could mean that the root level management logic that is operating on top of individual domains only decides the best candidate domains for NFs, to be created or moved, and then the final deployment details (inside the domains) are left for domain specific logics that know exactly how resource conditions are and are aware of all the domain internal dynamics.

Like with hierarchical orchestration, other supporting functionalities, such as monitoring and management (e.g., NF scaling), which can be a part of operation set of DFP, should also support this 2-layer approach. Monitoring is naturally something that is done within a domain and then the information is exposed externally as raw information or as some sort of (pre-processed) aggregate. For NF scaling, a domain's internal scaling properties and the scaling over multiple domains should be considered separately:

1) Intra-domain: internal scaling where the serving capacity of the deployed NFs are scaled up/down or in/out based on the local needs or based on the external event like a request from the root level.
2) Multi-domain: overarching scaling needs the utilisation metrics and load predictions from each involved domain to make decisions on requesting more serving capacity from domains or redirecting/refocusing service offerings from one domain to another.

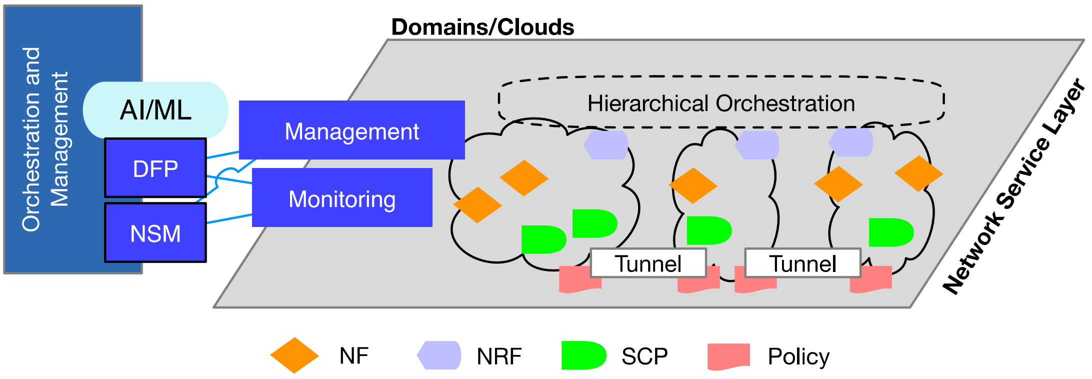


Figure 3-23 : Proposed orchestration approach in multi-domain landscape.

This hierarchy can also be considered from CL perspective. For multi-domain (root) level, naturally CLs over multiple domains are longer, i.e., increased delay than within a domain meaning that their execution is also slower and hence the execution of reactive operations takes longer. There is not necessarily only a single CL covering all underlying domains, thus domains can be further grouped together, and each group can have its own CL, where group members are treated according to uniform CL. This grouping can be done, for instance, based on domain type, e.g., far edge, edge or centralised domains, or based on geographical location so that all domains in certain geographical area are bound together and under the same CL for the same kind of treatment. Moreover, service types can play a role in this grouping as well, making the whole problem domain multi-dimensional. In one extreme use case with the most dynamics, it may be that CLs are also changing over time, but this type of functionality would require the support of high dynamicity in many areas including the related decision-making machineries that are parts of CLs.

Another aspect is that the length of CL correlates directly on how fast the loop can react to changes, e.g., network changes, based on external triggers or local probing. Taking this into consideration, the root level can be suited for predictive decision making in wider scope, i.e., over multiple domains. Respectively, domain internal loops are better for reacting fast by doing some local changes inside the domain or even inside a physical/logical node of the domain. In this kind of scheme, the root level does predictive adjustments in longer lifetime and each underlying domain then operates on that basis and reacts fast with local short-term changes to fix issues temporarily until more permanent changes are propagated from the root level. It is responsibility of each domain to communicate the local changes back to the root level, where the long-term solution ('optimum') can be modified accordingly and the needed management operations in the underlying domains can be initiated. This corresponds to the case of having AI/ML assisted functionalities as part of the root level to help in predicting system behaviour and finding system-wide/global optimum. Additionally, the root level can give to the faster domain specific CLs quotas/budgets/policies of autonomy, which contains frame of references and thresholds when to inform the root level.

In 5G, NRF is the key element of dynamic NF discovery and selection procedure. This procedure can be based on request criteria, such as load of the candidate NF providers. The procedure is clearly intra-domain by nature, because according to the current 5G SBA, there is only a single domain. In single domain, this approach can provide sufficient scaling properties. However, in an envisioned 6G architecture with a multitude of different types of domains, this approach does not scale anymore and the whole service surface for which service discovery provides means to find the preferred destination service (NF) instance needs to be hierarchical. This provides means to hide domain specific dynamics from the root level and works well with indirect communication schemes via SCP where the SCP does the final NF instance selection and hides, for instance, the loads of NF instances behind it. Even in this hierarchical case, NRF could do some pre-filtering based on various criteria resulting that some candidates are already filtered out from the response. Especially, this could be the case of AI/ML assisted NRF, then it could make more sense that AI/ML knowledge is exploited in the NRF as early as possible by selecting the candidate target NF(s) and then communicating it to the consumer. How exactly AI/ML helps in the processing of NF candidates to filter out non-feasible ones, it depends on the used AI/ML algorithm(s) and model(s).

3.5.2 Multi-domain environment implications

Like in case of orchestration, multi-domain environment drives also DFP solutions towards hierarchical 2-layer model, where the DFP decision making is split into two phases; i) multi-domain phase, and ii) domain internal phase. In the 1st phase, there needs to be upper layer "root-DFP" that manages operations on domain level, i.e., on top of multiple domains, and based on these decisions and actions, the 2nd phase is initiated in the selected target domains. During the 2nd phase, the execution of DFP related tasks is done internally inside the domain(s) according to the guidance from the root-DFP. In practise this means that the root-DFP is co-operating with multi-domain orchestration. Respectively, the domain specific DFP updates NRF information to correspond the latest DFP operation results, and with NWDAF/Hexa-X analytics framework in receiving NF related analytics to be used in its decision making and operation execution. Additionally, SCP can be involved in co-operation with NRF, if the related NF(s) are accessible indirectly via SCP, to ensure that the service discovery has the latest knowledge of NF statuses and NFs' reachability.

There are strict timing constraints in the interaction between DFP and NRF to ensure that the results of DFP operations impacting service availability and resourcing are updated in service discovery. For updates with indirect access via SCP which are not directly visible in NRF can potentially be faster compared to updates in domain wide NRF/registry for direct access updates. Naturally, DFPs in different layers must be interfaced for distributed (multi-step) operations.

Regarding operational side of DFP, multi-domain environment not only challenges the architecture for DFP, but also the ways how NFs are managed in the future. Foreseen evolution of technical topics, such as Network Slicing and UE programmability, drives progress towards end user specific NSs. To be able to support such and even more advanced use cases LCM and DFP need to support coordinated operations over domain borders.

Network Slicing provides resource isolation and adds additional dimension onto the DFP decision making, which tries to find an optimal solution based on the defined criteria like resource type, location, and connectivity. The solution space for such decision making could be narrowed down so that any possible solution is only inside a given NS, or correspondingly the sandboxing of solution space could be wide enough to cover multiple NSs.

Besides scaling, the new type of operations to enable this support are relocation and offloading. Reader should notice that these operations can also be used inside a domain.

While these three operations are quite self-explanatory, it is not always easy to differentiate scaling, relocation and offloading from each other rationally. However, one could argue that multi-domain environment justifies their existence and short definitions are as follows:

- *Scaling* is a “legacy” operation mostly driven by the consumers’ demand and means that the scaling of existing capacity of producers (up/down or in/out) needed to ensure the “optimal” service level of a certain NF. KPIs for “optimal” are obviously here many-folded. Multi-domain environment drives new requirement(s), such as how to support distributed multi-domain transactions for NF scaling coupling together domains’ atomic transactions.
- *Relocation* is operation used to change (physical) location of NF instance(s) in the network topology. The reason for relocating NF instance can be as simple as AI logic or some other management entity suggesting optimisation. Alternatively, it can also derive from possible new 6G service model, where end user specific (far) edge services (with strict latency requirements) need to follow end user in the network topology. In practise, this means that either new NF instance is created or resources from the existing one are taken in to proactively or reactively in the new destination location and context transfer is done, if needed. Alternatively, also relocated VM/container could be copied to new location. In case of relocation the “old NF instance” is deleted after the relocation is finished.
- *Offloading* is “a nephew of relocation” and namely used to distribute workload of an existing NF instance to the new location(s) in the network topology based on policies/KPIs/etc. The typical scenario described in the literature even during 5G era has been offloading certain NFs from the core domain to the (far) edge domain, due to better response to latency requirements. Offloading does not involve the deletion of existing “old NF” instance, but instance in the new location and in the old location do co-exist. However, offloading may require context transfer or context update to the NF instance in the new location.

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

To assess the feasibility of Intelligent Networks, we propose an application for Forecast-based recovery in Real-time remote Control of robotics (FoReCo). Remotely controlled robot manipulators are typically controlled over wireless channels, hence, losses in the wireless medium impact the remote control of the manipulator. Namely, the remote-control commands are lost and do not arrive to the robotic manipulator.

To resolve the problematic situation of losing remote control commands upon network packet loses, we propose to periodically report the network status and commands to the Analytics Framework (Section 3.2.3). Later, the analytics are issued by a distributed AI service (Section 3.2.4) that forecasts the control commands that did not arrive on time due to network interference and congestion problem. The AI service is embedded into a NFV service that is dynamically placed (Section 3.5) on the factory floor Edge premises to meet low latency requirements of remotely controlled robotic services.

3.6.1 The remote control command loss problem

We consider a factory floor site that uses a Digital Twin (DT) of a robotic arm. The Digital Twin is a virtual replica of the robotic arm that is synchronised with its movements in real time, e.g., the Digital Twin replica will open the manipulator hand at the same time as the robotic arm is picking up a box. All the Digital Twin data is reported to the Analytics Framework (Section 3.2) for later use to assess monitoring and operational maintenance of the Digital Twin assisted robot. Moreover, the Digital Twin is dynamically deployed as a NFV service on the factory floor Edge premises – see Section 3.5.

In our scenario, the physical robot is operated either directly (and synchronizing the DT in parallel) or through operation over the DT which is echoed in the real world in real time. Note that the remote operation through the Digital Twin is feasible because of the underlying network. Namely, commands are sent via an IP-based connectivity that communicates both the computing device holding the Digital Twin, and the robotic arm that performs the actions in the real-world scenario. To meet the strict latency constraints of controlled robotics, the Digital Twin is deployed on Edge premises (typically deployed in local edge at the factory premises), and the Network Programmability (Section 3.4.1) finds optimal traffic paths to minimize the Round-Trip Time (RTT) latency. Once the Digital Twin is deployed and has connectivity to the robot, the joystick commands are sent over the network from the Digital Twin to the robotic arm, e.g., moving up the joystick results in a command instructing both the Digital Twin and robotic arm to move up by a given offset.

It is necessary to understand how the joystick operation results into movement instructions for the robotic arm. Whenever the remote operator holds the joystick in a given direction, the joystick drivers do not only generate one movement command, but rather several of them with a frequency around 20ms. That is, holding the joystick to the right during 1sec. results into 50 commands with a fixed offset to the right. If any of those commands get lost in the network, or they are not delivered on time to the real robot, this results into a trajectory deviation – see Figure 3-24.

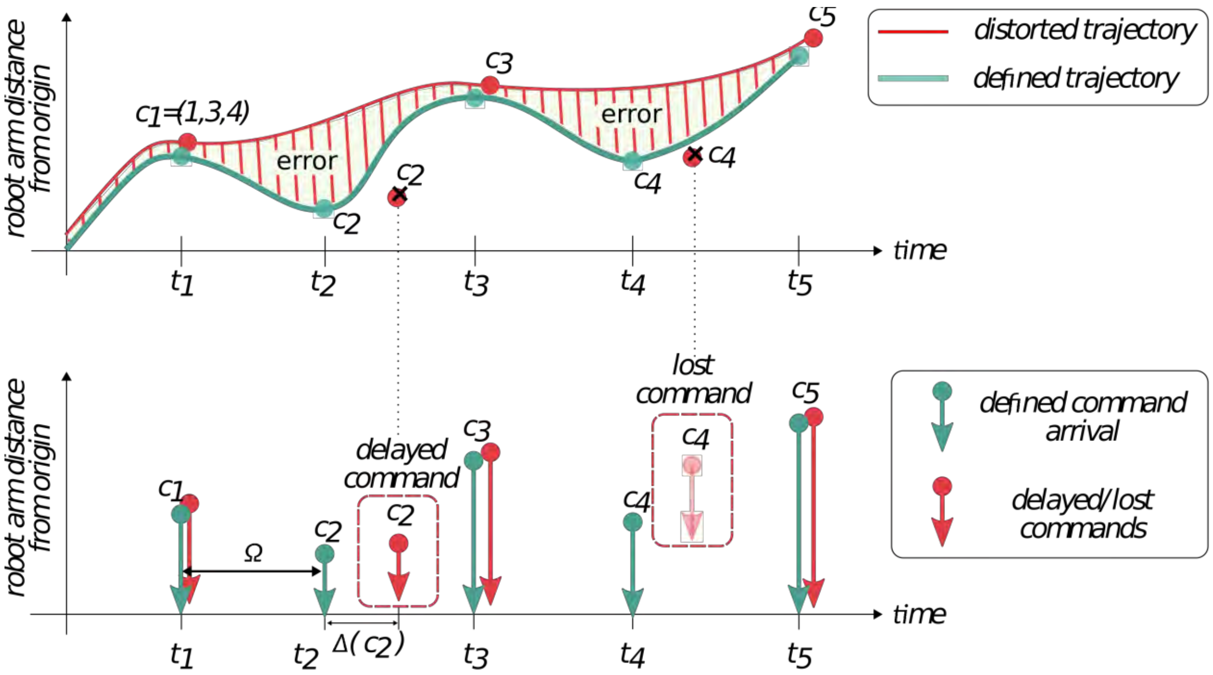


Figure 3-24: Effect of command losses in the robot trajectory.

Note how the trajectory error is impacted by both delayed and lost commands. From Figure 3-24 we draw that it is necessary to mitigate the situation when commands are not timely delivered. To solve such problem, we train an AI model using an AI distributed service – Section 3.2.4, (FoReCo) that solves such problem by forecasting what should be the command that has been lost or delayed in the communication between the Digital Twin and the robotic arm. Specifically, FoReCo infers what would be the (x,y,z) coordinates of such commands and inject them in the drivers of the robotic arm.

FoReCo is trained using the collected network analytics (Section 3.2.3) from multiple factory floors in a FL fashion (Section 3.2.5). To validate FoReCo as an Intelligent Network application for factory floor, we use a Nyrio One robotic arm with its Digital Twin exchanging commands with an Xbox joystick controller. We conduct pick and place tasks performed by experienced and novel remote operators and collect 22893 of their instructions into the Analytics Framework. The dataset collected contains the (x,y,z) coordinates of the robotic arm commands, which are sent each 20ms.

With the help of both datasets, we train different AI/ML algorithms to perform command inference upon losses and delays. Namely, we advocate for Vector AutoRegression (VAR), sequence-to-sequence (seq2seq), and Moving Average solutions (MA). The three algorithms were trained using python3 along scipy, tensorflow, and the numpy libraries, respectively. And we used the 80% of the experienced operator dataset to train all models over 100 epochs in the case of seq2seq, and over the whole 80% of the trace in VAR. Note that MA does not require training, for it only requires to know the moving window over which it performs the average.

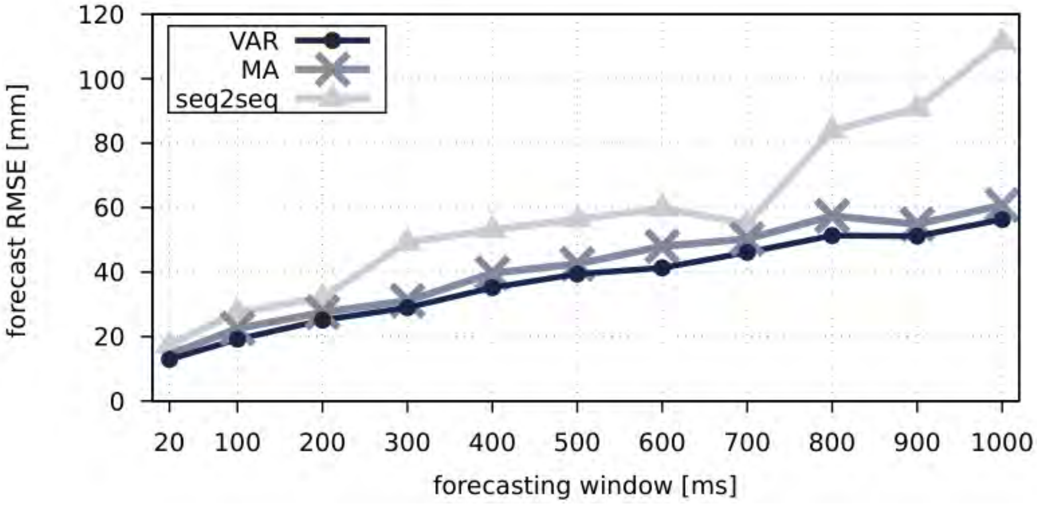


Figure 3-25: Root-Mean-Square Error (RMSE) in function of the forecasting time window.

In our experiments (see Figure 3-25) we increased the forecasting window of FoReCo, i.e., for how long it had to perform forecasts of commands delayed or lost. Results evidence that up to 50 commands (1000ms) we keep a trajectory error around 50mm, which is about a fourth of the >200mm oscillations that happened in the dataset in periods of 1000ms. Experiments showed that FoReCo would achieve its best accuracy using VAR, for it captures the autocorrelated nature of the three-time series considered, i.e., the x, y, and z axis robot coordinates over time.

These initial results set the basis to keep on developing FoReCo as an Intelligent Network application that uses: the analytics framework of AIaaS, DFP, and network programmability. With FoReCo we show that the intelligent network envisioned by Hexa-X recovers accurately lost commands with errors below 20mm upon spurious losses in the communication between the Digital Twin and the robotic arm. Therefore, FoReCo, decreases by a ~95% the 368.74mm trajectory error that the robotic arm would suffer without any recovery mechanism. Hence, we propose an E2E proof of concept for Industry 4.0 in 6G networks.

3.6.2 Inference models considered in FoReCo

As aforementioned, we assessed FoReCo performance using different inference models, namely, VAR, MA, and a seq2seq neural network. In this first stage of FoReCo research we have conducted the validation of these models using the collected dataset.

Now we present in detail how MA, VAR and seq2seq are encoded in FoReCo. In our analysis we have considered the (x, y, z) coordinates followed by the robotic arm during its operation. However, our robotic arm does not move according to three dimensional shifts, but rather with rotation offsets for each of the joints of our robotic arm. We used a Nyrio one robotic arm with six joints $(j_1, j_2, j_3, j_4, j_5, j_6)$ that allow rotation actions along the base, to lean over an object (x2), rotate the manipulator, and grab the object with the gripper.

Rather than running the inference directly over the six axes $j_i, i \leq 6$ we convert the six joints offset to the corresponding (x, y, z) coordinates using the forward kinematics. Specifically, we compute the forward kinematics using the Universal Robotic Description Format (URDF) file of the Nyrio One robot, and the forward kinematics operation provided by the package kinpy. That is, thanks to the six-axis description within the URDF we can obtain the (x, y, z) coordinates knowing the joint offsets, for the forward kinematics is precisely the operation used in robotic mechanics to perform such transformation $\{j_i\}_i^6 \mapsto (x, y, z)$. The figure below (Figure 3-26) illustrates the distance from the origin of the robot as it performed the remote operator performed pick and place actions.

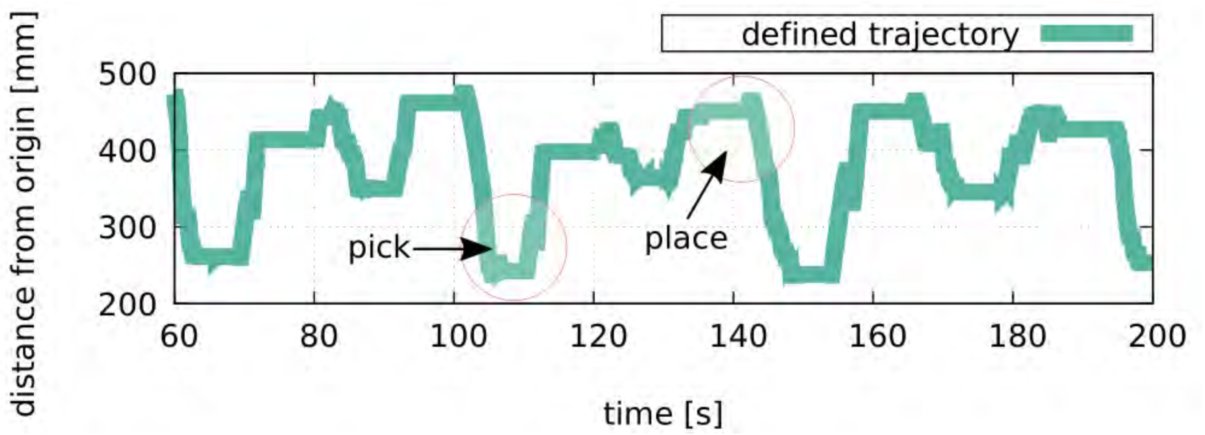


Figure 3-26: Distance from origin when performing pick and place task.

Given the forward kinematics, FoReCo infers the lost command $\hat{c} = (x, y, z)$ using:

- MA: the lost command is obtained by averaging all the prior axis position during a window of length W, e.g., the x-axis of the inferred command is:

$$\hat{x} = \frac{1}{W}\sum_{i=1}^{W} x_{-i}$$

- Vector AutoRegression (VAR): the lost command is obtained using assuming that it is expressed as a linear combination of the prior commands. If we denote $\vec{x} = (x, y, z)$, the VAR inference for the next command is expressed as:

$$\hat{\vec{x}} = \vec{b} + \sum_{i=1}^{W} A_i \vec{x_i}$$

  with A_i being a 1x3 matrix that weights the contribution of the command i in the computation of the inferred command, and $\vec{b}$ the bias.

- Sequence-to-Sequence (seq2seq): the lost command is obtained using a seq2seq neural network that first encodes the available information - the prior commands – and then decodes the encoded sequence as a $\hat{c} = (x, y, z)$ command. We programmed the encoder using an encoder layer of 200 Long Short-Term Memory (LSTM) neurons that repeats the encoded sequence as many times as times ahead in the forecast. That is, if the forecast consists of the next 3 commands, the encoder layer creates three repeated encoded sequences. Such repeated sequences are fed into a densely connected feed forward layer that yields its output to a timely distributed layer that yields $\widehat{c_1}, \widehat{c_2}, \widehat{c_3}$. Thus, the considered seq2seq architecture is what is also known in the literature as a many-to-many LSTM sequence predictor. That is, it receives many prior commands $c_{-1}, c_{-2}, c_{-3}, \ldots$ and yield many predictions of the future $\widehat{c_1}, \widehat{c_2}, \widehat{c_3}$.

It is worth mentioning that the least accurate method turned out to be the seq2seq architecture, for using 200 LSTM neurons resulted in an overwhelming number of parameters to train. To give an idea, predicting up to 50 commands requires training a seq2seq architecture of >16K parameters, thus, the convergence is really challenging. The MA and VAR do not suffer from the curse of dimensionality as we increase the forecasting window, for both methods are applied on top of the prior forecast. That is precisely the autoregressive nature of VAR, for the command $\widehat{c_2}$ is forecasted using prior data and the prior forecast: $\widehat{c_1}, \widehat{c_{-1}}, \widehat{c_{-2}}, \ldots$. On top, VAR was accurate enough given the autocorrelation existing in between the axis, for the (x, y, z) coordinates are interrelated as the robot moves. For example, as the robot rotates along its base, it moves across the $z = 2$ horizontal plane with the (x, y) coordinates satisfying $\sqrt{x^2 + y^2} = r$, i.e., moving along a circle of radius r. Further details and deep analysis of the methods will be provided in the next WP5 deliverable D5.3.

4 Flexible networks

Flexible Networks 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 can be achieved by developing solutions that are capable of managing local ad hoc networks, in coordination with the infrastructure, as well as distributing their functionalities among the infrastructure, the edge and the far-edge devices (e.g., in-X networks such as in-car or in-body) e. In addition, this can be achieved by a network of networks that can both incorporate different (sub)network solutions as well as a network that can easily (flexibly) adapt to new topologies. For example, different network solutions can be a network using sub-terahertz spectrum, different variants of mesh networks, NTNs, High-Altitude Platform Stations (HAPSs) or drones, and local device networks. The network functionality and architecture must then be flexible enough so that it can adapt to these different topologies.

The current vision of 6G has been enhancing the idea of ’ecosystem’ of networks (or network of networks), where each network is a subnetwork (see Appendix A.1 for a complete definition). 6G will finally integrate satellite, aerial, and terrestrial networks in a unique dynamic-adaptive network infrastructure. Another important principle is that new intelligence will be needed to make decisions “on the fly”. More specifically, after the ‘network of networks’ is built, architectural/technology enablers are developed to manage local ad hoc networks in coordination with the (physical) network infrastructure as well as distribute their functionalities between the infrastructure, the edge and the devices.

This chapter addresses both these two aspects, namely the integration of subnetworks in one network infrastructure and then the extension of the infrastructure with local ad hoc networks, as explained below.

First, it comes to address how B5G/6G flexible topologies – local structures (nodes with ad hoc assigned compute-network resources, terminated spanning from the core to the edge nodes and far-edge devices) can be integrated as coordinated extensions of infrastructure even in a temporary manner (Section 4.1). The flexible topologies comprise the discovery and selection of the most appropriate and trustful nodes and far-edge devices to be admitted in the "ad hoc" network formation, as well as the selection of the technologies to use for D2D communications. The D2D/Mesh architecture should be clearly mapped to a M&O architecture (Section 4.1.1) that is able to cope with the complexity of such an architecture and integrate all of its components in a 6G fashion. To be able to scale up, the solution should also include a modular approach to flexible network integration (Section 4.4), which allows the formation of a 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.).

Second, it comprises architectural solutions of sub-networks that support full global coverage through the concept of an NTN architecture capable of efficient inter-satellite-link hops (Section 4.3.2). Several trade-offs do exist, to hop between all orbits (best performance) or hops to closest orbit (relatively good performance with less complexity). In addition, the functional split of the baseband unit for NTN is investigated (Section 4.3.1), with the choice of lower altitude CubeSats, despite the shorter orbit lifetime, to give better results for what concerns the achievable QoS. In addition, a new 6G multi-connectivity proposal is proposed (Section 4.2) that enables efficient spectrum usage. Such a proposal becomes even more important with even more high spectrum bands, now in the sub-THz region (100-300 GHz).

4.1 Device-to-device (D2D) and mesh networks

The Defense Advanced Research Projects Agency (DARPA) Packet Radio NETwork (PRNET) concept was the dawn of ad hoc networks [JT87], also known as D2D networks. Later on, as laptop computers started to become more accessible, the idea of infrastructureless networks started to arise and, thereupon, the Internet Engineering Task Force (IETF) Mobile Ad hoc Networks working group introduced the term of "mobile ad hoc networks". The Institute of Electrical and Electronics Engineers (IEEE) 802.11s task group defined the Wireless Local Area Network (WLAN) Mesh Networking [802.11s]. Since those days, this technology has evolved buoyantly, and several types of Wireless Ad hoc NETworks (WANETs) have been developed. Below, the most relevant types of WANETs, for the scope of this section, are described:

- **Mobile Ad hoc NETworks (MANETs):** Computer network comprised of several mobile nodes, without a fixed topology, which can have the role of both, a router and a host. It does not require an infrastructure to organize and configure the network nodes, they are self-organizing and self-configuring nodes. Most MANETs use the WLAN ad hoc mode and routing on layer 3 with IP addresses. See also [CMC99].
- **Wireless Mesh Networks (WMNs):** Computer network comprised of several mobile nodes, without a fixed topology, which can have the role of both, a forwarding node and an end station. It does not require an infrastructure to organize and configure the network nodes, they are self-organizing and self-configuring nodes. Most wireless mesh networks use the IEEE 802.11 4-address mode and forwarding on layer 2 with Medium Access Control (MAC) addresses.
- **Vehicular Ad hoc NETworks (VANETs):** MANET sub-type where vehicles act as the mobile nodes. Its main focus is to address the issues related to Intelligent Transportation Systems (ITS). VANETs are characterised by the high mobility and velocity of the nodes, real-time data exchange requirements and Vehicle-to-Vehicle (V2V) or Vehicle-to-Infrastructure (V2I) communication channels.
- **Flying Ad hoc NETworks (FANETs):** This kind of MANETs aim at enabling communication between UAVs. Besides, FANETs are able to exploit the communications between ground-based devices an UAVs.

As it can be seen from the above-mentioned MANETs, these technologies are a key component for the development of future 6G architectural components that enable the management of "spontaneous" far-edge local ad hoc Networks. However, it is important to remark that MANETs are not exempt from drawbacks and security concerns such as: *(i) energy efficiency*, mobile nodes are attached to limited batteries in most cases and, therefore, computation operations should be optimised ; *(ii) scalability,* the network should be able to provide an acceptable level of service even in far-edge environments where a huge volume of mobile nodes, with an heterogeneous range of computing resources, may enter the network or to reduce the resources when these are not needed; *(iii) blurry boundaries,* in the absence of fixed infrastructure, MANETs are characterized by their highly dynamic nature, since mobile nodes of unknown so far identity can leave and enter the network at any time, leaving space to eavesdropping, impersonation and DoS attacks; *(iv) trustworthiness,* in MANETs, mobile node categorization is complex as there is an inherent lack of protection given their highly dynamic nature due to the fact that these nodes can be of unknown so far identity.

D2D communication is defined as direct communication between two mobile UEs without traversing the Base Station (BS) or CN, and it can occur on the cellular frequencies or unlicensed spectrum. A mesh network is a local area network topology in which the mesh nodes connect directly, dynamically, and non-hierarchically to other nodes and cooperate with one another to efficiently route data to and from clients. In this respect, mesh networking can be considered to support multi-hop D2D communications in 5G/B5G/6G networks.

This chapter addresses the following research questions and problems:

1. How “trusted” does a device have to be in order to be part of the D2D/mesh network?
2. Unified modelling of far-edge nodes and devices, in terms of network and computational resource characteristics, capabilities and constraints.
3. Definition of interfaces to control and interact with far-edge devices for resource advertisements, synchronization, reachability verification, etc.
4. Design algorithms for selecting the best possible nodes and far-edge devices depending on specific parameters (e.g., position, signal quality, battery level, availability, reachability, available computational resources etc.).
5. Integration with network and service orchestration for seamless management, control, and enforcement of D2D/mesh network communications to satisfy end-users application constraints and requirements. It includes proper abstraction of D2D mesh network topologies towards orchestration layers.
6. Methods and procedures for discovery of nodes and far-edge devices (including synchronization aspects for capabilities advertisement).
7. The best possible routing for multi-hop D2D communications (creation of routing tables), minimizing latency or increasing resilience and cost efficiency.
8. Select technology/technologies to use for D2D communications.

In order to investigate the above research questions and problems, the following architecture and workflow is defined.

The nodes (including the far-edge devices) are publishing information, in which they define their expected behaviour by providing information about themselves (e.g., verifiable credentials), their capabilities, the resources they offer (e.g., service endpoints) and usage policies (access and exploitation). We consider as nodes any physical or logical entity, including end user devices, belonging to any type of network (access, transport, core etc.) and technology (4G (E-UTRA), 5G NR, WiFi, etc.) and having network/compute capabilities.

Specifically, each node and far-edge device should follow a common information model to advertise its characteristics and capabilities, providing at least:

- Unique identifier (e.g., temporary identifier for a certain area)
- Type of node (e.g., mobile or static node)
- Verifiable credentials
- Network interfaces available, and for each
    - type of interface (e.g., air interface vs. wired interface)
    - protocols supported and associated security concerns
    - max UL/DL capabilities
- Compute resources available, including
    - Amount of CPU, storage, memory
    - Supported virtualization technology
- List of interfaces/APIs to be used (e.g., by the ad hoc NW control component) to control and manage the use of resources within the node (both network and compute) trying to optimize several objectives (e.g., power consumption, coverage, capacity, etc.).

This information will be later used by Node Discovery for discovery and to evaluate interactions afterwards in the Trust Manager, serving as an extra basis for trust management. That is, adherence to declared behaviours will help build nodes' trust.

The Trust Manager performs a continuous monitoring and evaluation of Node participants, initially not trusted or with an initial trust computation based on capabilities and identity information. IoT/Edge/Cloud nodes will be able to join, leave, or move between different domains. The Trust Manager performs dynamic trust evaluations and sharing of auditable public data (and trust values) throughout the IoT-edge-cloud continuum.

The Node Discovery comprises methods and procedures for discovery of nodes and far-edge devices (including synchronization aspects for capabilities advertisement) based on their self-descriptions and the trust values. The discovery process can be based on a publish/subscribe mechanism, similar to the SBA approach, where each involved entity can send or receive data asynchronously (e.g., Node Discovery as information consumer, and nodes and far-edge devices as information producer). This approach allows the decoupling of the node discovery functionality from the dynamicity of the nodes and far-edge devices. Indeed, specific synchronization and availability mechanisms have to be considered, especially when we consider nodes and far-edge devices which are not static and follow some mobility patterns. In practice, periodic availability and reachability messages (e.g., in the form of keepalive) should be published by each node and far-edge device to confirm their status and availability. Therefore, with the aim of keeping an up-to-date view of available nodes (i.e., reachable) the Node Discovery should embed a repository function to be exploited by other components (e.g., the Adhoc NW control, the Trust Manager)

The Adhoc NW Control 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.

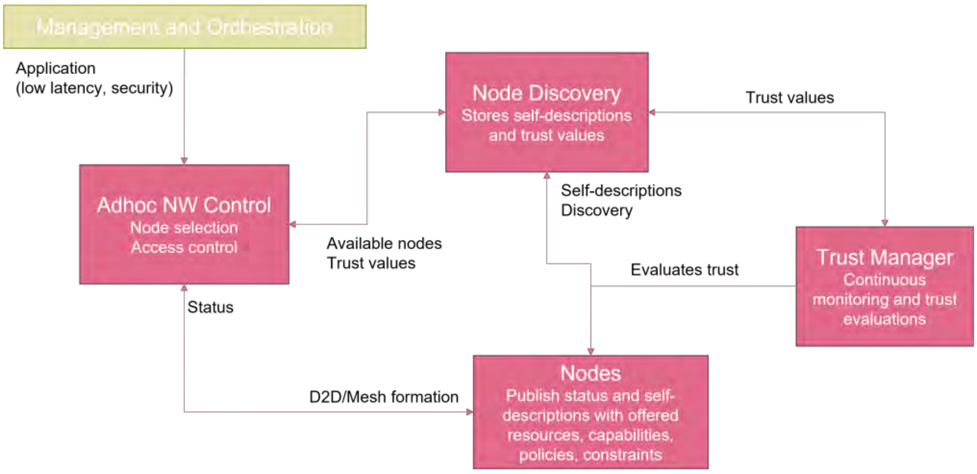


Figure 4-1: High level architecture for D2D and Mesh networks

The following picture highlights which research questions are answered by which components.

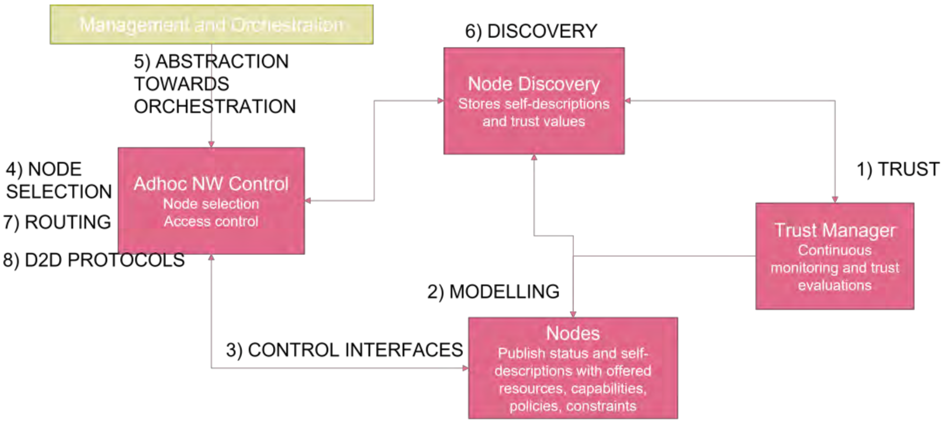

Management and Orchestration
5) ABSTRACTION TOWARDS ORCHESTRATION
6) DISCOVERY
Node Discovery
Stores self-descriptions and trust values
1) TRUST
4) NODE SELECTION
7) ROUTING
8) D2D PROTOCOLS
Adhoc NW Control
Node selection
Access control
Trust Manager
Continuous monitoring and trust evaluations
2) MODELLING
3) CONTROL INTERFACES
Nodes
Publish status and self-descriptions with offered resources, capabilities, policies, constraints

Figure 4-2: Mapping of research questions and problems with the architectural components

4.1.1 Local ad hoc network management

To solve the issues presented in Section 4.1, novel technological enhancements are being implemented, jointly with “legacy” MANETs, such as the inclusion of AI/ML learning algorithms to optimize energy efficiency [HCJ15], scalability and route selection [KY20][KDN+21]; using blockchain [LDY+22] or fuzzy-based trust computing patterns [JTS17] to increase the overall network security and even creating SDN-based architectures for infrastructureless efficient MANETs [DS21].

Nonetheless, future 6G management of local ad hoc networks (i.e., WANETs or MANETs of any kind) should be able to address the majority of the afore-mentioned drawbacks at once. In order to fulfil these requirements, the D2D/Mesh architecture proposed in Section 4.1, which aims at solving most of these issues, should be clearly mapped to a M&O architecture that is able to cope with the D2D/Mesh architecture complexity and integrate its components as per the expected M&O requirements for future 6G networks. Hexa-X WP6 D6.2 has developed such a M&O architecture [HEX-D62]. This sub-section aims at answering Section 4.1 research question #5: “*Abstraction towards orchestration”,* by creating a clear mapping between Section 4.1 D2D/Mesh proposed architecture components and Hexa-X WP6 D6.2 M&O Network Layer elements. This mapping is depicted in Figure 4-3.

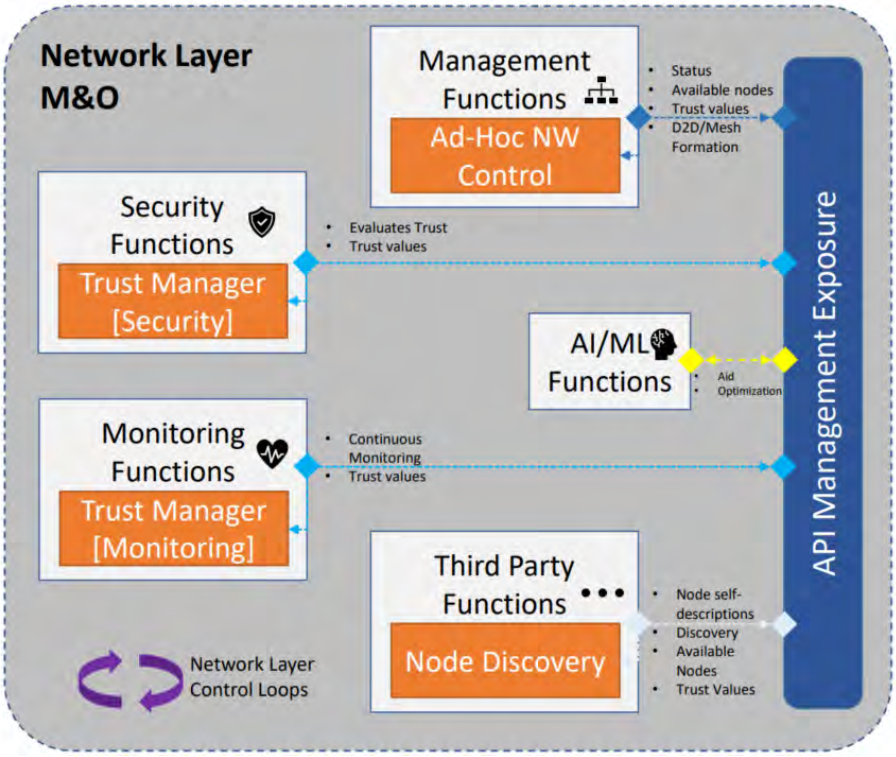


Figure 4-3: D2D/Mesh network architecture Components mapping to M&O Components [HEX-D62].

This innovative M&O Architecture sets four main layers*: (i) Service Layer,* oriented for verticals and in charge of service creation, operation, data aggregation, intent-based service management and service quality management; *(ii) Network Layer,* allocates NFs (mainly as CNFs but also as VNFs, Physical Network Functions (PNFs) or Hybrid Network Functions (HNFs)), manages the NFs LCM and is in charge of monitoring and AI/ML orchestration actions; *(iii) Infrastructure Layer,* includes extreme-edge resources, edge/central-cloud, hyperscalers, public/private networks, transport network, etc.; *(iv) Design Layer,* in charge of M&O-related operations from 3rd party Software providers (i.e., Intent-based service definitions, DevOps/AIOps Framework, SW& Descriptors Design). The communication between these layers and the network elements that comprise each layer is carried out through a new cross-layer known as the "*Application Programming Interface Management Exposure*" which enables the adoption of the Service Based Management Architecture (SBMA) model. Finally, this M&O brings a clear separation between MOs (instances of managed resources) and Managing Objects (M&O resources that offer management capabilities to act upon MOs).

Considering the capabilities of each of the architectural elements of the D2D/Mesh architecture proposed in Section 4.1, the following mapping is proposed:

- **Ad hoc NW control**: Can be mapped directly into the Primary M&O Functions, also referred as "*Management Functions*" as they offer fulfilment, assurance and artifact management capabilities (i.e., ad hoc NW control performs node selection and considers data input from other M&O to take decisions). Acts as a Managing Object.
- **Trust Manager:** The functionality of this module may be divided as follows. Monitoring and evaluation of participant nodes' operations can be mapped within the "*Monitoring Functions*" scope while the trust evaluations and sharing of auditable data may be mapped into the "*Security Functions*" as they are in charge of providing information regarding the operational processes

and protecting the confidentiality and integrity of operations, and data, as well as ensuring the continuity of the provided services, respectively. Both parts act as Managing and Managed Objects.

- **Node Discovery:** This module has a specific functionality for the D2D proposed architecture and, therefore, falls into the M&O "*Third Party Functions*" module. Acts as a Managed Object.
- **Nodes:** These elements have not been depictured in Figure 4-3 because they are part of the *"Infrastructure Layer"*. Node status and self-descriptions will be published through the API Management Exposure cross-layer to the corresponding D2D architecture modules.

Furthermore, the *"AI/ML Functions"* M&O module has been added to Figure 4-3 as a proposal to enhance the overall D2D architecture behaviour. This module may aid the other elements to optimize selected processes or operations using the data given by the Trust Manager, ad hoc NW Control and/or node discovery modules. All the communication links between these modules, within the Network Layer M&O scope, will be performed through the API Management Exposure cross-layer module.

4.2 6G Multi-connectivity (MC)

MC occurs when a device has multiple radio connections to the network. The multiple connections to the UE involve one or more base stations, which may or may not be physically in different locations. In 3G, the soft HO is a case of MC, and in Long Term Evolution (LTE), DC and Carrier Aggregation (CA) are examples of MC. Soft HO transmits (and receives) the same signal (same data) via different base stations to and from the UE in a synchronous manner. Thus, the main purpose of soft HO is reliability for intra frequency HO. However, the MC solutions in 4G (DC and CA) transmits different data flows over the connections. Other types of MC solutions are multiple Transmission and Reception Point (multi-TRP) and Distributed Multiple Input Multiple Output (D-MIMO) solutions [HEXA-D51], but here we will focus on DC and CA type of MC.

In DC and CA, the data radio bearer is terminated in either the master or the secondary base station. The Split Data Radio Bearer (DRB) splits the data and distributes (or receives) data to/from the Master Cell Group (MCG) and the Secondary Cell Group (SCG), see Figure 4-4. DC splits the data between the cell groups at the Packet Data Convergence Protocol (PDCP) layer while CA splits the data at the MAC layer, see Figure 4-4.

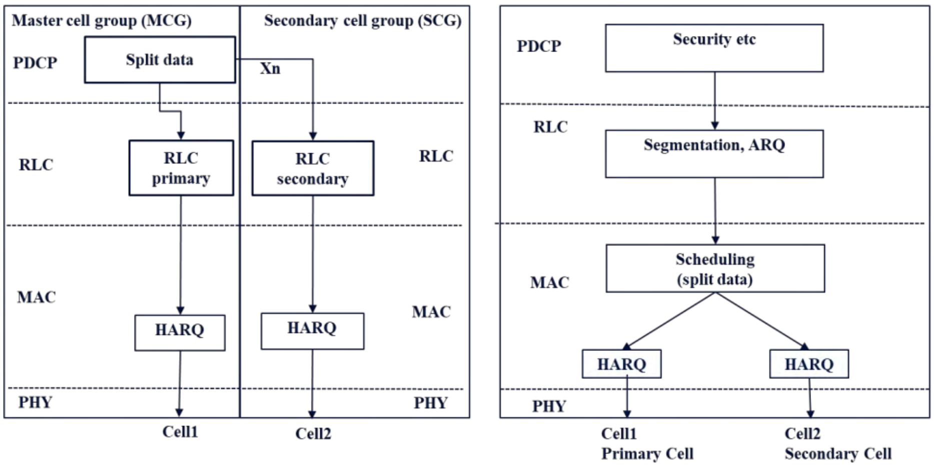


Figure 4-4: Protocol stack view of the DC (left) and CA (right).

The main purpose of CA and DC is to improve the throughput and improve the utilization of different frequency bands. The utilization can be improved since the user can be connected to one low frequency cell with good coverage and decent throughput and at the same time connect to a high frequency cell with sporadic coverage but with very high throughput. CA requires that the involved base stations are synchronized, and that the connection (backhaul) is very fast and has a very low delay (less than 1 ms). DC is designed for a relaxed backhaul (Xn) connection and allows delays in the order of several ms. DC and CA are especially beneficial for networks with varying frequency bands with both Frequency Range 1 (FR1) and Frequency Range 2 (FR2) frequencies, where the low frequency node (FR1) can give reliability while the high frequency (FR2) node can boost the user throughput when it is available. However, DC has some drawbacks. One of the main drawbacks is the connection between the master and the secondary node. Since the connection between the nodes may be rather slow, the master will not have the most recent information about the secondary node performance. To handle this, a flow control protocol can be implemented between the master and the secondary node. The flow control works in a similar way as TCP/IP, i.e., the master can estimate the throughput based on the acknowledgements it receives from the secondary node. In case there is a rather high latency of the Xn interface, and that the secondary node throughput varies, data can easily be stalled in the secondary node. Another feature of DC is that DL and UL are always coupled. This means that all connections in the UL must be able to send acknowledgements of the Hybrid Automatic Repeat reQuest (HARQ) or Radio Link Control (RLC) packets. This can in some cases be beneficial, for example, if one of the connections fails, the remaining connection can keep the user from entering RLF. However, since the secondary connection may have worse UL coverage than the master (the difference may very well be of several dBs, depending on the frequency range), the secondary node feedback may become so bad that this may cause a sharp increase in the round-trip times (or even a timeout), and this will cause a decrease the TCP/IP connection throughput. In CA, there is no need for a flow control since the backhaul is assumed to be very fast and low latency; instead, there is a centralized scheduler in the Primary Cell (PCell). The DL and UL connections are not coupled in CA, the best UL (i.e., the PCell) can be used for UL response, which means that the UL coverage is often better for CA compared to DC, as the UE does not have to split its limited uplink transmit power between two concurrent uplinks. Yet another disadvantage of DC in 5G is the many architecture options due to the multi- Radio Access Technology (RAT) aggregation solution between the Evolved Universal Terrestrial Radio Access (E-UTRA, also known as LTE) and NR – E-UTRA-NR Dual Connectivity (EN-DC), Next Generation RAN (NG-RAN) E-UTRA-NR Dual Connectivity (NGEN-DC), NR-E-UTRA Dual Connectivity (NE-DC) – some of which were never implemented. These options increased the complexity of the specifications and increased the number of test cases to avoid multi-vendor interoperability issues to the expense of supporting fewer options.

With 6G, it becomes even more important with a MC solution with the ability to have efficient spectrum usage and be able to aggregate resources between the current frequency bands and the new sub-THz spectrum bands. This calls for a new improved MC solution. Based on the above descriptions of DC and CA, both CA and DC have its pros and cons. Therefore, a 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 MC solution should support decoupled DL and UL as well as decoupled CP and UP. To simplify the solutions, the number of architecture options should be limited, e.g., by only allowing MC between 6G enabled base stations. Figure 4-5 shows a possible 6G MC concept. In this example, the user has two DL connections and one UL connection, as well as two inactive connections. 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 example in Figure 4-5 shows one UL connection and two active DL connections connected to a cloud RAN/CN. The new concept should also support "in-active" connections.

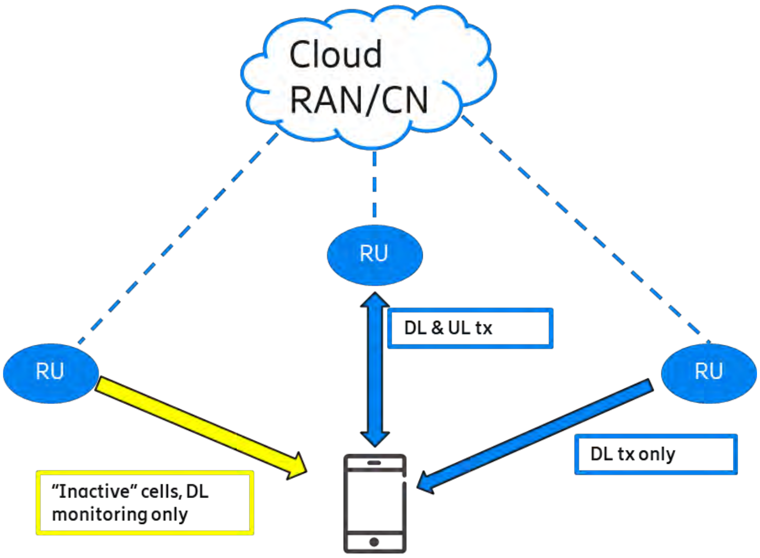


Figure 4-5: Possible 6G MC concept.

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. Also note that the more details will be given in next deliverable D5.3. The implication for the 6G architecture is that this will reduce the complexity and that this also fits well with the cloudification of the RAN functions as shown in Figure 4-5.

4.3 NTN architecture

One important subnetwork of the network of networks is the NTN (Satellite) subnetwork which can complement the terrestrial subnetworks to provide coverage to exceptionally large and isolated rural areas at a relatively low cost. For urban areas (i.e., areas with high population density), the benefits with a satellite system are lower since it can be difficult to efficiently support the needed high capacity per area. All non-terrestrial systems need to have some sort of connection to the terrestrial network, typically via a so-called ground station, see Figure 4-6. UEs connect to the satellites via the service link, and the satellite in turn connects with the ground station, the feeder link. The ground station is in turn connected to the CN for further routing of the session call. The angle of the feeder and service links should not be too small for a reliable connection [HH19]. MEA_F is the feeder link's Minimum Elevation Angle (MEA), and MEA_S is the service link's MEA.

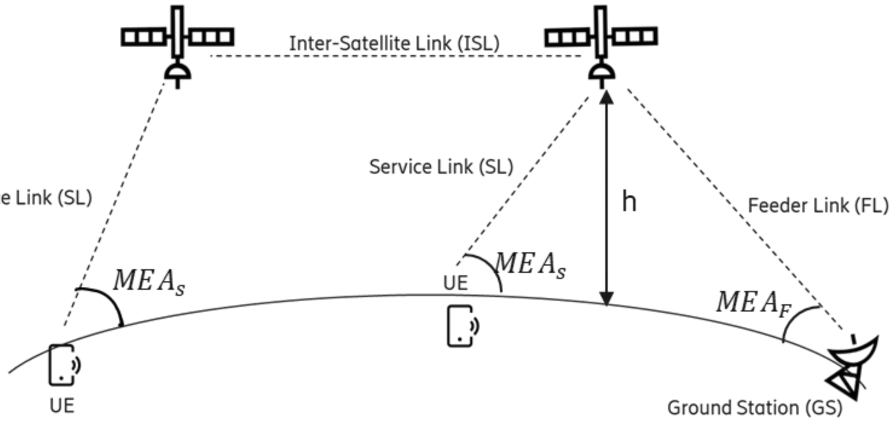


Figure 4-6: Nodes and link involved in a 6G satellite system.

There are basically two types of architecture options for NTN: Transparent and Regenerative payload. For the transparent payload the NTN node serves as a relay of the signal between the UE and the gNB on ground. For the regenerative payload the gNB is located onboard the satellite whereas for the transparent payload the gNB is located in the ground station. The main advantage with the regenerative architecture is that it is more capable to perform, e.g., beamforming and (multi) hops over the inter-satellite links. The main advantage with the transparent architecture is that there is no need for hardware that supports a full gNB, which means that the weight and energy consumption of the satellite can be lower compared to the case when the full gNB functionality is onboard.

4.3.1 3D architecture

From the first generation of wireless cellular networks, the main focus has been the service provisioning to terrestrial users. Even if some interaction and coworking with aerial platforms and satellites have been investigated, no seamless integration has been implemented and standardized yet. Further, the requirement of constant network availability envisioned in 6G (also in remote and rural areas) has brought to the concept of three-dimensional (3D) networking. This means that satellites, HAPS, and aerial platforms are seamlessly part of networking and routing as well as managed by in-network intelligence and part of the softwarized network continuum. No distinction will be made from network orchestration between terrestrial networks, and aerial and satellite ones. By providing network coverage and resources to remote and complex areas with limited or no network infrastructure, base stations are normally deployed via UAVs (the so-called mobile base stations). In this way, the edge of the network and the respective computing is hosted in HAPS or satellites. However, this great potential obtained creates a trade off with the limited resources available at network nodes. Because of that, especially considering the base stations on UAVs, the functional split of virtual baseband units can help in saving computing resources at the mobile base stations, offloading the tasks partially to the HAPS and satellites. In the first feasibility studies, we considered the scenario, in which UAVs are DUs and the CU is on a Low Earth Orbit (LEO) nanosatellite, see Figure 4-7. The LEO satellites are typically below 2000 km in altitude while Geostationary Equatorial Orbit (GEO) satellites orbits have an altitude of around 35000 km. Particularly, nanosatellites in LEO orbits are used because of the reduction of propagation delay compared to e.g., GEO satellites. The link capacity should be greater than 180 Mb/s (achieved with a Bit Error Rate (BER) lower than 10^{-12} [CPRI15]). Next, the link delay should be less than 150 μs, but with the possibility of increasing up to 4 ms, accepting higher error probability [JIT+16]. According to the evaluations provided in [BBG+20] [BGS+20], an upper bound of 2.1 ms cannot be exceeded. As far as channel coding and waveform design are concerned, the fronthaul standards, basically relying on fibre connections, do not provide specific insights for the design of satellite radio connection links. Anyway, the fronthaul employs Reed-Solomon (RS) codes as a tool for Forward Error Correction (FEC). Such a solution is characterized by a satisfactory trade-off between high error correction capacity and low redundancy, obtained with a tolerable decoding complexity

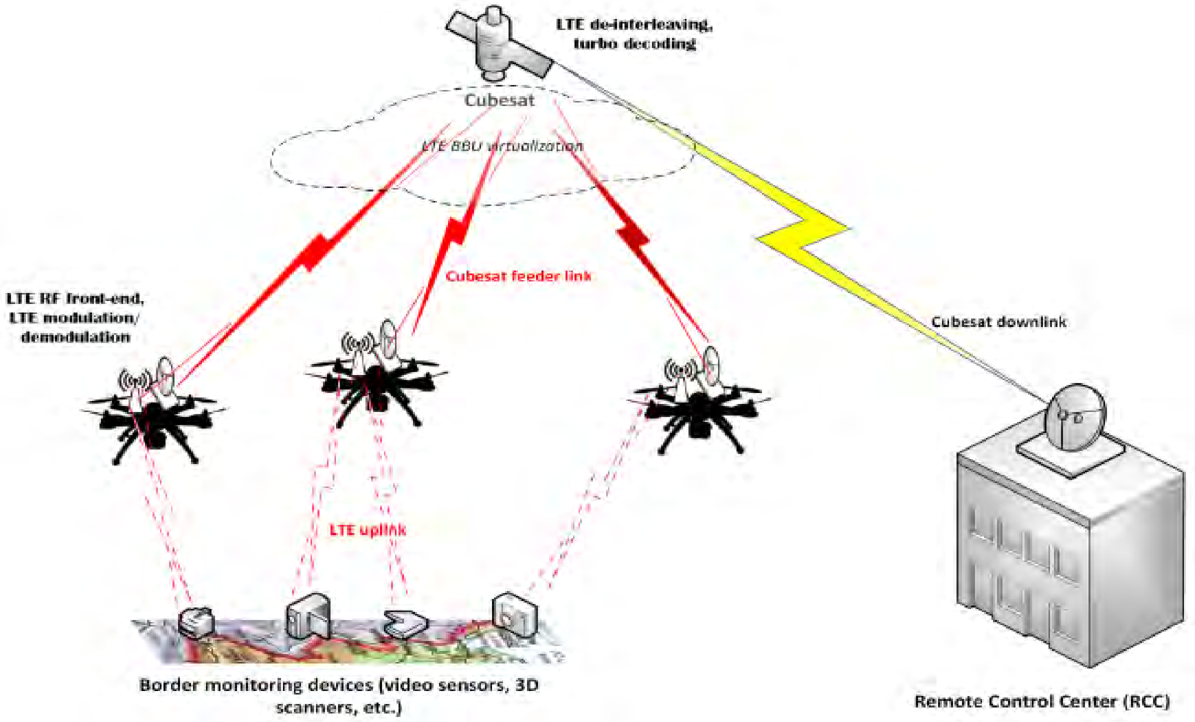

LTE de-interleaving, turbo decoding
Cubesat
LTE BBU virtualization
Cubesat feeder link
LTE RF front-end, LTE modulation/ demodulation
Cubesat downlink
LTE uplink
Border monitoring devices (video sensors, 3D scanners, etc.)
Remote Control Center (RCC)

Figure 4-7: Conceptual representation of the 3D scenario in which virtualised BBU is partially hosted by the drone and partially by the nanosatellites (CU/DU split). -

This 6G 3D scenario has been simulated in MATLAB and SIMULINK, considering the characteristics and the modelling in [BBG+20] [BGS+20], called CubeSat simulator. The specific use case is the virtualisation of the BaseBand Unit (BBU) and its functional split. The BBU subfunctions are then partially offloaded to the nanosatellite to save computing resources at the UAVs, which have more strict battery constraints. The CubeSat orbit simulator, which receives as input the satellite altitude and the minimum number of turbo decoding iterations to be performed by the remote BBU at the nanosatellite (such a value is directly linked to the CubeSat link delay as shown in [BBG+20]). The orbital simulator computes the overall orbital parameters, providing as output a plot of the satellite speed vs. the time of visibility (related to the amplitude of the angle of transmission), hereinafter denoted by CubeSat flight time and the distance between the satellite and the Earth position of the drone. These parameters allow derivation of the time varying curves of Doppler shift and link pathloss that are sent as input to the CubeSat Radio Frequency (RF) link simulator, parameterized by link budget (see Table 2 in [BBG+20] for detailed explanation). The output of this simulator is substantially the uncoded link BER mapped vs. the CubeSat flight time that is afterwards converted to the BER after RS decoding. Then, the compliance of the simulated BER with Common Public Radio Interface (CPRI) requirements is checked. If the check is positive, the last simulation step related to the BBU iterative turbo decoding starts on MATLAB. Otherwise, a revision of the CubeSat feeder link budget is needed, and the related simulations should restart. The iterative turbo decoder receives as input the link budget related to the connection between the ground sensor and the drone. The last step of our simulation process, related to the LTE turbo decoding results, are performed by the virtualized BBU, installed on the CubeSat. The BBU will process the turbo-encoded bits with errors that are produced by the propagation impairments of the ground-to-drone link.

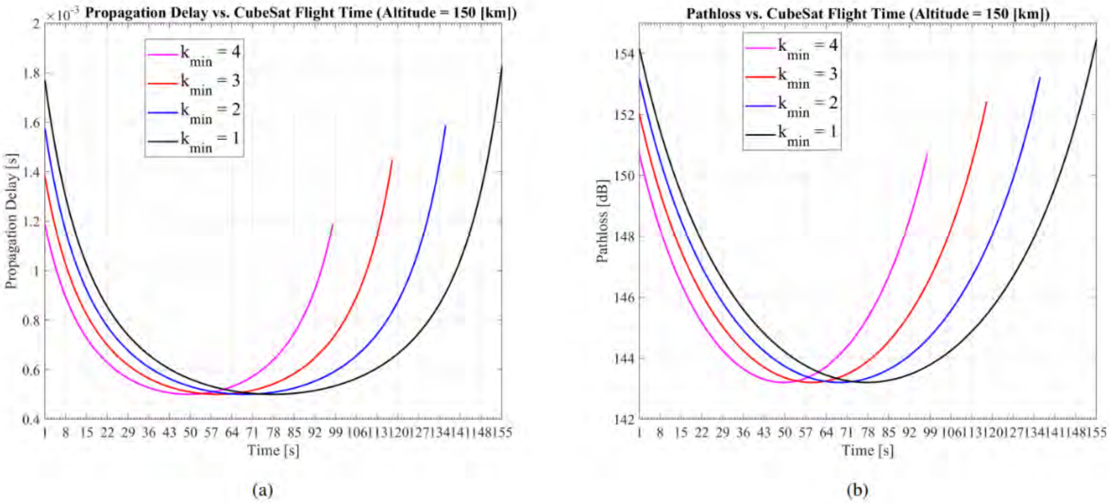


Figure 4-8: Orbital simulation results for CubeSat altitude of 150 km: (a) propagation delay vs. time (b) pathloss vs. time.

Figure 4-8 shows the propagation delay versus CubeSat simulated flight time for the satellite altitude of 150 km. The curves have been plotted vs. the CubeSat flight time for different values of the minimum number of turbo decoding iterations allowed in case of Split D (outsource of the HARQ and FEC operations of the BBU) of virtual BBU (k_{min}). There is an indirect correlation between pathloss, flight time and the number of decoding iterations at DU side. This is since all the baseband processing should be performed within a certain delay budget, which is mainly the sum of transmission delay and the time for the baseband processing. Thus, increasing the number of decoding iterations k means a higher baseband processing time at the price of an overall lower transmission delay. Vice versa, lowering k leads to a lower baseband processing time and, on the contrary, a greater transmission time, saved for physically transmitting the signal. Consequently, the distance between UAV and CubeSat can be raised, as well as the minimum angle (between satellite and CubeSat) can be reduced, by paying the price in terms of QoS, which translates into a higher pathloss for the longer path to be travelled by the signal and an augmented flight time for a fixed altitude. The choice of lower altitude CubeSats (i.e., a LEO satellite with the shape of a cube of size 10 cm or with the shape of a Parallelepiped consisting of multiple cubes of 10 cm), despite the shorter orbit lifetime, looks better for what concerns the achievable latency. Indeed, the pathloss is overall reduced and it is possible for the BBU to execute a higher number of turbo decoding iterations for higher percentages of the flight time. Also, the flight time overall increases and the higher orbital speed of lower altitude CubeSats involves a Doppler increase.

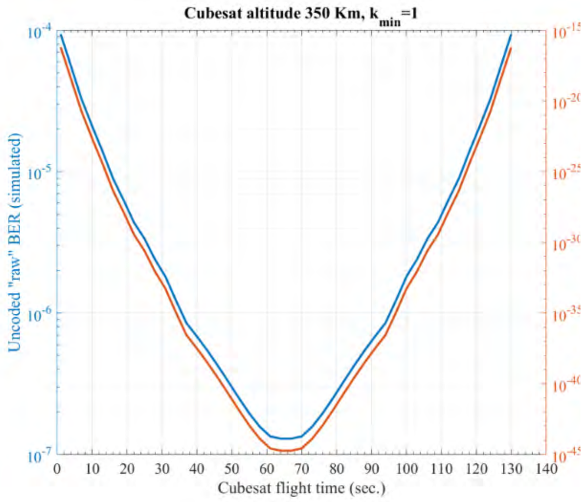


Figure 4-9: Remote Radio Head (RRH)-BBU CubeSat link: BER performance obtained by simulations (left-hand y-axis) and BER after on-board RS decoding (right-hand y-axis).

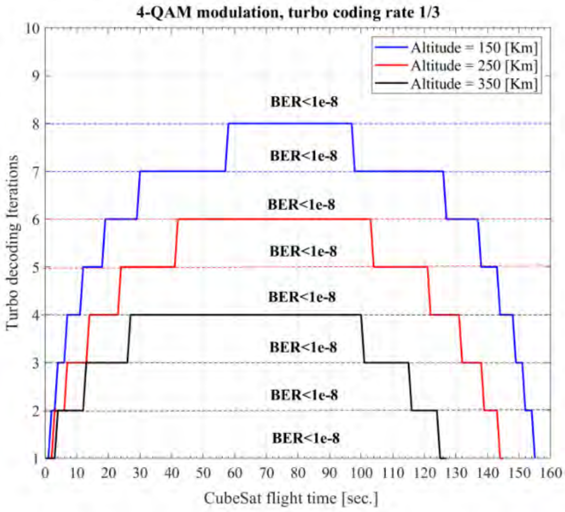


Figure 4-10: Performance of virtualized iterative turbo decoding vs. CubeSat flight time: 4-QAM modulation.

Figure 4-9 shows the uncoded BER results vs. CubeSat flight time; the blue (left-hand y-axis) line is simulated while the light-brown curve (the right-hand y-axis) is the analytically derived BER results after on-board RS decoding. Depending on the CubeSat position during its flight, the uncoded BER ranges from 9×10^{-5} and 1.5×10^{-7} while the BER after RS decoding ranges from 10^{-16} and 10^{-45}. These last values widely satisfy the quasi-error-free CPRI link requirements and confirm that the link budget is suitably designed. Finally, Figure 4-10 displays the performance of the virtualized iterative turbo decoding vs. CubeSat flight time for $k_{\text{min}} = 1$ (the minimum number of iterations of the turbo decoder) and five core processors installed inside the remote BBU, with 4 Quadrature Amplitude Modulation (4-QAM). It is possible to see that if the BBU can perform a single decoding iteration, all channel errors coming from the ground link are corrected, with a measured BER lower than 10^{-8} during the entire period of visibility of the CubeSat. Increasing the spectral efficiency with a factor 2, in case of applying a 16-QAM modulation, implies a slight degradation of BER performance at the output of the virtualized turbo decoder. Indeed, a single decoding iteration does not yield error-free transmission, but some channel errors remain uncorrected. Error-free is reached for decoding iterations greater than 2 and, therefore for a reduced percentage of the CubeSat flight time, as compared to the 4-QAM case. A further increase of spectral efficiency, obtained by using a 64-QAM modulation, is paid with higher error-rates at the output of the virtualized turbo decoder. Indeed, after four decoding iterations, the turbo decoder converges to a BER equal to 2×10^{-6} and does no longer improve for higher iteration numbers. The evidence is better performance yielded by lower-orbit CubeSats (altitudes 150 km and 250 km) in terms of superior QoS for a higher percentage of the CubeSat flight time. However, the use of a CubeSat flying at a for example 350 km altitude can be regarded as a good compromise between performance, platform stability and orbital lifetime. This initial feasibility study shows the complexity and the careful choice of parameters that the design of 6G 3D networks will need. Several trade-offs can arise, especially when microservices' and agents' placement are also involved in the scenario.

4.3.2 NTN global coverage and inter-satellite link schemes

One important aspect of the satellite coverage is how important it is to relay the connection between satellites to improve the coverage, i.e., Inter-Satellite Link (ISL). If the inter-satellite links are important for the coverage, we need an architecture that supports an efficient ISL. This most likely means we should be able to decode and encode the signal from another satellite (and not use amplify and forward), i.e., probably a regenerative or a hybrid architecture (see [HEX-D51]). To decide this, we performed a calculation of the potential coverage with and without inter-satellite links hops. The satellites are placed in a bin (one per degree in latitude and altitude) pattern at LEO at 600 km over earth surface. Also, we place a ground station in each bin on earth in the same manner. The feeder link has a minimum elevation angle (MEA_f, see Figure 4-6) of 20 degrees, to limit the distance. If hops are allowed, the ISL links are also limited, the angle must be at least 30 degrees (MEA_S) and the minimum altitude is 400 km (to avoid too bad connections). The results without and with ISL hops are shown in Figure 4-11.

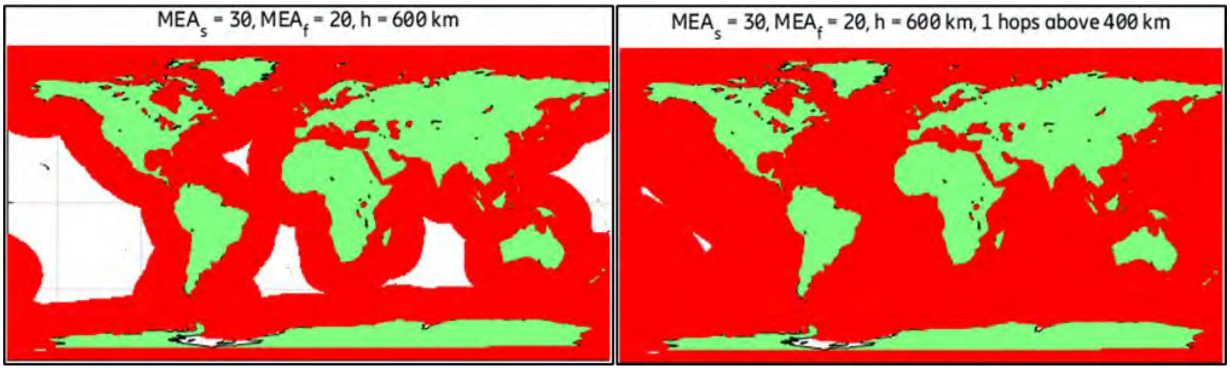

$\mathrm{MEA}_s = 30, \mathrm{MEA}_f = 20, h = 600$ km
$\mathrm{MEA}_s = 30, \mathrm{MEA}_f = 20, h = 600$ km, 1 hops above 400 km

Figure 4-11: Red dots indicate ocean coverage: without any ISL hops (left) vs with ISL hops (right)

Thus, for a full global coverage, including the ocean areas, we need a solution that permits efficient ISL hops between satellites to reach a ground station. However, the previous evaluation was an ideal case, with satellite and ground stations unrealistically densely placed uniformly along the orbit. Also, it may be difficult to perform ISL from one satellite to a satellite in any other orbit. The reason is that due to satellite movements ISLs need to be set-up and torn down continuously on a very fast basis. Allowing any type of ISL scheme between the satellites may be difficult and expensive. One solution to this is to reduce the number of maintained ISLs or use ones that are less mobile. The question we want to answer here is what the impact on coverage and performance is if we use a less advanced ISL scheme (with fewer setup and teardown connections). Here we will investigate some ISL schemes how to hop between the orbits compared to a more ideal situation, using a more realistic number of satellites. There are two main options for the ISL hops: same-orbit ISL and inter-orbit ISL, see Figure 4-12.

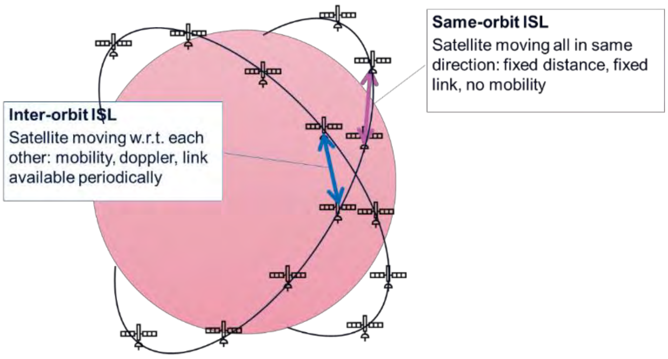

Same-orbit ISL
Satellite moving all in same direction: fixed distance, fixed link, no mobility
Inter-orbit ISL
Satellite moving w.r.t. each other: mobility, doppler, link available periodically

Figure 4-12: Topologies and satellite orbits.

The same-orbit ISL hop is, from the satellite's perspective, a fixed connection. With the inter-orbit ISL, the satellites are moving, and the connection will experience different propagation and other impairments such as Doppler losses. This means that these latter links must be monitored by the involved satellites and in the case the connection becomes too bad the ISL needs to be torn down and a new ISL needs to be found. In this study we investigate four different schemes, see Table 4-1.

Table 4-1 Inter-Satellite Link schemes

Scheme	Criteria	Connection	Altitude	#Satellites
All@1200	ISL has a minimum altitude of 350 km	ISLs are time-varying and created and destroyed continuously	1200 km	720
All@600	ISL has a minimum altitude of 350 km	ISLs are time-varying and created and destroyed continuously	600 km	720
PrevNextSides	Connect closest in same orbit and closest in neighbouring orbits	ISLs are created once but are time-varying	600 km	720
PrevNext	Connect to closest in same orbit	ISLs are created once and are fixed	600 km	720

The number of ground stations are 200 and they are randomly placed on the coastline. Figure 4-13 shows the one-way delay for different ISL hop schemes.

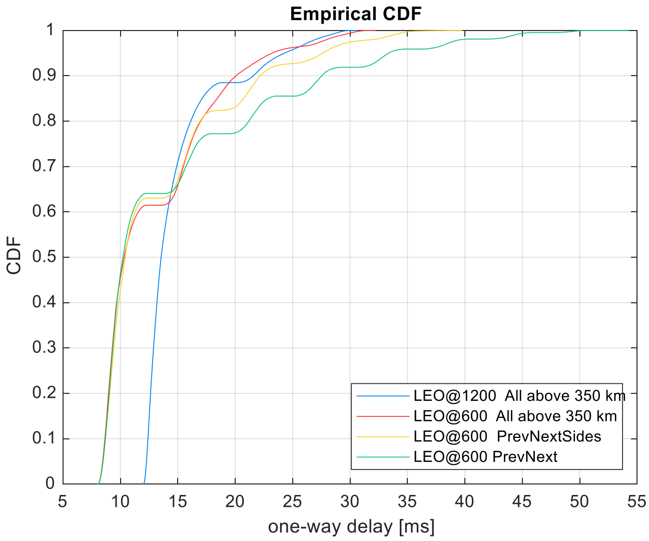


Figure 4-13: One-way delay for the different NTN inter-satellite-link hop schemes.

Increasing the altitude slightly (LEO at 1200km height) improves coverage but also increases the minimum one-way delay. Assuming a maximum delay of 30 ms, the PrevNext scheme has a delay above that for 8% of the connections. The other three schemes show rather similar one-wat delay, around 0-3% of the connections above 30 ms. This means that it is probably possible to use a simplified scheme such as PrevNextSides but not the simplest scheme (such as PrevNext).

4.4 Modular approach to flexible network integration

The 5G network complexity compared to the previous generation is much higher that is justified by new functionalities. There are however some concerns related to scalability, limited programmability

[IKC20], interaction complexity of the management plane, strong dependencies of the CP protocols and limitations of UP which is not able to accommodate different transport solutions like Disruption Tolerant Networking (DTN), Time Sensitive Networking (TSN) or segment routing). Despite using virtualisation technologies, the solution still lacks openness, making it hard to use in a heterogeneous environment, for example it is impossible to create E2E network slices with segments different than NR or 5G Core network (5GC) [MBC22]. The full potential of software-based solutions is not unleashed – some issues have been described in [KTK+21]. The M&O approaches are highly centralised and actually there are ongoing works on distributed architecture for network slicing [KKT+21]. Integrating updated functionalities (new transport, etc.) without modifying the existing protocols is hardly possible. Moreover, the NR nodes interoperability is problematic [MBC22]. From 4G, separation of UP from CP at node level has been introduced. The CP of 5G provides high granularity, programmability, and flexible communication between CP functions. Despite all the mechanisms, the CP of 5G raises some issues. The 5G complexity is high and it seems that in some cases the separation of concerns the architecture is disputable and adding a new feature to one of the protocols requires updating other' "cooperating protocols", what is reflected of new versions of protocols in a 'next Release'. Unfortunately, the 5G ecosystem system is partly closed and the interactions with external subsystems transport, service platforms) are limited. Moreover, 5GC is not yet integrated with MEC or O-RAN [IKC20]. The UP of 5G uses a modified GPRS (General Packet Radio Service) Tunnelling Protocol (GTP) that is criticised for significant overhead and traffic concentration due to tunnelling [BAH+18].

In the context of 6G, it is worth citing [6GFlag20]:

- Modularisation of system functions should minimise the dependencies between them.
- Functions should be defined and partitioned according to which services and what kind of services they provide rather than how they deliver the services.
- The reuse of procedures should be maximised. A procedure can be considered as a service to reuse the interactions from one function to another.
- The control functions and enforcement functions should be separated to allow independent implementation, deployment, scalability, and customisation.
- The framework (or platform) functions and radio-service-specific functions above the platform should be decoupled so that the radio-service-specific functions can evolve independently and much faster than the framework.
- There should be support for on-demand ”stateless” control functions, whereby the usage and storage of context are separated.
- There should be support for on-demand implementation and deployment of functions to allow the flexibility to achieve the balance of flexibility and efficiency.
- 6G might provide an opportunity to go towards a genuinely access-agnostic CN, where mobility management and access management functions can be deployed independently from each other.

The 6G comes with diverse UP requirements regarding the delay, reliability, etc. Such 6G requirements are listed in [HEX-D13]. Some possible methods to achieve high reliability are simultaneous data forwarding combined with multipath, node and path disjoint routing, Deterministic Networking (DetNet), TSN, and segment routing [ST20]. An open issue is whether a single transport network can fulfil 6G requirements or whether multiple dedicated transport networks should be used. The proposed solution is a system created out of multiple, self-managed Functional Domains (FD). The domains can be of the same or different types (access, transport, etc.), of the same or other access technology (4G E-UTRA, 5G NR, WiFi), have a unique ID, and a definition of the area served. The FDs are self-described; however, some domain types can be predefined. The FD domain has a different meaning than the 3GPP domain. For example, *Radio Domain #n* is just a set of RAN nodes of a specific RAT covering a particular area, not the whole radio domain. The size of an FD can be optimised, and the size depends on multiple factors. A desired feature would be the dynamic resizing of FDs based on predefined KPIs. To cope with technological differences, high-level APIs (i.e., intents) between the FDs are proposed. The role of these intent-based APIs is to translate high-level, inter-domain protocol primitives into domain-specific atomic protocol operations. Each FD is composed of modules designed to minimise the inter-modules signalling. The federation of modules can be defined and deployed dynamically, similarly to a network slice. Such an approach provides the ability to modify the system's internal blocks (modules) without changing other cooperating components of the system. The solution uses the classical split of a system into user, control, application, and management planes. Planes are composed of highly granular modules implementing a single goal (mobility management, resource allocation).

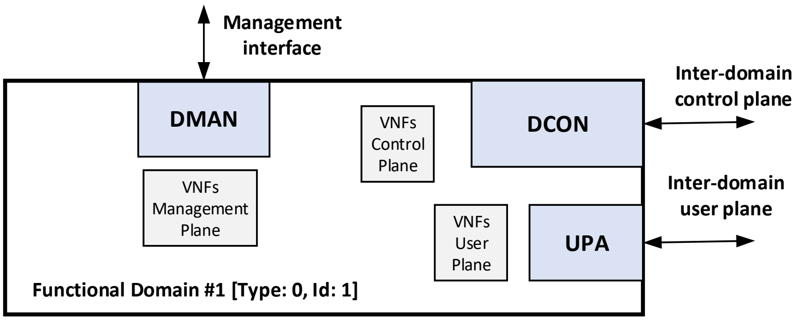


Figure 4-14: Functional Domain concept.

Each FD has a Domain CONtroller (DCON) that is responsible for CP operations (near-rt RIC of O-RAN, SDN controller fits this category), User Plane Adapter (UPA), and Domain MANager (DMAN), cf. Figure 4-14.

DCON is mainly used for the FD CP interaction with other FDs - it provides an abstraction of the domain to the external world. FD translates high-level orders obtained from other FDs into a set of local actions and exposes to other FDs an abstracted view and status of its domain. The DCON functionalities are split into layers focused on specific operations, i.e., resource management, mobility management, etc. The entities of DCON layers of different FDs can cooperate to achieve their goal. For example, mobility management entities can be deployed in multiple FDs that are chained, and they may interact. The functionality of CP can be orchestrated. For CP programmability, DCON may expose services similarly to MEC/ Multi-access Edge Platform (MEP) (Radio Network Information Service (RNIS), etc.).

DMAN is responsible for the overall management of its FD. The management is automated and exhibits a high-level, intent-based management interface. The management functions include Fault, Configuration, Accounting, Performance, Security (FCAPS) and the DMAN services (including security) are programmable – they can be orchestrated. DMAN is also responsible for triggering orchestration requests concerning its domain application, CP, and UP functions. For that purpose, it interacts with the resource orchestrator (see description below). It exposes abstracted management information and the management policies/reconfiguration in the form of intents. To achieve its goals efficiently, the DMAN should be AI-driven and implement control-loop-based real-time management. DMAN functions include dynamic resource discovery and self-configuration. The usage of DMAN in the case of solutions composed of multiple DFs will be described later.

UPA is an optional entity that can be a part of the UP of each FD. Its primary role is data adaptation at the UP, and it may implement application-level data conversion, etc. There can be multiple UPAs per FD, and they may be orchestrated.

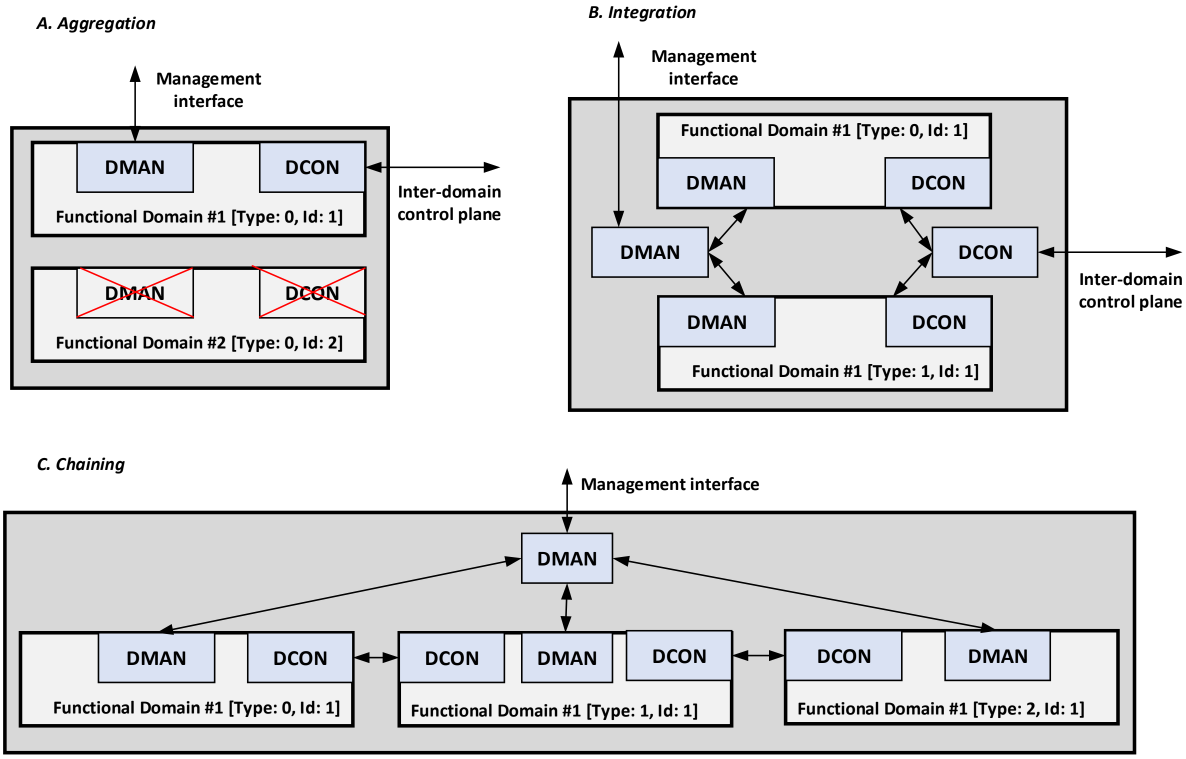

A. Aggregation
Management interface
DMAN
DCON
Functional Domain #1 [Type: 0, Id: 1]
Inter-domain control plane
DMAN
DCON
Functional Domain #2 [Type: 0, Id: 2]
B. Integration
Management interface
Functional Domain #1 [Type: 0, Id: 1]
DMAN
DCON
DMAN
DCON
DMAN
DCON
Functional Domain #1 [Type: 1, Id: 1]
Inter-domain control plane
C. Chaining
Management interface
DMAN
DMAN
DCON
Functional Domain #1 [Type: 0, Id: 1]
DCON
DMAN
DCON
Functional Domain #1 [Type: 1, Id: 1]
DCON
DMAN
Functional Domain #1 [Type: 2, Id: 1]

Figure 4-15: Functional domains (A) Aggregation, (B) Integration, (C) Chaining. UP interconnections have been omitted for clarity.

The E2E solutions are to be built using multiple FDs. The following operations on FDs are identified (see Figure 4-15):

- **Aggregation.** This is a process of FD of the same technology (e.g., WiFi) grouping. In such a case, the same DCON and DMAN are in use, but the domain size in terms of functions or node is increased (aggregation of resources).
- **Integration.** This is a process of grouping different technological solutions of the same type (i.e., transport, access network) but other technical solutions (for example, E-UTRA, NR, WiFi). The operation's primary goal is to hide the technical difference and aggregate resources of different technical solutions of the same type. In such a case, additional DCON and DMAN are added to existing ones and interact with them. They are responsible for the integrated FDs external information exchange. Integrated group DCON handles CP requests, whereas DMAN is responsible for managing and orchestrating the integrated FDs. Integration of multiple, different transport technologies (IP, TSN) of specific features (time-sensitive, IoT, streaming) may lead to traffic redirection to adequate networking solutions (despite single abstraction, multiple transport networks can be used).
- **Chaining.** This operation involves using different functional domains to create a functionally more complex system (for example, a complete mobile network). If the chain is creating an E2E solution, only the DMAN of a chain is added to the DMANs of DFs that compose a chain. If the chain is forming a partial solution only, a DCON must be added to a chain as it has to be used in subsequent operations. The abovementioned operations are recursive. Multiple E2E chains may share the same domain that is implemented in the PaaS form.

The concept described in the section uses network virtualisation and softwarization to allow dynamic interconnection of different domains to obtain an E2E solution. The modular concept comes with the promise of system design and operation simplification, but many details have yet to be elaborated.

4.5 Demo: FLEXible TOPologies (FLEX-TOP) for efficient network expansion

The FLEXible TOPologies (FLEX-TOP) for efficient network expansion demo (described in D5.1) will deliver insights on the efficiency of the flexible topologies concept. The key benefits will be coverage extensions (in line with challenge: Global Service Coverage), service provision with lower latencies (since local structure is terminated at the infrastructure edge), security (engagement of selected devices, in line with challenge Trustworthiness), and lower energy consumption (at infrastructure, challenge: Sustainability). There will be leverage on mesh/ad hoc/D2D networking, disaggregated devices with the ability to flexibly allocate functionality (management of computing resources), ultra-high spectrum, and on coordination with the infrastructure, e.g., in terms of resources to use. Architecture aspects and data flow issues will also be studied and validated.

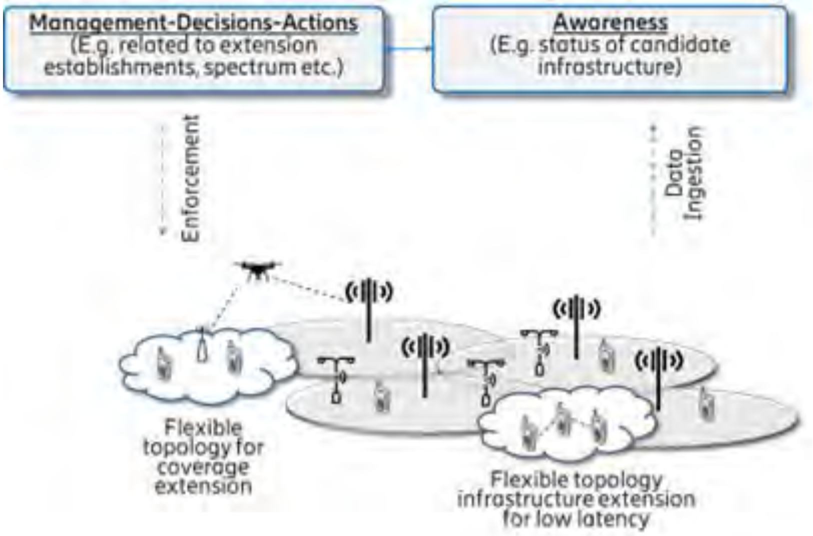

Management-Decisions-Actions
(E.g. related to extension establishments, spectrum etc.)
Awareness
(E.g. status of candidate infrastructure)
Enforcement
Data Ingestion
Flexible topology for coverage extension
Flexible topology infrastructure extension for low latency

Figure 4-16 Flexible topologies for efficient infrastructure extensions demonstration

The scenario comprises the following steps:

- Trigger: traffic increase or starting a sensitive service or mobility.
- Decision to change the topology and go for the local structure; selection of nodes that will be admitted in the "ad hoc" network formation; splitting of functionality between the devices, participating in the local structure, and in an edge (MEC or other) node (which will terminate the structure).
- Local system operation to deliver the service.
- Situation is overcome, structure is decommissioned, and things “go back to normal”.

The architecture in Section 4.1 will be leveraged. The Nodes (including the far-edge devices) are publishing information about themselves as “self-descriptions”, including their expected behaviour, capabilities, resources, and usage policies. This information will be later used by Node Discovery for discovery. The Trust Manager performs continuous monitoring and evaluation of Nodes participants, in terms of dynamic trust evaluations. Then, the Adhoc NW Control selects the best possible nodes and far-edge devices, and it configures the D2D/Mesh formation among the selected Nodes.

Synergies will be pursued for coordinating with the overall M&O (Section 4.1.1) and for incorporating Nodes from various Functional Domains (Section 4.4). In addition, synergies will comprise the consideration of NTN (Section 4.3) and 6G MC (Section 4.2) solutions, as communication capabilities that are supported by specific Nodes. These synergies will be conceptual (captured in the self-descriptions of the various Nodes) but based on the actual capabilities as derived by the results of the relevant research.

The current plan is to have a demonstratable version at the D5.3 delivery time (M28 – end of April 2023), with the aim to demonstrate it in the European Conference on Networks and Communications (EuCNC) & 6G Summit 2023 (June 2023). At the current moment, we are finalizing the architecture and the scope of each component, so as to start some detailed discussions with the partners involved in the synergies described above.

5 Efficient networks

Efficient networks can be characterized in many ways. Here the idea is to define a network architecture that can support all types of traffic anticipated in 6G. However, to support such traffic is not enough, but the efficient network should, for the cases where comparisons with previous generations are possible, be more efficient in terms of, e.g., capacity, coverage, (signalling) overhead, scalability, and energy consumption. Furthermore, we foresee a network constructed with as little dependency as possible among network service enabling, e.g., upgrades of individual network functions with little or no effect on other network functions. In this chapter we explain some changes needed to meet requirements of a future SBA for both the network segments currently associated to CN and RAN. The chapter also includes evaluations to support the proposed changes. Given that the efficient network exists, the chapter also includes examples of services that can be supported. Finally, the chapter includes a discussion on how to evaluate TCO of the future efficient network. This evaluation is an important part since TCO is one very important measure of efficiency and one important KPI for 6G.

In [HEX-D51] a set of KPIs were described which are further developed in this deliverable (see section 6). The intention with the KPIs is to understand what enablers can be used to define an efficient network (see section 6.1.4). In section 5.1 we introduce a method for how to characterize or design network functions, based on the mentioned KPIs, such as separation of concerns and ease of adding new NFs. In the following two sections (section 5.2 and 5.3), this is taken a bit further with descriptions of concrete changes required on top of today's architecture. Section 5.2 proposes the introduction of function elasticity and in particular 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. Signalling procedures that benefit from being in the regional edge cloud comprises 6G mobility management and 6G session management. As a result of localising 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 next section (section 5.3) proposes improvements to interfaces. Many services today require information transfer from one NG-RAN node to another NG-RAN via the 5GC, where the information is relayed via the Access and Mobility management Function (AMF) with limited or even no involvement of the AMF. When using service-based interfaces this information can be exchanged directly between the NG-RAN NFs without the need to pass through the AMF. One possible path to achieve a more efficient network is to remove, e.g., some of the functionalities provided by L2-3 protocols and allow application layer protocols, e.g., TCP or Quick UDP Internet Connections (QUIC), take care of retransmissions and in-order delivery. This is explained in some detail in section 5.4 below together evaluations to support the proposed changes. There is also a section 5.5 presenting a method for how to evaluate the effects on signalling from changes to the architecture. The first examples, based in part on the discussions in the first sections, show that the proposed changes have the capability to reduce time for signalling, which will affect overall latency.

With these proposed changes the current SBA will, in 6G, evolve into an architecture that supports the network segments currently associated to CN and RAN. This provides a range of opportunities for deployment and services. So, in section 5.7 we present how one important service can be optimized, namely CaaS. Finally, a method for how to evaluate TCO for the 6G architecture is described in section 5.6. The idea is to establish a baseline TCO using numbers for 5G Standalone (SA) and compare new (6G) features with this baseline.

5.1 Independent network functions

3GPP defines an SBA in which the CP functionality and common data repositories of a 5G network are delivered by several interconnected NFs, each with authorization to access each other's services via standardized interfaces. Dependencies between individual CN NFs as well as between CN NFs and RAN nodes may create long and complicated (signalling) procedures, with more error cases, race conditions, flavours, etc. Further, dependencies that split responsibilities across different NFs can result in unnecessary complexity and make the system less future proof.

Also, dependencies between NFs may cause unnecessary complexity and even latency. Admittedly, what has been an apparent "unnecessary" complexity may have been the best possible solution since that was the design at that time. However, when designing a future architecture, possible alternatives that result in other outcomes can be tested. So, in this section we continue our investigation, from deliverable [HEX-D51], on dependencies between NFs.

Based on a thorough analysis and characterization of NF dependencies the following candidate actions can be applied to reduce effects of dependencies:

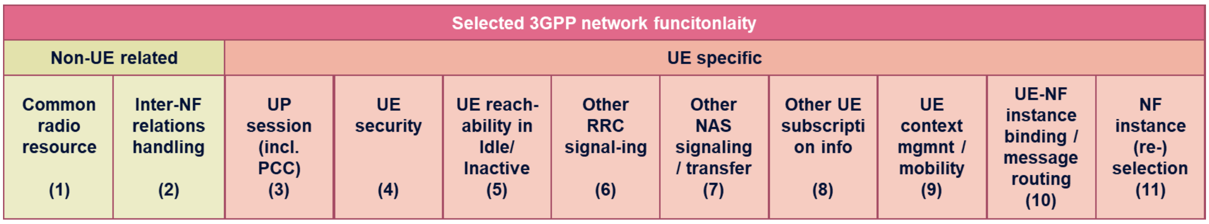
- Merge entities responsible for “radio interface” configuration. Splitting CU and DU adds delay while adding complexity for innovation and optimization of radio performance, see example in the following section.
- Separate UP control to allow direct signalling between UP nodes. This will reduce latency and provide opportunity for vendor optimizations, e.g., co-implementation
- Separate security associations so that each service has its own security, with keys generated by a generic security function. In 5G the AUthentication Server Function (AUSF) holds K_{AUSF} based on which AUSF cloud generate NFs specific keys. .
- Separate subscription, policy handling and UE capabilities for RAN, functions that currently are handled by the AMF
- Separate UE signalling to enable separate association for UP controls, security, etc.
- Harmonize the service framework, e.g., discovery and security, as well as the context handling procedures, e.g., HO, re-establishment, resume, etc.
- Move all state transition functionality closer to the radio or to the UE handler in the RAN to avoid duplicated procedures.
- Remove duplicated sleep states to optimize the state transition procedure so that it does not require bundled service request procedures.

In the following the characterization and remedies are explained with some examples.

In the process of creating independent NFs, we should avoid duplicated functionality, unnecessary options, and multiple processing points. We need to see if there is functionality that can be left out, i.e., a NF will still provide the necessary output/function but in a different way. This naturally shows what functionality that is crucial. When looking at functionality, the interfaces should be also studied. A solution with too many interfaces is costly in many ways, e.g., such a solution may increase time to market for new features in standardization and deployment. Solutions optimized for every specific situation or for a subset of users is also expensive, however, may not always be possible to avoid.

A first step is to structure functions in a way so that they can be analysed (for the sake of identifying dependencies). Looking at 3GPP network functionality the first subdivision could be grouping non-UE related and UE specific functions, as the example in Figure 5-1. The resulting items can then be analysed resulting in function areas, e.g., one UE specific functionality group is UE context management which comprises:

- UE context setup, re-establishment in RAN based on input from CN,
- UE context management in RAN nodes,
- UE context relocation due to mobility,
- UE context relocation due to scaling.



Selected 3GPP network funcitonlaity										
Non-UE related		UE specific								
Common radio resource (1)	Inter-NF relations handling (2)	UP session (incl. PCC) (3)	UE security (4)	UE reach-ability in Idle/ Inactive (5)	Other RRC signal-ing (6)	Other NAS signaling / transfer (7)	Other UE subscripti on info (8)	UE context mgmnt / mobility (9)	UE-NF instance binding / message routing (10)	NF instance (re-) selection (11)

Figure 5-1: Subdivision of 3GPP functionality to identify dependencies.

Next, we characterize the dependencies. Example of dependencies are cross function area dependencies and cross NF dependencies within a function area. The prior can be dependencies between function areas such as UP sessions, UE security, UE context management (/mobility) and UE-NF instance binding/message routing, all of them needed for e.g., Xn mobility. An example of dependencies of the former kind is how a UP session may involve many different nodes, such as, DU, Central Unit – Control Plane (CU-CP), Central Unit – User Plane (CU-UP), AMF, Session Management Function (SMF) and UPF.

Using such analysis on e.g., Xn-HO [23.502], we can identify dependencies that can be studied further. For example, there are security dependencies where Access Stratum User Plane (AS-UP) keys are based on Access Stratum Control Plane (AS-CP) keys while Access Stratum (AS) keys are based on NAS keys. Also, the AMF acts as a proxy forwarding N1 and N2 signalling to/from SMF from/to UE and RAN.

Summarizing the analysis with regards to Xn-HO some possibility for improved separation or grouping could be removing the need for AMF proxy functionalities, by allowing direct communication between UE/RAN and other CN functions. Further, security could be made less hierarchical, e.g., by having independent security associations and key generation.

There is another possible source of dependencies coming from the evolution in network deployments, namely internal RAN split dependencies (Figure 5-2). In the figure PDCP and part of RRC are located in the CU, while RLC, MAC and L1 together with the remaining part of RRC are located in the DU. It is a general misconception that the CU-CP hosts the whole RRC. With this split around 90% of the RRC parameters are determined by the DU and sent in an ASN.1 container to the CU-CP. The CU-CP adds the configuration to an RRC message that is sent via the DU to the UE.

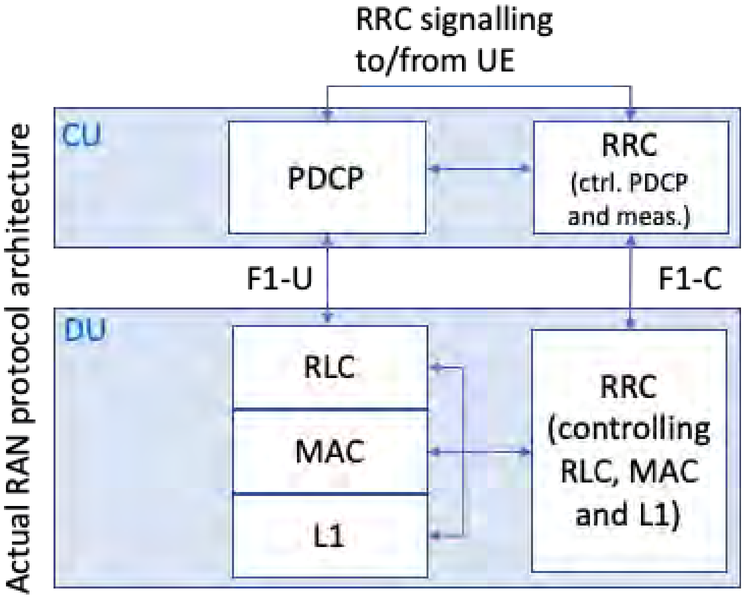


Figure 5-2: Internal RAN dependencies.

The CU handles mobility and selects the primary carrier. The CU also receives the L3 measurements from the UE. However, due to the split of functions it may be difficult to select the best overall primary carrier and also difficult to configure L3 measurements in an optimum way. The DU configures the cell groups, which is a large part of RRC. Further, the DU receives L1 and L2 measurements from the UE. However, since the DU does not have all necessary measurements, it cannot select the primary carrier. It may also be difficult to configure measurement gaps. Thus, it is difficult for the DU to create the best cell group.

To overcome some of the identified hurdles (for functional split dependencies) we need to change communication patterns. Instead of today's very long sequential procedures with several variants we need small independent atomic transactions. Also, we need to remove unnecessary signalling proxy functionality. In other words, we reduce hierarchy and unnecessary dependencies. This can be done by keeping radio related configurations together. The communication needs to be made service-based with loosely coupled services. Finally, it is necessary to optimize time-critical procedures, such as HO, RRC resume, radio link reconfiguration, etc., however, without requiring tight bundling of services.

5.2 Function elasticity

As stated in Section 4.4 "Cloud and Service-Based Architecture" of [HEX-D51], the 5G architecture applies a service-based approach in the CN CP and defines network functions applying service-based principles. However, the SBA concept is only applied to the 5GC, not to the 5G RAN. Cloud-Based SBA was adopted to the CN to facilitate scoped, efficient, and fast means to add new functionality without impacting other parts of the system. Some 5G RAN deployments have adopted virtualization of RAN functionality (e.g., proprietary virtual RAN (vRAN) and cloud RAN solutions) enabling RAN cloudification. With RAN cloudification, it is possible to use a shared edge infrastructure for edge cloud and RAN. This architectural evolution using cloud technology is motivated by introducing 5G radio capacity to new services in the enterprise space [Nok22]. With the continued trend of cloudification of RAN functionality, the question of adoption the SBA approach across 6G-RAN and 6G-CN CPs becomes relevant. Regarding applicability of SBA in the UPF, it should be noted that UPF is a specialized high-throughput, often line speed, GTP-processing and packet forwarding engine for user traffic between N3, N9 and N6 interfaces. Even if UPF has been implemented as a VNF, it doesn't benefit from the plug-and-play nature offered by SBA other than in the interfaces towards CP and network management. Furthermore, there is a performance overhead in the use of Hypertext Transfer Protocol Secure / Transmission Control Protocol (HTTPS/TCP) of SBA to support highspeed user data tunnelling functionality.

6G provides an opportunity to re-visit the interactions between RAN and CN, taking full advantage of the latest cloud technology trends in favour of seamless deployment of SBA across these domains. To leverage the full potential of SBA enabled flexible and agile service developments for the long-tailed distribution of use cases for 6G, the role of NAS termination between network and terminal needs to be studied.

When developing a RAN-CN CP interface to be service based for 6G, the starting point should be to consider the need for dynamic elasticity (e.g., scaling and location) of the services exchanged between 6G RAN and 6G CN and the need to introduce new services. Hardware and latency requirements limit the potential locations of the NFs in a multi-cloud environment, as show in Figure 5-3.

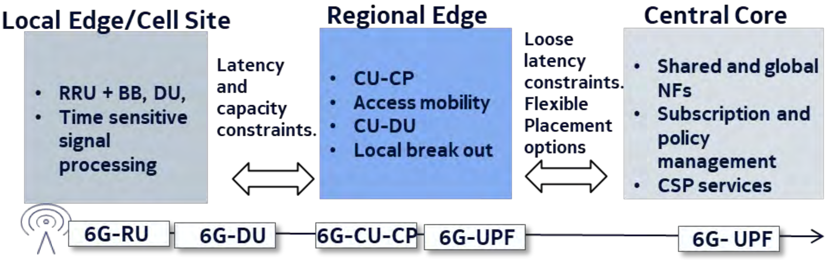


Figure 5-3: Functionality distribution between RAN and CN clouds.

The services across 5G RAN and 5GC interface are classified into UE associated (e.g., UE Context management procedure) and non-UE associated services (e.g., Cell tracing) as defined in clause 5 of [38.413]. Advancing non-UE associated and associated services to be service based poses different challenges to the overall system.

The non-UE associated services are related to the whole RAN-CN interface instance between the 6G AN. They do not need any UE specific context information as opposed to UE associated services. Therefore, the scalability of non-UE associated services is not a major issue.

Communication with the network functions implementing 6G-UE associated services from the 6G-CN network side requires that CN NFs have access to the UE context information which contains, among others, destination node ID of the 6G node containing the service or the functions as well as 6G-UE specific UE-ID keys including security credentials. The UE context with a reference to the target 6G-RAN can be learnt from a 6G-NF that has visibility to the full UE context creation through service discovery and selection process executed before initiating the communication. The scalability of UE associated services needs careful considerations. The first approach 6G-RAN-CN to function elasticity is by co-locating some of the common 6G-CN NFs with the 6G RAN CP in the cloud environment. To improve signalling performance, reduce latency and improve scalability straight forward approach would be to localise critical signalling processing in the regional edge cloud. This is already doable with the current 5G, but the dependencies between the rest of the CN limits its applicability to deployments where both 6G RAN and 6G CN functions are run in the same cloud without means for further distribution. 6G mobility management and 6G session management can co-reside with the 6G RAN CP in the regional/edge cloud whereas the central registrars and databases could remain in the central cloud provided that the UE and session context handling are properly distributed as part of the mentioned NFs. This approach is applicable to 6G-UE associated services as well since the 6G-UE context handling would remain within the control of the 6G mobility management without creating new or additional dependencies beyond the context sharing through e.g., distributed DBs. Furthermore, the mobility messaging between the collocated RAN specific NFs and the rest of the 6G-CN would stay within the mobility management function irrespective whether the functions are located in any cloud in the cloud-native continuum stretching from core to distributed edge. Particularly, deployments that require low latency signalling and reactive local traffic policies would benefit from shortening of the signalling path between the RAN and CN NFs by co-locating mobility management and session management.

In addition to co-locating selected 6G CN and RAN NFs, possibly in the same cloud, 6G NFs can be implemented in a distributed, elastic manner consisting logically of two parts: 1) a centralized part that takes care of global functions, such as authentication and interfacing to central DBs and registers, etc., and 2) distributed part(s) that can be provisioned across the cloud continuum as needed by the orchestrator. Such an elastic and distributed NF will share a common context between its constituent parts. The connection between these parts could be left to be deployment specific issue, or it could leverage the Service Based Interface (SBI) messaging (without the need for distributed DBs). Such elastic NFs will leverage the orchestration mechanisms introduced in section 3.5, i.e., DFP, to provision the distributed parts of the NFs in their optimal locations in the cloud continuum and to co-locate dependent centralized parts close to each other meet the performance and scalability requirements.

To leverage the full potential of SBI, a fully distributed NAS termination (see Figure 5-4) would be needed in 6G, so that even functions in the 6G-UE can signal with the appropriate network functions without having to cross a single point of termination. Therefore, also 6G-RAN and 6G-CN NFs should be able to directly communicate with each other irrespective of their location. The 6G-NAS security associations should be per service to enable direct secure communication between the 6G-UE and the 6G-NFs. A distributed 6G-NAS architecture allows to add and remove new services and functionality in line with SBA principles making the overall system easy to be modified and evolve. Another benefit of distributed 6G-NAS per service security association is to strengthen the security as the service specific security keys are in use only in those NFs that need them.

Serval options exits for how the UE-to-CN NF security associations could be created and maintained. One option is that when a 6G UE is contacting the network it will undergo a subscriber authentication during which the network will assign a service specific context identifier in a similar way to how 5G AMF associates 5G Globally Unique Temporary Identifier to a 5G UE. This way UEs and NFs do not need to share separate per NF certificates.

The cost of the flexibility brought by the distributed NAS is in terms of additional complexity in the UE side, additional interaction of NFs with the "generic security function" to get the keys, additional procedures for refreshing or renewing the keys as well as aspects of exposing a part of network topology to the UE. The details of the how distributed NAS should be implemented requires further security analysis.

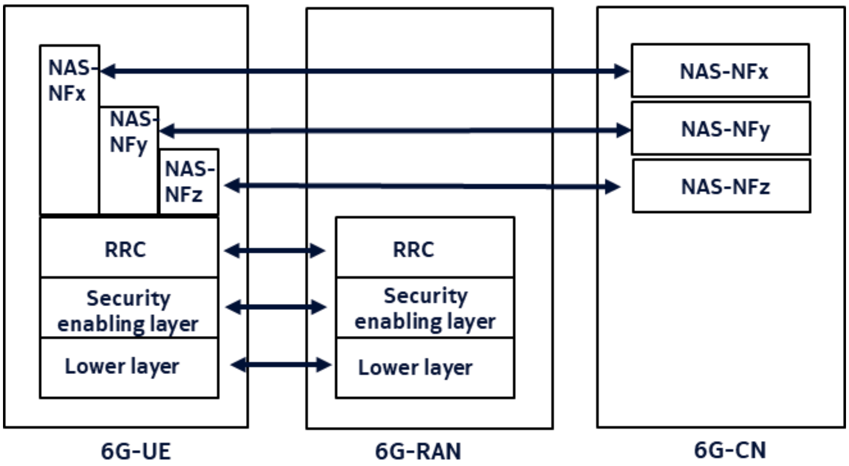


Figure 5-4: Distributed NAS enables per NF service signalling.

5.3 RAN cloudification

For simplifying cloud native CN RAN implementations as discussed in Hexa-X D5.1 [HEX-D51] clause 8.1, it is proposed to replace the existing N2 interface, as specified in 3GPP TS 38.413 [38.413] with a SBI using RESTful APIs (REST stands for REpresentational State Transfer) in alignment with the concepts and protocols outlined in 3GPP TS 29.500 [29.500]. A new NF (NG-RAN) containing CU-CP functionality in alignment with the NF Service Framework, as specified in 3GPP TS 23.501 [23.501] that enables the use of NF services such as:

- NG-RAN service registration and de-registration for making the NRF aware of the available NG-RAN instances and supported services.
- NG-RAN service discovery to enable a NF Service Consumer (e.g., AMF or other NG-RAN NF to discover NG RAN Service Producer instance(s) which provide the expected NG RAN service(s).
- NG-RAN service authorization to ensure the NF Service Consumer is authorized to access the NF service provided by the NG-RAN Service Producer.

Figure 5-5 depicts the proposed architecture supporting NG-RAN as new NF using the reference point presentation. The architecture allows for any access network support including non-3GPP access (not shown in Figure 5-5).

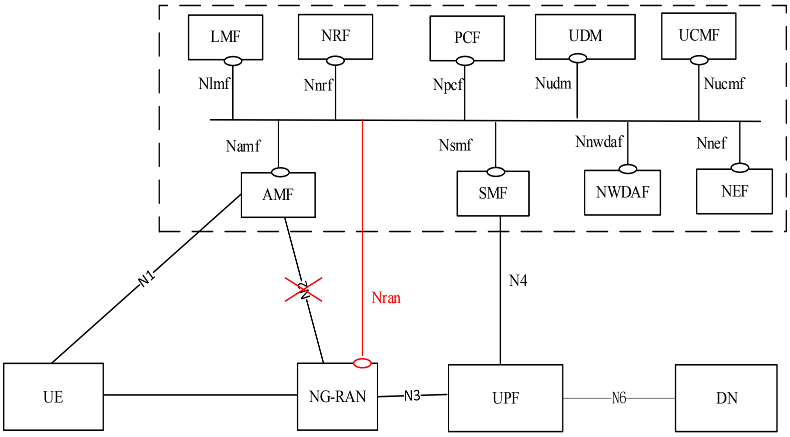


Figure 5-5: Architecture with NG-RAN supporting SBI.

Please note that this section focuses on the evolution of the CN-RAN interface whereas the evolution of other network interfaces is discussed in other sections of this document.

5.3.1 Transport protocol considerations

With the introduction of a service-based interface exposed by NG-RAN towards the CN (replacement of the legacy N2 interface) the underlying transport protocol will be based on Hyper Text Transfer Protocol 2 / Transport Layer Security (HTTP2/TLS) with TCP/IP as specified as part of the SBI Protocol Stack in [29.500]. However, the legacy N2/NG Application Protocol (NGAP) is based on Stream Control Transmission Protocol / User Datagram Protocol (SCTP/UDP) which is highly optimised for Public Switched Telephone Network (PSTN) signalling across the IP network. Potential drawbacks of a TCP based solution need to be carefully considered. For example, compared to TCP, SCTP as specified in RFC 2960 [RFC2960] offers multiple features which are relevant for the transport of PSTN signalling across the IP network such as:

- support for reliable transfer with partial or without sequence maintenance. This is to avoid Head-Of-Line (HOL) blocking issue that may occur in TCP causing unnecessary delay due to strict order-of-transmission delivery of data.
- support for applications to add their own record marking to delineate their messages and to make explicit use of the push facility to ensure that a complete message is transferred in a reasonable time.
- highly available data transfer capability using multi-homed hosts.
- less vulnerability to denial-of-service attacks, such as TCP SYN attacks.

With the introduction of QUIC(HTTP/3) for 5GC Service Based Interfaces as proposed in 3GPP TR 29.893 [29.893], TCP is replaced with QUIC as the transport protocol for HTTP. This may also help to facilitate the introduction of a service-based interface exposed by NG-RAN as with QUIC the following improvements over TCP are expected:

- QUIC allows to overcome HOL blocking among different streams from which HTTP/2 is suffering if a TCP packet is lost or becomes corrupted.
- loss detection mechanisms of QUIC are using more accurate means than TCP to indicate lost bytes and RTT measurements resulting in assumedly more efficient recovery mechanism.

- faster connection establishment compared to TLS/TCP (1 RTT instead of 2), for short lived connections; however, when using persistent connections, this will not lead to a performance improvement.
- integrated support for connection migration to a different network interface or local address by the client during the lifetime of the connection or by the server during the connection establishment.

Even though QUIC is not considered in 3GPP Rel-17, e.g., due to the early stage of the QUIC specifications and related implementations we foresee the introduction of an evolved QUIC as part of 6G.

5.3.2 Use cases and possible benefits

In this section, we focus on existing N2/NGAP use cases and procedures, while keeping the legacy functional RAN and CN work-split as a reference which is not necessary but useful for better understanding. As discussed in section 5.1, the use cases can be divided into non-UE-associated services and UE-associated services. Specific to the non-UE-associated services there are several existing use cases requiring information transfer from one NG-RAN node to another NG-RAN via the 5GC where the information is relayed via the AMF with limited involvement or even no involvement of the AMF. When using service-based interfaces this information can be exchanged directly between the NG-RAN NFs without the need to pass through AMF. Example use cases are:

- Remote Interference Management (RIM) Information Transfer procedure, as specified in 3GPP TS 38.413 [38.413] clause 8.16;
- Configuration Transfer procedure for Self-Optimized Networks (SON) as specified in 3GPP TS 23.501 [23.501];
- Warning Request Transfer procedures as specified in 3GPP TS 23.041 [23.041] clause 9A.

In addition to the existing N2/NGAP use cases as mentioned above also new use cases can be addressed with a service-based interface. For example, with the introduction of RAN intelligence enabled by AI (as described in 3GPP TR 37.817 [37.817]), several new use cases will be specified which require interactions between NG-RAN nodes and O&M and interactions between two NG-RAN nodes for the purpose of AI/ ML input data collection (for model training and inference), AI/ ML model deployment and update and AI/ ML model performance feedback:

- Network Energy Saving as in [37.817] clause 5.1
- Load Balancing as in [37.817] clause 5.2
- Mobility Optimization as in [37.817] clause 5.5

5.3.3 Further optimisations based on an optimised RAN and CN work-split

Table 5-1 illustrates the existing CN-RAN functional allocation split in 5GS as specified in 3GPP TR 23.799 [23.799], which serves as baseline for our investigation:

Table 5-1 CN-RAN functional allocation split in 5GS

Function	NG-RAN	CN
Mobility management		
Mobility management control, (Subscription and Policies)		X
Determination of mobility restriction		X

Roaming restrictions execution		X
Mobility restrictions execution, [CN Connected]		X
Mobility restrictions execution, [CN Idle]		X
UE registration		X
UE unreachability detection		X
RAN UE unreachability detection	X	
NAS state transitions		X
RRC state transitions	X	
Paging initiation and control in RAN Inactive state	X	
Paging initiation in CN Idle state		X
Access Stratum UE Context storage in RAN Inactive state	X	
Control of connected state mobility	X	X
UP buffer for UE in CN Idle state		X
UP buffer for UE in RAN Inactive state	X	
Session Management		
PDU Session address allocation		X
Session Management		X
Termination of UP security	X	
Subscription Data Handling (incl. default QoS profile)		X
Authentication and Key Agreement		X
QoS		
Radio Resource Admission Control	X	
Radio Resource management (QoS attributes)	X	
QoS Policy Control		X

In the table above there are functions available in NG-RAN and CN having a similar purpose such as unreachability management and paging. Assuming NG-RAN is accessible inside the CN those duplicated functions could be merged.

Unreachability detection and paging is needed in RAN as part of the RRC INACTIVE state. The RRC INACTIVE state was introduced in NR to minimize CP latency and UE power consumption during state transition. Compared to RRC IDLE state in RRC INACTIVE state the CN/RAN connection is kept together with the UE's AS context. Figure 5-6 illustrates the RRC/ Connection Management (CM) / Mobility Management (MM) states in 5GS:

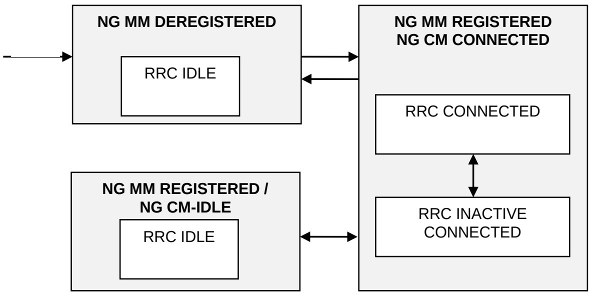


Figure 5-6: RRC/CM/MM states in 5GS

With the proposed removal of CN UE unreachability detection and paging and the introduction of a service-based interface exposed by the RAN (i.e., the CU-CP) direct request from NFs (e.g., SMF, Location Management Function (LMF), ...) towards RAN could be facilitated without AMF involvement. A centralized CU-CP can serve as the mobility anchor towards the CN. The CU-CP may combine RAN-based Notification Areas (RNAs) managed by the gNBs to larger RNAs similar to the size of tracking areas managed by the AMF. With this approach CN signalling and CP latency can be further reduced as the UE can remain in RRC INACTIVE state for a potentially longer period of time. In the extreme case the UE is kept in CM-CONNECTED state as long as the UE is registered. This would mean the complexity in CN and UE can be reduced by removing CN based unreachability management together with the CM-IDLE/RRC IDLE state.

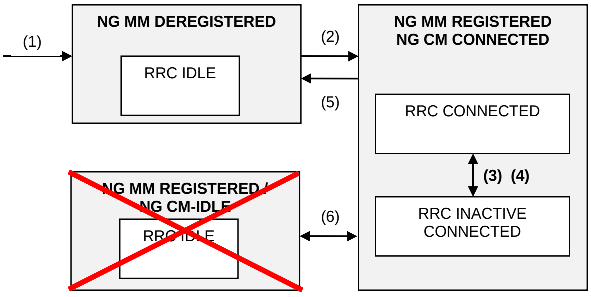


Figure 5-7: RRC/CM/MM states without CM-IDLE/RRC IDLE

However, using larger RAN-Notification areas also has some trade-offs regarding the number of UE's which can be served within an RNA due to the size of the Inactive Radio Network Temporary Identifier (I-RNTI) (e.g., up to 48 bit) and a potential signalling overhead between gNBs. The I-RNTI is used to identify both the UE and the gNB which hosts the UE context. Furthermore, the impact on energy saving (e.g., UE power consumption) needs to be further analysed (not part of this document).

5.4 Performance of transport protocols over mobile networks

Transport protocols such as TCP and UDP [KR12] provide communication services for applications while abstracting the underlying networking technology. TCP is one of the most popular transport protocols providing services such as reliable and in-order delivery. It was first designed for fixed networks over the Internet to handle small amounts of packet losses and out-of-order delivered packets. When Internet was introduced over the cellular links, to be spectrally efficient the cellular network needed to allow a higher packet loss rate than fixed networks. However, in the early versions of TCP all packet losses are interpreted as congestion in the network. Another problem is also related to out-of-order packets. Excess number of out-of-order packets arrived at TCP is also interpreted as congestion, resulting in congestion control mechanism to reduce the sending rate. To solve this problem the 3GPPstandardization introduced Acknowledged Mode (AM) with local retransmissions of packets and in-order delivery of the packets. These services hide the packet losses and out-of-order packets from TCP at the cost of increased delay and hardware/algorithmic (e.g., buffer) complexity needed by those services. At the time this was a very good trade-off because the delayed or out-of-order packets were not incorrectly interpreted as congestion.

However, over the years transport protocols have also evolved. Equipped with features such as Selective ACKnowledgement (SACK) [FMM+96], Fast Retransmission and Recovery (FRR) [Ste97] and Recent ACKnowledgement (RACK) [CCD+18] they are better at handling losses and out-of-order packets. Moreover, additional enhancements such as reduction in the overall E2E latency and applications hosted in operator environments e.g., edge, bring closer the points (RAN/network, application) where functionalities such as retransmission of lost packets could be performed.

This motivates to revisit the impact of UP features and functionalities on the performance of end point service. More specifically, as shown in Figure 5-8, we investigate the effect of RAN UP in-order and reliable delivery services to see how the end points are impacted. The findings will serve to understand whether it is possible to simplify the architecture while maintaining good performance at the end points.

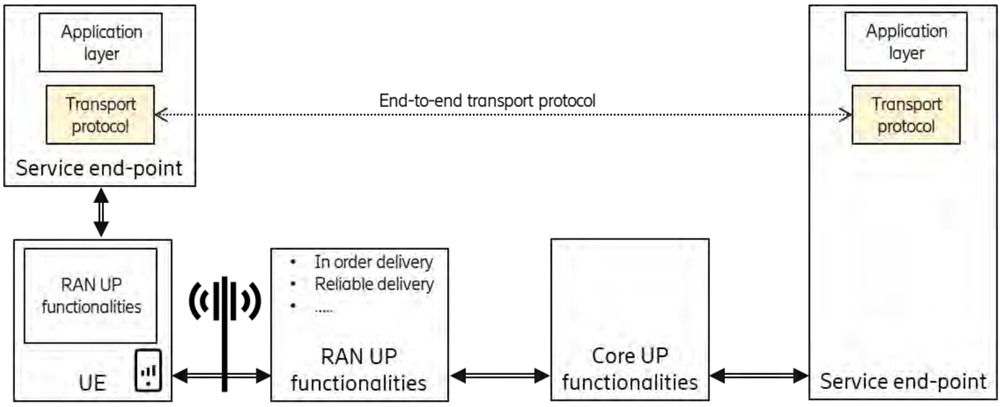


Figure 5-8: RAN UP providing in-order and reliable delivery services to transport protocols.

We study the proposed research question in a simulation scenario with a deployment that closely models future deployments. Specifically, the setting involves low latency transport network from the site to the application server, large bandwidths and CA resulting in low latency and high data rates. To evaluate the E2E performance, as shown in Figure 5-9, we consider a 5G SA NR deployment with 4 carriers.

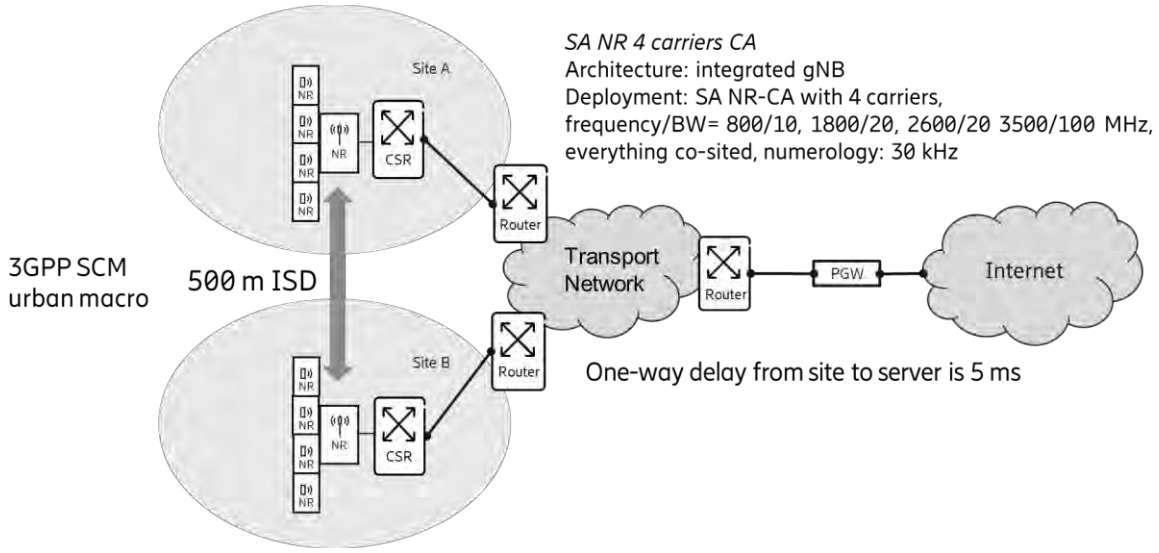

SA NR 4 carriers CA
Architecture: integrated gNB
Deployment: SA NR-CA with 4 carriers,
frequency/BW= 800/10, 1800/20, 2600/20 3500/100 MHz,
everything co-sited, numerology: 30 kHz
Site A
Site B
NR
CSR
Router
Transport Network
PGW
Internet
3GPP SCM urban macro
500 m ISD
One-way delay from site to server is 5 ms

Figure 5-9: The deployment for the simulation. The simulation involved two NR SA gNBs each equipped with 4 carriers: a 10 MHz DL + 10 MHz UL carrier on 800 MHz, a 20 MHz DL + 20 MHz UL carrier on 1.8 GHz, a 20 MHz DL + 20 MHz UL carrier on 2.6 GHz and a 100 MHz carrier on 3.5 GHz.

The simulation is intended to be as realistic as possible with detailed channel models for the physical layer, realistic HARQ with timing offsets aligned with the standard, as well as NR compliant PDCP and RLC layers. The transport protocol at the endpoints is TCP with CUBIC [HRX08] used as congestion control algorithm. Data users are generated into the system according to a Poisson distribution where each user upon arrival connects to the File Transfer Protocol (FTP) server and downloads a 10 MB file.

We simulated 4 different scenarios: 1) RLC AM bearer and in-order delivery, 2) RLC AM bearer and out-of-order delivery, 3) RLC Unacknowledged Mode (UM) bearer and in-order delivery, and 4) RLC UM bearer and out-of-order delivery, iterated over user arrival rate to observe the effect of in-order delivery and reliable transmission provided by PDCP and RLC layers, respectively. Figure 5-10 illustrates the perceived object bitrate at the application side with respect to number of user arrivals per second. First, it is observed that there are no significant performance variations across the different scenarios (4 % degradation due to out of order delivery and 5.5% due to UM delivery in the worst case.). This implies that the modern TCP with a properly tuned congestion control algorithm is capable of handling packet losses and out-of-order delivered packets very well. Second, we observe that AM performs slightly better than UM confirming that frequent packet losses are better to be handled close to the RAN. Third, we see that there is still a small degradation when in-order delivery is not provided by the RAN stack.

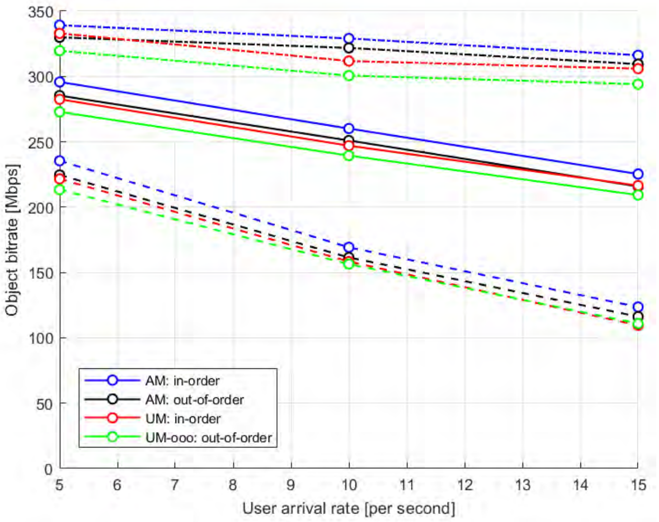


Figure 5-10: The object bitrate perceived at the application side for different scenarios and user arrival rate settings. The solid lines represent the mean value, dashed line represent the 10th percentile and dot dashed line represent the 90th percentile.

To conclude, it is observed that features such as in-order delivery and reliable transmission, which were once paramount in RAN, are not as essential in modern networks with advanced end point protocols. These results open windows of opportunity for simplification of the RAN stack, either in terms of removal of functionalities or simplifications in their behaviour and move some functionalities of UP towards the end points and alleviate duplication. The results still need to be verified for a wider range of settings and KPIs such as latency and jitter. As future work, other transport protocols such as QUIC and congestion control algorithms like the Bottleneck Bandwidth and Round-trip propagation time (BBR) will be studied. Moreover, we need to study other traffic types such as UL/DL video, small objects, and large web pages.

5.5 Efficient signalling performance in 6G architecture

Efficient signalling may be characterized in different ways, e.g., when comparing to a baseline signalling becomes faster, or signalling consists of fewer messages. In this section we look at both aspects but measure efficiency as a reduction of latency. In the following we show how this can be done using the principles introduced in the previous section.

In D5.1 [HEX-D51] we introduced changes to signalling with the objective of increasing efficiency. We saw that, for instance, latency has been a driver for how different layers or functionalities have been bundled together today for service request procedures. Separating these functionalities could potentially make the architecture cleaner, and cloud friendly as well, but it could also lead to higher latency due to more handshakes among separate functionalities. So, in some cases cloud friendliness could lead to increased latency.

As mentioned in the previous section, the process of making the architecture efficient is preferably done by scrutinizing important and often used procedures, such as service requests, state transitions and handovers. Some options for reduce latency in service request handling could be to:

- Bundle more functionalities together, e.g., have a more "single controller" approach. It is not clear though that this always solves the problem since if we for instance bundle RRC / NAS we then instead separate the RRC from MAC

- Another option for the service request would instead be to try to rely on the RRC INACTIVE state and store the full RAN/CN configuration of the UE. In this way it would be possible to resume the connection faster even if functionality is more clearly separate.

Bundling of functionalities may seem like a contradiction since we in previous work [HEX-D51] argued for independent NFs. However, by bundling functions that serve a particular UE the need for signalling between NFs and transfer of, e.g., UE context and other UE data, are reduced.

In the following we investigate several variants of HO execution procedures and calculate the total (relative) execution time. Figure 5-11 shows the baseline HO procedure (left figure), including signalling between AMF, SMF and UPF for a path switch. The execution time comes both from the execution in each node of signalling messages, and due to the delay between different nodes. The right figure shows an example where the AMF and the SMF are combined (due to e.g., a cloud-based CN). In this example we assume that the PDU update command is not needed and therefore we ideally set this internal execution to take zero ms. The total time for the procedure can be decreased roughly 20-25% compared to normal HO.

Figure 5-12 shows another example when we assume that parallel execution is possible. Here we assume the AMF, SMF and the UPF are all independent NFs so the target gNB can send all signalling in parallel. The gains are here rather similar, around 20-25%. Note that there may be other sources of latency, e.g., from handling failure cases, prioritization and sequentially among messages sent in parallel.

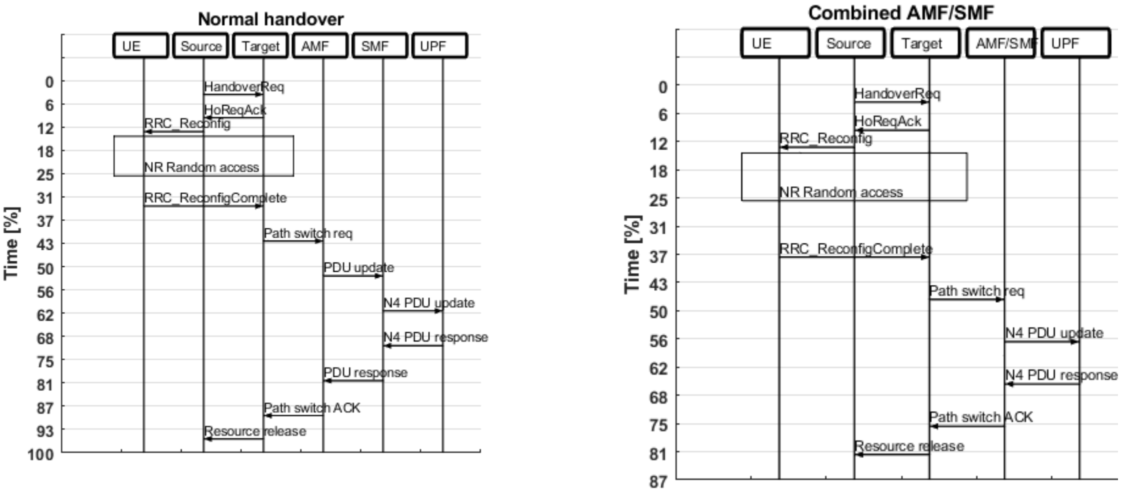


Figure 5-11: Normal HO (left) vs HO with combined AMF and SMF (right), the execution time is normalized vs. the normal HO case.

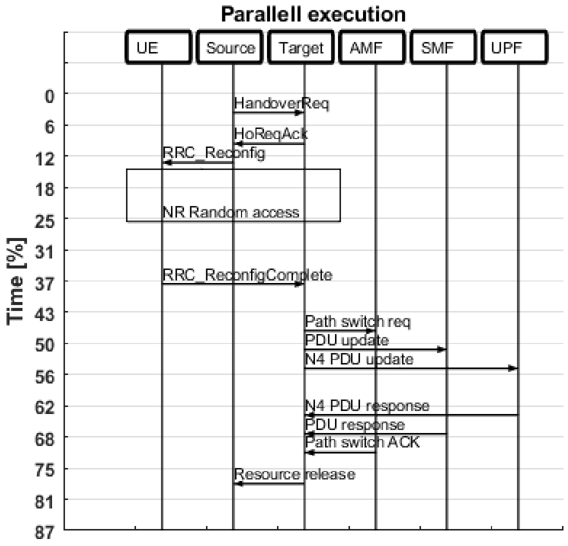


Figure 5-12: Parallel execution example. Here we assume the AMF, SMF and the UPF are all independent NFs so the target gNB can send all signalling in parallel.

5.6 Total Cost of Ownership (TCO) aspects

In [HEX-D51] some initial considerations regarding the TCO of a mobile network were provided, by specifying the cost structure that a mobile network operator has to deal with in terms of both Capital Expenditure (CapEx) and OpEx. Moreover, some insights on the possible future directions on the TCO evaluation for 6G were proposed, requiring the identification of a *baseline* mobile network architecture with respect to which the most promising network enablers characterizing the 6G architecture are evaluated in terms of cost benefits when deployed. In this sense, the TCO is to be evaluated in relative terms (i.e., percentage of cost savings) with respect to the baseline architecture. In a broad sense a baseline architecture – commonly referred to as *as-is* architecture – represents *the set of products that portray the existing enterprise, the current business practices, and technical infrastructure* [FCI01]. Therefore, when applying this definition in the context of mobile networks and more specifically to the TCO definition for 6G networks, the baseline architecture which needs to be taken into account is inevitably 5G.

3GPP Rel-15, that is, the first 5G release of technical specifications, introduced multiple architecture flavours [GSM18] by exploiting a novel approach based on which network elements of different generations – basically 4G's Evolved Packet Core (EPC) and E-UTRA, both with proper specified enhancements – can be integrated with the newly specified 5GC and access technology (NR) in different network configurations (also referred to as *options*). Such network configurations are classified into two main categories: SA and Non-Standalone (NSA), where the former comprises those architectures encompassing one radio access technology only (i.e., either E-UTRA or NR) while the latter includes architectural options making use on up to two radio accesses (i.e., both E-UTRA and NR) combined in a DC fashion – refer to Figure 5-13.

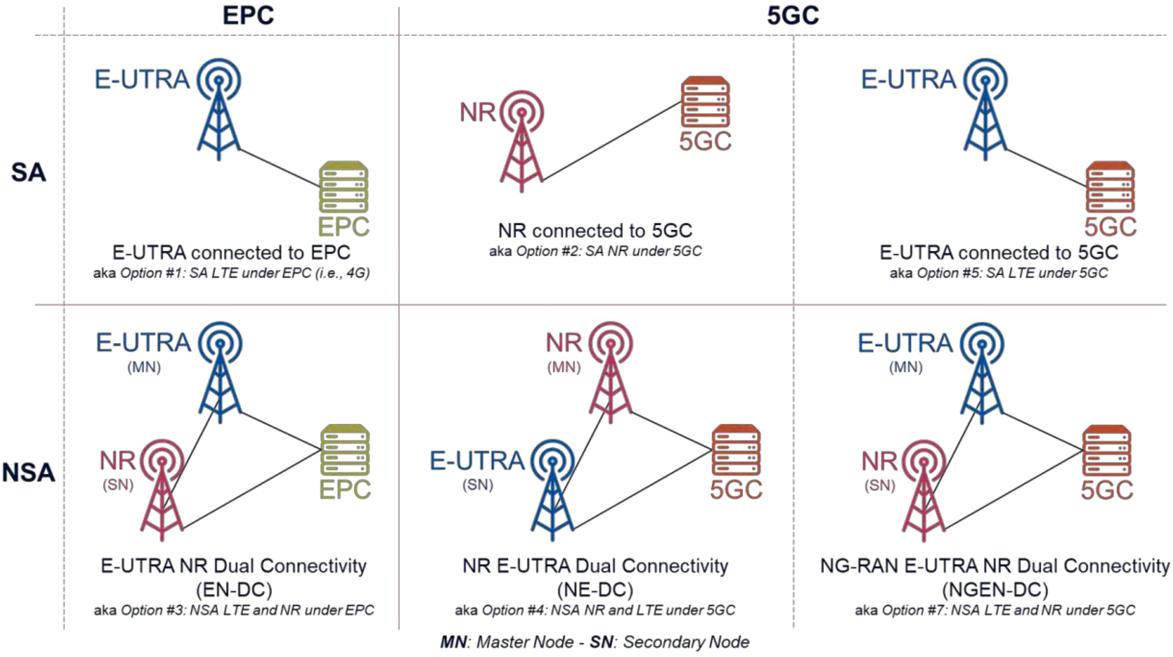

EPC
5GC
SA
NSA
E-UTRA
EPC
E-UTRA connected to EPC
aka Option #1: SA LTE under EPC (i.e., 4G)
NR
5GC
NR connected to 5GC
aka Option #2: SA NR under 5GC
E-UTRA
5GC
E-UTRA connected to 5GC
aka Option #5: SA LTE under 5GC
E-UTRA
(MN)
NR
(SN)
EPC
E-UTRA NR Dual Connectivity
(EN-DC)
aka Option #3: NSA LTE and NR under EPC
NR
(MN)
E-UTRA
(SN)
5GC
NR E-UTRA Dual Connectivity
(NE-DC)
aka Option #4: NSA NR and LTE under 5GC
E-UTRA
(MN)
NR
(SN)
5GC
NG-RAN E-UTRA NR Dual Connectivity
(NGEN-DC)
aka Option #7: NSA LTE and NR under 5GC
MN: Master Node - SN: Secondary Node

Figure 5-13: SA and NSA 5G architecture options.

Because of the availability of multiple architectural choices there are several possible *migration paths* operators can follow to first introduce 5G and then migrate it to the target configuration(s) reflecting their own market requests. Operators started deploying 5G networks based on NSA Option 3, i.e., EN-DC as in 3GPP terminology [37.340], where the NR radio access provides additional UP resources to UEs served by the E-UTRA access and connected to the EPC. EN-DC allowed 5G to be brought to market quickly with minor modifications to the existing 4G network but, on the other hand, it does not introduce and exploit the new 5GC and therefore it is not optimised for 5G use cases beyond mobile broadband [GSM19a].

From a long-term TCO perspective, a recent study [Pao21] concluded that operators can save costs up to 36% by moving to a cloud-native converged 5GC – i.e., a CN being able to support multiple wireless generations (e.g., from 2G to 5G) – with 5G NR SA (Option 2) support as they deploy their 5G network, instead of relying on legacy EPC technologies and EN-DC support. This is because, even though pushing the transition to the converged 5GC into the future delays most of the operator's investments (CapEx), it does not eliminate the investment and it increases the OpEx since the operator has to simultaneously manage both the legacy CNs (i.e., the legacy EPC, the EN-DC capable EPC, etc.) and the newly introduced 5GC. Key features of the converged 5GC solution include 1) cloud-based, virtualized, containerized architecture based on microservices; 2) use of Commercial Off-The-Shelf (COTS) hardware; 3) UPF footprint reduction (due to the containerization of the UPF software) and 4) use of AI-based automation tools. Hence, a fast transition to a converged 5GC solution (with 5G NR SA support) allows the operator to roll out new services fully leveraging the 5GC capabilities and features while still being able to provide services to legacy UEs.

Considering that EN-DC is not the target 5G configuration for most of the operators and that there is not much interest from the mobile industry in making available those architecture options different from 5G NR SA and EN-DC, it is deemed natural to assess the TCO for 6G with respect to the 5G NR SA architecture. The latest 5G NR SA deployment status report released in December 2021 by the Global mobile Suppliers Association (GSA) [GSA21] states that 20 operators in 16 countries/territories – out of 487 operators in 145 countries/territories - that were investing in 5G – actually deployed/launched 5G NR SA in public networks. Therefore, it is expected that those operators that have 5G NR SA deployed/launched will look for ways to fine-tune their 5G deployments, monetise existing and plan new use cases as well as optimise the 5G network's cost structure. The latter objective is of utmost importance for the 6G TCO evaluation in Hexa-X and could be subject to continuous refinements due to the "feedback" coming from the live network operation and new emerging market needs.

An initial work on the assessment of the cost structure for a 5G NR SA network – the baseline architecture for the 6G TCO from the Hexa-X perspective – has been performed by GSMA and reported in [GSM19a]. Such work considers the dynamic interplay of a diverse mix of factors broadly falling into three groups:

a) *Cost drivers*, representing the "reasons why" a new (5G) network is needed, such as the mobile data traffic growth, the (operator-specific) strategy choices in terms of use cases being exploited for monetization, etc.

b) *Cost accelerators*, that is, factors such as the Radio Access Network (RAN) and the backhaul upgrades, the Edge Computing deployment, etc., which increase the overall cost of owning and operating a (5G) network – and that can be classified as being CapEx or OpEx – due to the need to cope with the presence of multiple cost drivers

c) *Cost optimisers*, which can serve as a catalyst to accelerate the (5G) network evolution while keeping the TCO at an affordable level from the operator's perspective; cost optimisers include new RAN architectural approaches, e.g., vRAN instead of legacy distributed RAN (D-RAN), architectural enablers such as automation and AI for planning and executing modern mobile network operations, low energy and or CO_2 reduction solutions such as liquid cooling replacing air conditioning for the equipment, etc.

The GSMA work takes 4G as the baseline architecture for evaluating the cost changes in terms of **RAN infrastructure**, **backhaul**, **CN infrastructure**, **energy** and **other network costs** (people, network management and maintenance, etc.); such cost changes are evaluated in relative terms, i.e., *x*% cost variation for each cost item with respect to the 4G reference case and considering not only the impact of the most significant cost accelerators but also how the adoption of a certain item-specific cost optimiser reduces the effect of the concerned cost accelerator. As an example, when considering the RAN infrastructure – accounting for almost 60% of the network TCO, averaging RAN's typical CapEx and OpEx over a 10-year period [Gha20] – in order to ensure coverage, capacity and an increase in network performance (i.e., throughput and latency) there is the need to upgrade the RAN so to properly cope with the ever increasing data traffic foreseen in the 5G era (from 2018 to 2025), hence leading to an increase of the overall TCO, with significant impacts also on the network energy consumption up to 2-3 times higher with respect to 4G if left unoptimized. Such upgrades typically include 5G-specific enhancements to existing 4G macro-cells, e.g., massive MIMO (mMIMO) installations, as well as network densification with additional 5G small and macro-cells (or macro-cells alone), provided that ElectroMagnetic Field (EMF) emissions' related requirements are met. Hence, mMIMO and 5G network densification represent the most significant cost accelerators impacting the RAN infrastructure, whose effect on the RAN infrastructure specific overall cost can be mitigated by deploying the Centralized RAN (C-RAN) / vRAN as a cost optimizer. Clearly the impact of cost accelerators and optimisers should be considered also for the other cost items considered in the TCO: still referring to mMIMO and RAN densification, these cost accelerators negatively impact not only the RAN infrastructure but also (at least) the network's energy consumption and backhaul. Moreover, even though v/C-RAN vendors typically forecast RAN infrastructure savings of up to 45% [Pao17], such cost optimiser requires high capacity, low latency fronthaul links (typically expensive high-end fibre links), which offset some of the RAN infrastructure savings when assessed on a TCO basis.

An important outcome of the GSMA work is that the 5G rollout strategy choice impacts significantly the relative TCO evaluation with respect to 4G (the so-called *TCO Delta* as in the mentioned study); in its study GSMA considers three core 5G deployment strategies:

a) *Rapid, full scale 5G deployment*: a 5G rollout covering 80% of the population with high capacity 5G network by 2025 and targets new 5G use cases in both enterprise and consumer market segments

b) *Enterprise-focused 5G deployment*: a fast-paced deployment covering 65% of the population with high-capacity 5G network in enterprise hubs by 2025 to address existing and selected new enterprise-specific use cases

c) *Capacity-backfilling 5G deployment*: a measured 5G deployment covering 50% of the population with additional 5G capacity by 2025, targeting existing use cases, capacity backfilling and Enhanced Mobile Broadband (eMBB) services.

Each of the above 5G deployment strategies is associated with a hypothetical Compound Annual Growth Rate (CAGR) in terms of data traffic with respect to 4G baseline – refer to the table in Figure 4 in [GSM19a]. The data traffic growth is a significant influencer of 5G investment requirements since higher network investments are needed to maintain a given level of network performance (e.g., capacity): typically, a new generation will lead to higher TCO, however trying to achieve the same extended capacity with the previous technology generation will lead to even much higher TCO. This means that the TCO evaluation needs to be normalized against the data traffic growth (e.g., TCO/Gbps) when comparing against a previous baseline generation.

For each deployment strategy, the TCO relative assessment has been performed with and without the introduction of cost optimisers for each of the considered cost items, i.e., RAN infrastructure, backhaul, CN infrastructure, energy, and other network costs. For instance, the adoption of the strategy based on rapid, full scale 5G deployment is, of course, costly for an operator: it can lead to an overall 5G network TCO Delta of up to 71%, as depicted in Figure 6-4 (see *5G Baseline Case* column). Such significant costs for the operator can be somehow mitigated by deploying proper 5G cost optimisation tools whose aggregate effect is a reduction from 71% to 39% of the overall 5G network TCO Delta (see *5G Optimised Case* column within the dotted box in Figure 6-4).

A 5G cost difference has been determined for each of the considered cost item: among them, energy has the highest cost difference of up to 140% as a result of mobile data traffic growth and subsequent RAN densification and mMIMO techniques deployment. Aggressive energy optimisation techniques may reduce the energy cost difference to 70%: these techniques include 1) reducing AC/DC conversion; 2) placing base stations in sleep mode; and 3) the use of AI in conjunction with Data Centre Infrastructure Management (DCIM) tools. Hence, the study highlights that energy is currently the most significant cost optimisation priority for 5G vendors and operators alike, pushing the energy component from 23% of the 4G network TCO to up to almost a third in the 5G case. Clearly this trend could also be still valid for 6G, motivating the collaboration with task T1.6 of the Hexa-X project on this critical matter.

RAN infrastructure has the second-highest cost difference with respect to 4G (45-65%), linked to the extent of network sharing, densification and RAN virtualisation. This implies that RAN infrastructure remains one of the most prominent mobile operator network cost components within the overall TCO (accounting for almost 50-60%). Backhaul cost difference broadly follows the RAN cost dynamics, but with a tighter range of 45-55%, a result of more limited opportunities for operators to mitigate costs in this area. Backhaul is therefore likely to stay within the 10-12% range as a proportion of network TCO. Regarding the CN infrastructure, it results to be the smallest and most stable cost component, with a cost difference with respect to 4G of -10% to +10% (this is broadly similar across all three deployment scenarios). In combination with the significant cost difference of other network cost components, the CN share of the overall network TCO in this scenario reduces from 10% to 6% in the *5G Optimised Case*. Finally, the other network costs (people, network management and maintenance) cost difference has a wide range of -20% to 10%, with the level of Automation deployed being a key variable. The methodology developed within the Hexa-X project allowing to achieve the objective of an overall reduction of TCO of at least 30% with respect to the 5G NR SA architecture is explained in section 6.1.5 of this deliverable.

5.7 Further architectural elaborations on the CaaS paradigm

The European Telecommunications Standards Institute (ETSI) Technical Committee Reconfigurable Radio Systems (RRS) has introduced a Software Reconfiguration Framework including a definition of a key Interface, i.e., the generalised Multiradio Interface (gMURI) [303681-1]. The basic principle is illustrated in Figure 5-14. Indeed, in the current approach, computational tasks can be distributed to (local) compute resources.

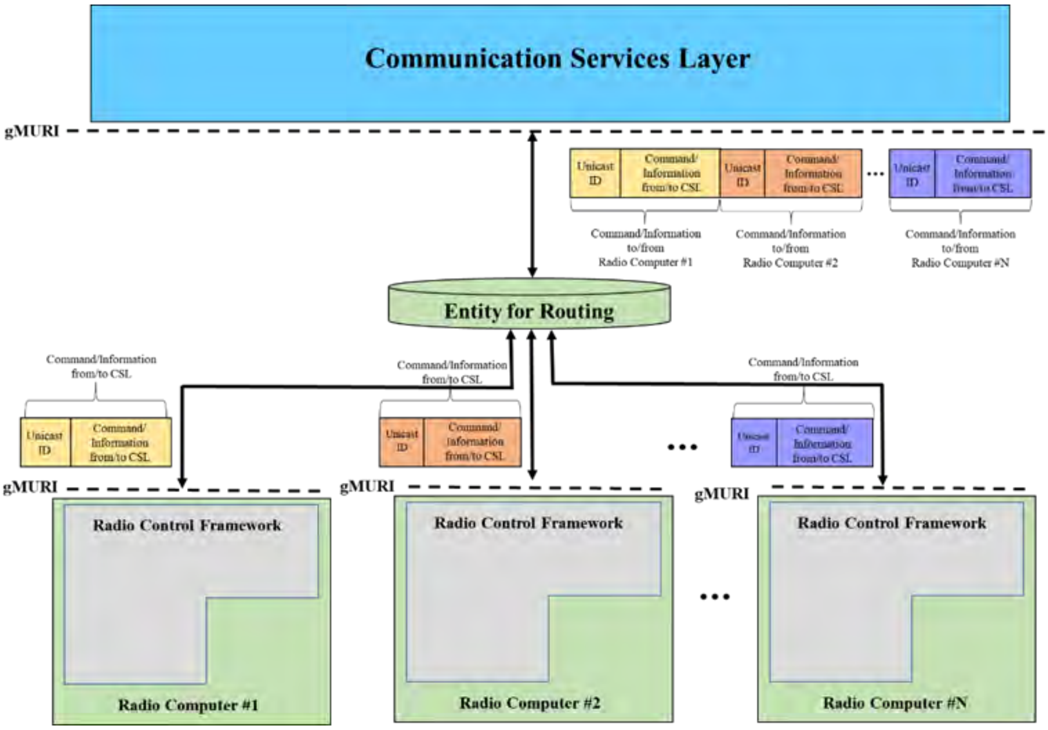

Communication Services Layer
gMURI
Unicast ID
Command/ Information from/to CSL
Unicast ID
Command/ Information from/to CSL
Unicast ID
Command/ Information from/to CSL
Command/Information to/from Radio Computer #1
Command/Information to/from Radio Computer #2
Command/Information to/from Radio Computer #N
Entity for Routing
Command/Information from/to CSL
Command/Information from/to CSL
Command/Information from/to CSL
Unicast ID
Command/ Information from/to CSL
Unicast ID
Command/ Information from/to CSL
Unicast ID
Command/ Information from/to CSL
gMURI
gMURI
gMURI
Radio Control Framework
Radio Control Framework
Radio Control Framework
Radio Computer #1
Radio Computer #2
Radio Computer #N

Figure 5-14: Architecture overview of locally co-located Radio Computers [303681-1].

In this section, a solution proposal consists in extending the upper architecture, as illustrated in Figure 5-14, in order to cover the case of a Compute Federation constituting of geographically distributed compute nodes. There are geographic locations #1…#L. There are a number K(1) of Radio Computers (per [303681-1]. A Radio Computer is part of Radio Equipment working under Radio Operating System control and on which Radio Applications are executed [303681-1]) at location #1, a number K(2) of Radio Computers at location #2, …, a number K(L) of Radio Computers at location #L.

In addition to the definitions of [303681-1], the proposal is to introduce an information field characterizing the transportation of computing task execution data as follows:

- Communication latency from Radio Computer to the Routing Entity (we assume that the delay depends on the actual geographic locations of the end-points; connection between a Radio Compute may be either wired, e.g., fibre optic-based, or wireless);
- Synchronization information: Information on synchronization is given, typically the synchronization is lost for remotely located compute resources and must be re-established when data is combined from different such sources. In specific cases, however, it may be possible that provisions were taken to maintain synchronization.
- Further possible information attributes/ fields, informing the other federated Radio Computers of the specific Radio Computer's service availability and autonomy are the following:

- Overall processing/ memory/ storage capabilities. Processing capabilities may e.g., refer to: i) the type of processing unit (e.g., CPU, GPU, FPGA, NPU), and ii) the number of cores and the capability of each core (clock size in GHz). Memory and storage size are measured in GB.
- Compute service availability of Radio Computer at time of self-probing. Such information may be either binary ("available"/ "non-available" for new processing requests) or expressed in terms of percentage of available CPU/ memory/ storage resources at time of probing. Example: "CPU 80% available at 09:00 UTC".
- Energy autonomy of Radio Computer at time of self-check. Such information may be expressed by the battery lifetime level in case the Radio Computer is not plugged to the power grid (e.g., "35% battery level") or inform that the Radio Computer is connected to the power grid (e.g., "50% battery level - charging". Additional sub-attributes may further inform whether the Radio Computer uses energy produced by renewable resources.

A possible additional information attribute may be an indicator of Radio Computer use experience, for example, based on service consumer reviews (e.g., similar to online application market ratings). The proposed enhanced software reconfiguration framework extending the current ETSI RRS architecture appears in Figure 5-15.

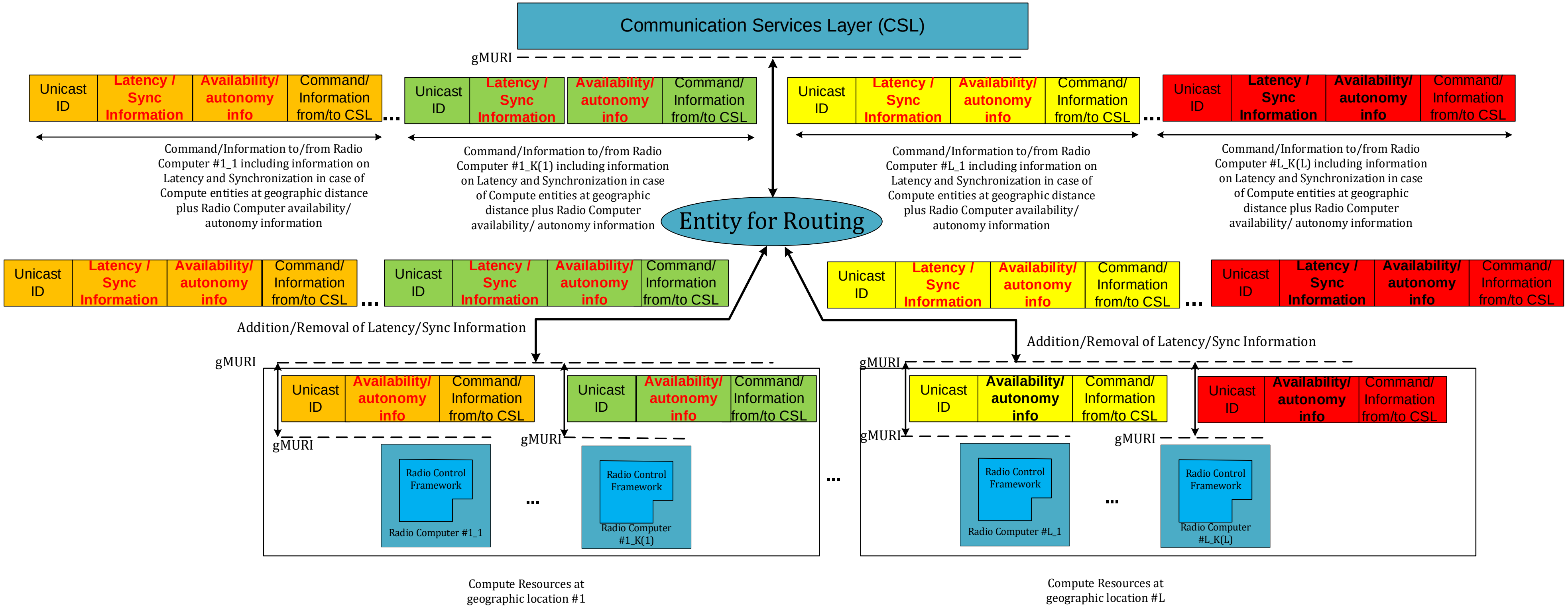

Communication Services Layer (CSL)
gMURI
Unicast ID
Latency / Sync Information
Availability/ autonomy info
Command/ Information from/to CSL
...
Command/Information to/from Radio Computer #1_1 including information on Latency and Synchronization in case of Compute entities at geographic distance plus Radio Computer availability/ autonomy information
Command/Information to/from Radio Computer #1_K(1) including information on Latency and Synchronization in case of Compute entities at geographic distance plus Radio Computer availability/ autonomy information
Command/Information to/from Radio Computer #L_1 including information on Latency and Synchronization in case of Compute entities at geographic distance plus Radio Computer availability/ autonomy information
Command/Information to/from Radio Computer #L_K(L) including information on Latency and Synchronization in case of Compute entities at geographic distance plus Radio Computer availability/ autonomy information
Entity for Routing
Addition/Removal of Latency/Sync Information
Addition/Removal of Latency/Sync Information
Radio Control Framework
Radio Computer #1_1
Radio Computer #1_K(1)
Radio Computer #L_1
Radio Computer #L_K(L)
Compute Resources at geographic location #1
Compute Resources at geographic location #L

Figure 5-15: Proposed architecture overview of Radio Computers of a (geographically distributed) Compute Federation - extending the reference architecture of [303681-1].

The proposed architecture is generic and can be applied to any number of geographic locations and any number of computational resources at any of those locations.

Typically, those computational tasks are distributed to Radio Computers at different geographic locations, depending on several factors, such as:

i. QoS requirements of the client/ device application issuing a request to address a computational task remotely. Examples are:

   a. output delivery delay;

   b. inferencing accuracy in case the task relates to an AI/ ML one.

ii. Billing/ charging scheme per mobile subscription of the device requesting computational task addressment by Radio Computers (in that case, a part of the Compute Federation will be chosen, where the Radio Computers are managed by the mobile operator the requestor has subscribed to, and possibly other operators with a business agreement in place. Selection will be made by a Central Controller in a way such that client application QoS requirements -e.g., task execution delay, output reception reliability- are addressed).

iii. Compute service availability of (candidate) Radio Computer(s) at the time of the request. Predicted compute service availability for the timeframe of task execution may also be calculated by the Central Controller, as frequently being updated by various Radio Computers in terms of task execution progress.

iv. Energy footprint of the selected part of Compute Federation, also considering the autonomy capabilities of the various available Radio Computers, that may affect fulfilment of QoS requirements (e.g., reliability of computation output delivery).

And within given geographic locations, ideally, the Central Controller may distribute computationally independent operations across the selected Radio Computers. For example, in case the task is an object identification one (e.g., identifying a person or object in images/ video footage), the Central Controller may split the image into segments and feed the selected Radio Computers with a segment each. However, it could be also cases where the same processing task is distributed across the selected Radio Computers for higher reliability (e.g., in case some of the selected Radio Computers are out of reach -due to mobility/ RLF events, or, in case of power shut-downs).

6 Architecture KPIs

To support the necessary features in the network needed to meet the requirements, a new generation of the architecture is needed; one based on the most forward-looking design principles together with trends in networks, use cases, and whatnot. This new architecture and its characteristics will also require new KPIs to allow for studying and analysing the performance of the designed system. Moreover, the new KPIs will also be necessary during the design phase itself both as guidelines and feedback for continuous optimisations.

The basis of our KPIs are our eight architectural design principles (see Figure 6-1) and the WP5 objectives (see section 1.1 and Table 1-1). The KPIs were initially defined in our previous deliverable [HEX-D51] and further developed in [EWS+22]. The architectural design principles in some cases corresponds directly to a KPI, for example the convergence time KPI in section 6.1.1 corresponds to the full automation (principle 2) and the extensibility and flexibility (principle 3).

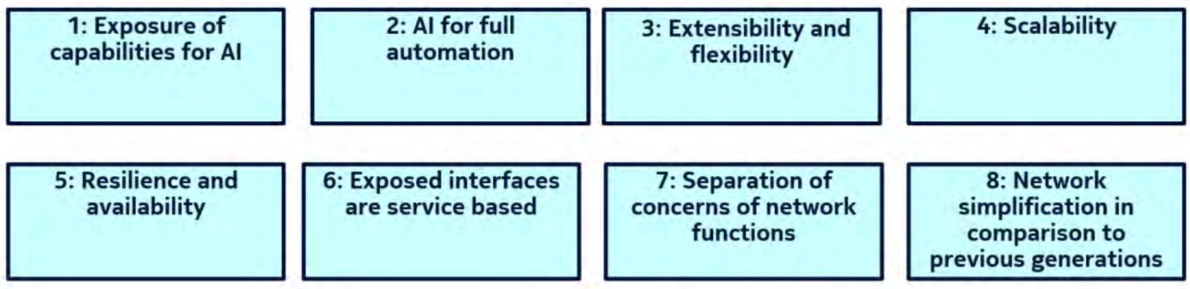

1: Exposure of capabilities for AI
2: AI for full automation
3: Extensibility and flexibility
4: Scalability
5: Resilience and availability
6: Exposed interfaces are service based
7: Separation of concerns of network functions
8: Network simplification in comparison to previous generations

Figure 6-1 Design principles for a 6G architecture

6.1 Architecture KPI definitions

In [HEX-D51] we proposed 7 different architecture KPIs which then were further developed in [EWS+22]. In this section, we also define the target (when possible) and elaborate on how our enablers can fulfil the KPIs.

6.1.1 Convergence time

The convergence time KPI estimates the time to adapt the network and its constituent elements, traffic routes and radio coverage to reflect the optimization decisions taken by the network management, orchestrator(s), and AI agents, see Figure 6-2.

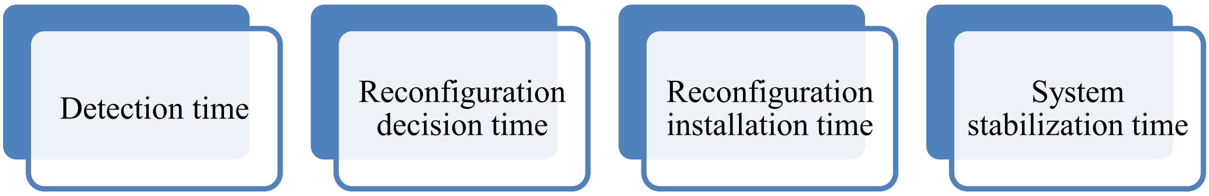

Detection time
Reconfiguration decision time
Reconfiguration installation time
System stabilization time

Figure 6-2: Convergence time components

These subcomponents are presented in the typical order where they appear when changing the network configuration:

1. **Detection time:** The time it takes to trigger the configuration change. The trigger could come from multiple sources: OAM, network analytics, policy control, performance monitoring of SLAs, as well as monitoring of cloud resources.
2. **Reconfiguration and decision time**: The time to decide what configuration shall be used. This also includes the reconfiguration decision and selection of impacted network nodes, services, and main functions. If new resources are added or old ones removed from the network, the related AI/ML models may need to be updated, and, in the worst case, retrained. In some cases, the configuration change, most likely caused by the addition of a new network node or a new type of service, may lead to a time-consuming retraining operation, if none of the pre-trained models can be used for this new configuration.
3. **Reconfiguration and installation time**: Time needed for propagation and installation of the new configuration and for the update of the AI/ML agents across the network.
4. **System stabilization time**: The time for the system to reach a new stable desired operation state. This includes path-switching times, and new QoS states of the service flows. Performance monitoring of the user flows shall indicate when the desired state has been reached. Closed-

looped control mechanisms managing affected subsystems require a synchronization phase with sufficient hysteresis to reach a stable state of E2E performance.

6.1.2 AI communication and computing overhead

This KPI refers to the amount of additional computing and communication resources allocated to optimize E2E QoS with in-network intelligence (AI/ML), compared to non-intelligent algorithms. Figure 6-3 shows an overview of identified overheads with regards to AI operations.

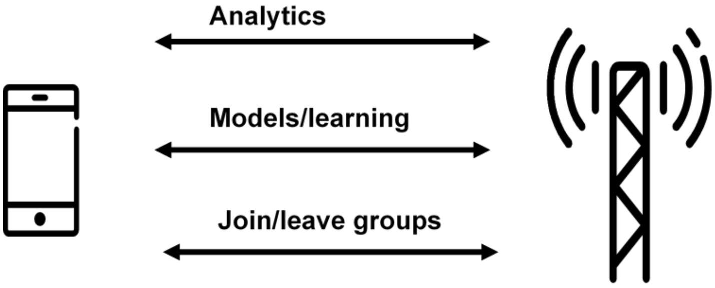


Figure 6-3: Identified overhead due to AI operations

The AI analytics overhead stems from the necessary data for learning, to and from the UE and the network. The overhead can also be that the network pushes AI models to UEs, or that UEs send learnings from an AI model back to the network AI functionality. This KPI can also include the signalling to join and leave certain AI groups as in the case with FL.

6.1.3 Network reliability for network of networks

This KPI measures the reliability and robustness of the network. These three aspects are related. This KPI assumes we have a network of subnetworks. The KPI then measures the RLF or session interruption time for each (sub)network and a minimum user's data rate. The subnetworks can overlap in area or be adjacent. The KPI is only fulfilled if it can be fulfilled over the whole areas, regardless of which the specific subnetwork that is operating in a subset of the whole area. An example can be an NTN covering rural and ocean areas which can overlap both with normal Terrestrial Network (TN) as well as mesh (indoor) networks.

6.1.4 Separation of concerns and ease of adding new functions in future

This KPI measures the number of dependencies between NFs with objective to minimize such dependencies. One advantage with few dependencies is more efficient signalling (fewer NFs need to be involved). To enable this, there must be clear division of responsibility (separation of concerns), especially in multi-vendor networks. This means that a NF will only handle one area of responsibility and no other NF will have same or similar functionality. In some cases, this means that the 5G NFs will be split into new NFs and in some cases, it may be so that some NFs in 5G may be merged. With this approach, the signalling may be more efficient, and the NFs can be developed and replaced independently from each other (principle 07).

This KPI is very much related to how many and what kind of transaction (earlier known as "procedures") a certain NF functional split result into. The behaviour and interfaces of a function need to be clearly defined, so exchangeability of functions is practically enabled. While in the "legacy/old" telco world, external interfaces were the focus of standardization, as functional entities were implemented by physical boxes and could not easily or dynamically be replaced. With the current/future virtualization paradigm and the SBA architecture refocus, it is critical to be able to define functions in a way that allow them to be re-used, while not adding a lot of dependencies.

Based on the above, this KPI can be defined depending on the number of involved nodes, NFs and interfaces for a specific procedure or number of specifications that need to be updated. Note that here we combine the KPIs “Separation of concerns of network functions “ and “Ease of adding new functions in future” in [HEX-D51] and [EWS+22], since the way the KPIs are defined are almost identical.

6.1.5 TCO reduction

In order to achieve the Hexa-X’s objective of an overall reduction of TCO of at least 30% with respect to the 5G NR SA, a methodology has been developed by the project based on the results available in the GSMA study [GSM19a] described in section 5.6. It should be noted that GSMA provided hints on 5G economics in an MNO-centric fashion, that is, it assumes that the ownership of the network is completely in the hands of the mobile operator. Such assumption could be revised when evaluating the 6G TCO due to the fact that the operator’s role within the whole 6G ecosystem actually depends on the use case being considered for the TCO computation, hence making the concept of “ownership” use case dependent to some extent. This is not new in 6G since already in 5G there are cases in which some of the network assets that were historically 100% owned by the operator are now assigned and managed by another player in the mobile ecosystem: for instance, in typical 5G industrial usage scenarios, some regulators offer spectrum to enterprises for private networks to support localized 5G applications. Proponents of this approach cited many benefits in terms of deployment costs, network management and customization, while MNOs have been somehow reluctant to this licensing approach due to impacts in spectrum utilization efficiency, lower quality of service and coverage as well as increased coordination complexity to avoid potential harmful interference [GSM21].

Among the 6G use cases identified by the Hexa-X project [HEX-D12][HEX-D13] some of them will be selected in order to determine a match with the deployment scenarios considered by GSMA, along with use case specific assumptions in terms of 6G data traffic CAGR; then the same cost items as in the GSMA study – i.e., RAN infrastructure, backhaul, CN infrastructure, energy and other network costs – will be evaluated for each identified “6G use case / deployment scenario” couple in order to determine the overall 6G network TCO Delta with respect to the *5G Optimised Case* being the baseline cost structure (see within the dotted box in Figure 6-4).

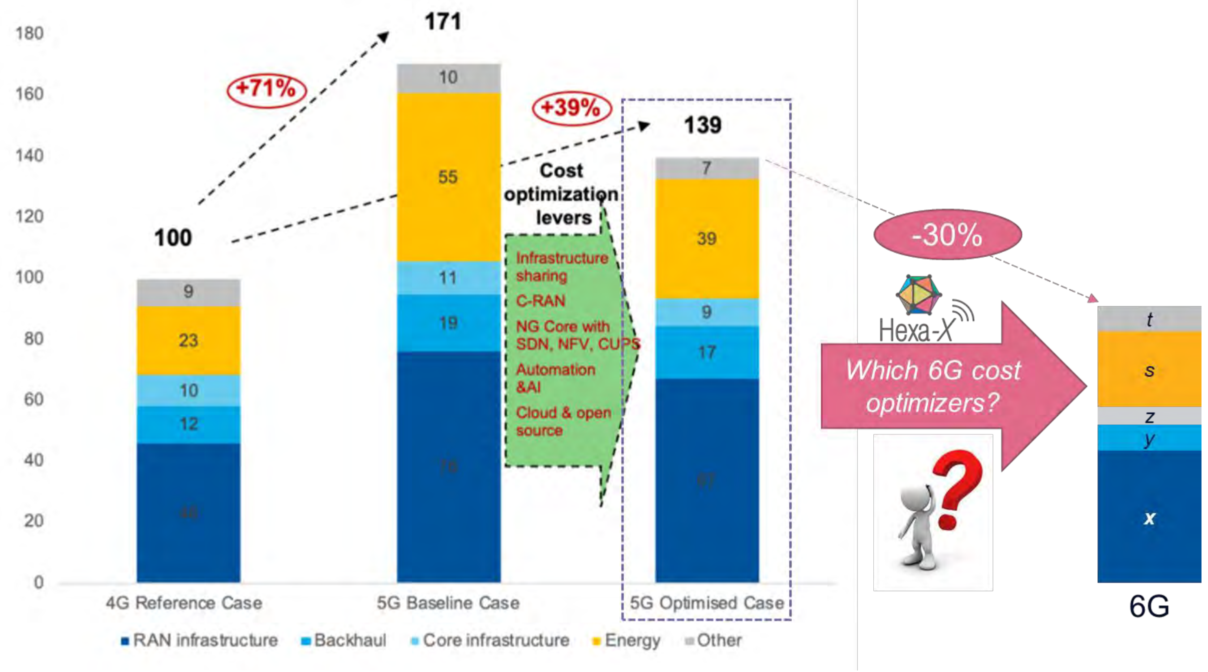


Figure 6-4: Example of the overall 6G network TCO (for a specific 6G use case) compared to the “rapid, full scale 5G deployment - 5G Optimised case”.

6.2 Enablers addressing the KPIs

Table 6-1 shows a summary of the architecture KPIs. The Table 6-1 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. The enablers are further explained below in section 6.2.1 to section 6.2.5.

Table 6-1 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, campus, 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
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.

6.2.1 Convergence time enablers

The main enabler for this is DFP, which helps to minimise reconfiguration and installation time and system stabilisation time once the related configuration changes triggering, and the selection of a new configuration are done. To achieve the envisioned delay value, the key is to have efficient proactive manners to shorten the overall durations of the needed functionalities by minimising the number of reactive tasks and their execution time. In practice, the cases where either the amount of proactive or reactive tasks is zero are not realistic, i.e., the extreme cases are more like theoretical ones. AI/ML based predictive (and, thus, proactive) function orchestration can reduce the convergence time by practically nulling out the detection time component in case predictions are both accurate and available to the network orchestrator before an event (e.g., user mobility) and with sufficient time advance for setting up the reconfiguration policy.

6.2.2 AI overhead enablers

The proposed AI (i.e., AIaaS and analytics in section 3.2) framework is envisioned to require less communication, processing, memory, and storage network-wide overheads, as compared to network domain-specific centralised AI agent deployments. The reason is that, as suggested, open interfaces across network domains can enable data and model transfer, thereby, saving model training time and, hence, releasing processing resources that would be otherwise reserved for model training. Of course, the communication overhead is expected to increase, and this framework may be more prone to security attacks, but algorithms and methods of only sharing significant data and model parameters can remedy these overheads.

The proposed FLaaS framework (see section 3.2.5) also contributes to keep the AI-related communication overhead low. In fact, FL enables the construction of global AI models by allowing UEs to share the parameters of locally trained AI models only, instead of whole sets of raw data. Moreover, the FLaaS framework is designed such that an FL process can select the most suitable communication paradigm (e.g., request-response/subscribe-notification), hence taking communication overhead into account.

6.2.3 Network reliability enablers

The main enablers for this KPI are the mesh network (see section 4.1) and the NTN coverage (see section 4.3). The mesh network includes algorithms for node discovery and selection of best routes (see section 4.1). Integrating NTN inherently in 6G enables an improved reliability and robustness. For example, a fast-moving UE can quickly switch to an NTN if the terrestrial network becomes insufficient or unavailable. DFP and network programmability are also important enablers that can add resources where they are needed on demand basis (assuming the hardware is available). Another enabler that can improve the reliability is a new enhanced MC solution, see section 4.2. This new MC solution can include features like decoupling of UL and DL as well as decoupling of UP and CP. This enables the network to send DL data via several cells to one UE, while in the UL just send data to one cell (to avoid splitting the UE transmit power). A challenge in 6G will also be that network of networks can combine public and non-public networks (e.g., campus networks) in a unique communication system, hosting and interconnecting heterogeneous technologies and services. This increases the necessity of this KPI, which can capture such a complexity of the combined flexibility, reliability, and robustness.

Several 5G use cases require high reliability, but so far, in 5G, this has not been provided; therefore, in 6G, this topic requires special attention. Please note that low latency services that require high reliability (Industry 4.0, etc.) pose a specific challenge on reliability mechanisms as the interruption of communication in such cases has to be very short. The reliability issue has to be considered from the E2E point of view; therefore, all network segments' reliability has to be considered. In the case of the mobile network, the overall service chain is typically composed of the radio part, edge platforms, the core part, and the external network that provides connectivity to service servers (or cloud). In the virtualization era, the reliability of infrastructure resources has to be considered.

Moreover, the reliability of data transport has to be taken into account. The data transport concerns all domains (radio, core, external network). In order to improve transport reliability, the multipath transmission that uses disjoint nodes and disjoint links is typically used. In the virtualized world, this disjointness has double meanings, as several transport nodes can be built using the same data centre, and several virtual links may share the same physical link. Such mechanisms are costly but necessary for high reliability as required by some 5G/6G services. The reliability of virtual nodes (functions) is also essential, and in the softwarized networks, the software-related issues (bugs, not dully tested behaviour) can impact reliability. Generally, a reliability model that considers the Mean Time Between Failures (MTBF) of different system components has to be considered, and an appropriate model has to be applied to each use case on that basis. For example, using edge clouds reduces the number of domains or components involved in the service chain. Still, on the other hand, the reliability of edge clouds is lower than the reliability of big, central clouds. Therefore, there is a trade-off of transport versus cloud reliability that can't be solved a priori.

6.2.4 Separation of concerns and ease of adding new functions in future

By enabling separation of concerns the number of dependencies between NFs can be reduced, thus reducing the need for some signalling (fewer NFs need to be involved). Important enablers for reshuffling of functionality are virtualization and Service based type architecture. These enablers also allow more reuse of functions. The optimal NF includes all functionality and data needed for the specific task so that no NF-NF signalling is needed. There is no single KPI capable of measuring how well this target is met. Instead on KPI will measure how well the separation of concern is applied by logging how much data that needs to be requested from/shared with other functions to be able to fulfil the task of the function. This can be observed in the number of signalling exchanges with other NFs. Another KPI looks at how many (e.g., 3GPP) specifications that need to be changed when the NF is updated, or the NF is added. A function that needs many updates in the specifications is more difficult to add, which would then be the opposite of what we wish to achieve.

Network and UE programmability enable reconfiguring the functionality to be executed on these two ends on demand. This allows easily adding, updating, and removing some functions from the UE and network to shorten the development cycles and enable agile integration of innovative solutions

6.2.5 TCO enablers

For the TCO evaluation of 6G the adoption of the most significant and use case specific 6G technical enablers will serve as means for quantifying the reduction of each cost item being considered, that is, RAN infrastructure, backhaul, CN infrastructure, energy, and other network costs. The actual enablers will be identified once there will be a clear match between a subset of 6G use cases and deployment scenarios considered by the GSMA study representing the 5G baseline.

7 Quantified targets for network evolution and expansion towards 6G

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 as well as some initial results. Full results will be included in D5.3.

7.1 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 D5.2 we do not consider the session set up time or any other actions preceding the session. Instead, our focus is on the DL of an E2E 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 full E2E path. The latency contributing factors on the E2E path depend on the RAN configuration in addition to the distance between the UE and the application server. The RAN functionality can be distributed into multiple units (also known as a split RAN) or a single RAN node can host all functions. To consider the RAN functional distribution we apply a 5G variant of functionality split, as for 6G there doesn't exist any yet. Starting from the UE and proceed towards the application server in the CN (see Figure 7-1), the E2E path consist of a UE including the protocol stack (PHY+MAC+RLC) and the radio layer processing and signal propagation over the air. Thereafter we have the Radio Unit (RU) that implements the RF-layer, then the DU that implements PHYsical layer (PHY), MAC and RLC layers and CU that implements the Service Data Adaption Protocol (SDAP) and PDCP-U processing, fronthaul and backhaul transport latencies (Eth), CN UP element processing (UPF), datacentre switch(es) and the application server IP-stack.

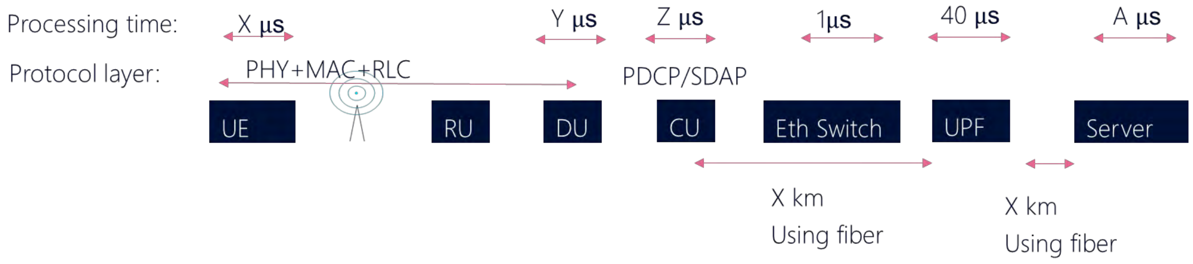


Figure 7-1: Involved components (nodes) and the latency assuming a distributed RAN

The mentioned RAN units can be combined into a classical single 6G RAN node, refactored, or co-located in one or more clouds, e.g., RU and DU are co-located, separate CU, in multiple ways.

This leads our analysis to consider different scenarios depending on how the RAN elements are located and where the application server is placed.

The radio latency estimate is common among all the scenarios. The estimate is based on the information gained from WP2 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 2 to 4 parallel streams should be employed, either as point-to-point MIMO or as D-MIMO. We find out that the choice of numerologies above 480 kHz has a limited impact on achieving the latency target. According to WP2 latency analysis, around 100 µs latency over the radio link (PHY and MAC layer) is achievable with multiple streams (2-4 parallel), thus, 100 µs will be used for radio latency in our analysis (see Figure 7-1). The real limiting factor is not numerology or bandwidth (assuming that they are large enough), but UE and BS processing times. For these we do not have good available estimates and we need to conclude what is left for them based on the latency consumed by the known latency contributors. The UE/server application stack above the radio and BS processing times are to be consider as the implementation dependent "unknowns".

7.1.1 Scenario 1 distributed RAN and edge server 50 km from RU.

In the first scenario we use a fully distributed RAN. The distance between the RU and the application server is selected to be 50 km, see Figure 7-1. For cable fabric latency we use values of hollow core optical fibres which is ~200 µs latency for 50 km. The cable fabric is segmented between the distributed RAN elements (RU-DU-CU) and UPF. For opto-electronics 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.

UPF one-way processing latency can be estimated from the state-of-the-art corresponding 5G elements to be 40 μs [Int20].

Summing the known latency contributions up we face 421 μs latency. The unknown latency contributors of the scenarios are processing time for UE stack, application stack (i.e., X and A in Figure 7-1 and Figure 7-2 that are unknows OS and application processing on top of the radio stack) and 6G DU, CU stack for which we have 579 μs left to stay within the given latency budget of 1 ms.

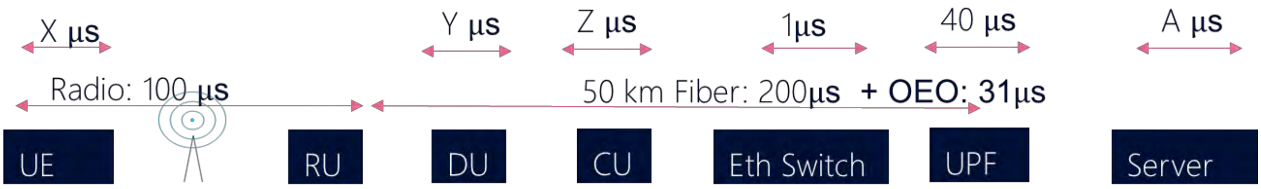


Figure 7-2: Latency components of a distributed RAN

7.1.2 Scenario 2 Co-located Cloud RAN and CN, DU and RU combined.

In this scenario RAN CU and CP functionalities are co-located in the same cloud with the CN, see Figure 7-3. RU is connected to DU over a 20 km long front haul. DU is connected to CU over a 30 km long backhaul. 6G UPF is collocated with the edge application server.

Cable fabric latency remains the same as in scenario 1, i.e., 250 µs. The number of opto-electrical devices is half of scenario 1, leading to 15 µs latency.

When unknown latency contributors for both scenarios are the same, by summing the known latency contributions up in this case we face 406 µs latency and 594 µs that remains to stay within the given latency budget of 1 ms.

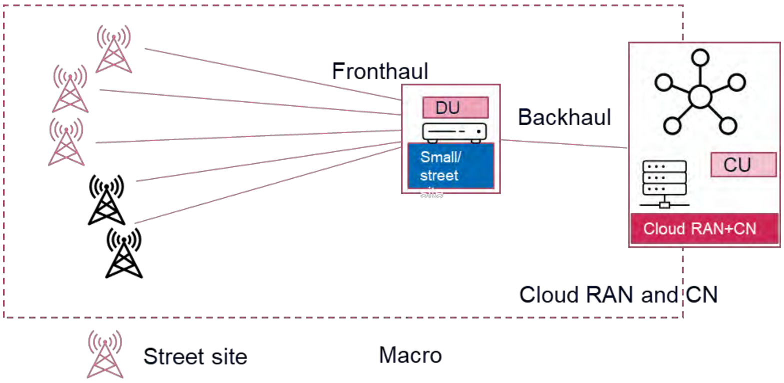


Figure 7-3: 6G Cloud RAN deployment. The radio unit (RU) is assumed to be connected to the distributed unit over a 20 km long fronthaul. The DU is then connected to the "cloud RAN and Core network", where the CU and the UPF and the server reside.

7.2 (>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. A refined evaluation will be given in D5.3.

7.2.1 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 is in [2412-0] 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 decided by using system simulations and link simulations.

The outputs of the simulations 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:

$$B_i = \frac{T}{R_i/W_{user}}, \qquad (7\text{-}1)$$

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:

$$C = \frac{N}{A} = \frac{N_{mux} \cdot W/mean(B_i)}{ISD^2 \cdot \sqrt{3}/6} \quad , \qquad (7\text{-}2)$$

Where N_{mux} is the number of users multiplex 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. The term $W/mean(B_i)$ is giving the number of users that a TRxP can accommodate.

Table 7-1 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 rather low interarrival time for the packets, the small size of the packet and the relatively low carrier frequency, but also from an efficient radio interface.

Table 7-1 Connection density from [37.910].

	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

7.2.2 Verifying connection density for NR

To estimate if the connection density can handle more than the target of 100 billion connections, we use a city with high population density, in this case Paris and Athens. We then compare this with the worst cases in Table 7-1. 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 7-2 shows that the maximum number of connections achieved in [37.910] exceeds the target connections with roughly 4-5 times.

Table 7-2 Estimate of 100 billion connections, per city areas

City	Population [millions]	Area [km^2]	Target connection density [millions]	Maximum connections for NR_FB_1732m [millions]	Maximum NB-IoT-RRCresume [millions]
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 density? Parameters that increase the connection density C in (7-2) are:

- Increased Signal to Interference plus Noise Ratio (SINR)
- Increased bandwidth W
- Increased multiplexing
- Decreased Inter-Site Distance (ISD).

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)

These enablers listed here are basically the same enablers as explained in section 6.2.4 Architecture KPIs.

7.3 Full coverage (100%) of world area

The objective with this target is to estimate the global coverage with a minimum data rate. Here we will describe the current assumptions and methodology. A full evaluation will begiven in D5.3. For this we assume that each area in the world is covered by 6G with a cell capacity of at least 1 Mbit/s. The enablers we will use to fulfil this objective are NTN and D2D mesh.

The assumptions are the following:

- Assume a certain cell area for the LEO satellite (and possibly GEO satellites too)
- Assume a maximum of X satellites
- Assume each satellite is equipped with beam forming and the gateway is a dish antenna with a certain antenna gain
- Assume the satellites can relay the data (inter-satellite links, see section 4.3.2)

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
- Estimate the feeder link capacity, i.e., the number of NTN cells served by a specific ground station and feeder link, see Figure 7-4
- Take the minimum of the feeder link capacity and the realistic TCP cell throughput

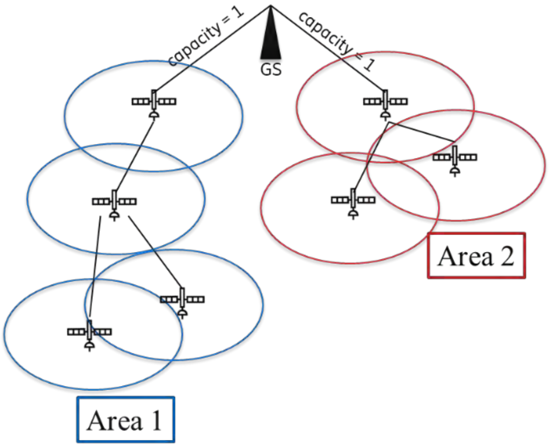


Figure 7-4 The feeder link and the ground station (GS) need to handle several satellite cells interconnected with each other

7.4 (>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. We will use a new name: (>99%) of global population reached with (>1 Mbit/s) data rates using estimated capacity over km^2 over 95% of the surface. The enablers for fulfilling this objective are NTN and D2D mesh. Here we will describe the current assumptions and methodology. A full evaluation will be given in next WP5 deliverable D5.3.

The target is divided into two parts: the first part is to estimate the capacity for a satellite system giving coverage for a (limited) number of users in rural areas and the second part is to provide the indoor coverage using mesh systems.

This target uses the same assumptions and methodology as in Section 7.3 but with the addition of M users per cell. The realistic TCP cell throughput is therefore divided among the M users. The M users are derived from a rural area in Europe. We exclude the densely populated areas since these are assumed to be covered by the terrestrial 6G network.

8 Conclusion

The main objectives of WP5 are to develop architectural components for 6G that support full AI integration and network programmability, a new flexible network design, while, at the same time streamline and redesign the architecture for a network of networks. These main objectives are addressed in this document. We envision that a 6G architecture can be built on top of a distributed multi-domain, multi-cloud environment, where functionalities span over heterogenous and specialised clouds.

For the **Intelligent networks,** a framework is developed for AI assisted network automation. Several initial solutions are presented on how to integrate AI/ML functionality to the Hexa-X 6G architecture. AI based automation is required in emerging 6G networks to manage the complexity in terms of technology and services and to meet quality, security, and resilience requirements. We have identified relevant AI functions needed to provide AI automation and seamless AI-driven 6G orchestration. The functions are AI repository, training, monitoring, and finally AI agent.

An analytic framework that exchanges knowledge and data across planes and domains to support the AI agents and ML model training are developed. Two possible solutions are presented: i) a fully distributed edge-based solution, where all AI-functions are instantiated at the edge nodes as cloud native applications and, ii) a hybrid solution where computational-intensive AI-functions are executed in the core cloud on the behalf of the AI agents located in the edge cloud. Further on, we propose a Federated Learning as-a-Service framework and related protocols to discover and join learning federations of UEs which allows UEs to train their own AI-models collaboratively in a privacy-preserving manner. To assess the feasibility of a network with AI functionality, we show that an AI based solution of wireless remote control that predicts the next command for robot movement in case the command is lost can decrease the robot trajectory error.

To handle the regulatory aspects of data governance, we propose a framework in which trust levels of multiple cross-domain AI-service consumers are managed to respond to data privacy needs within each domain. The proposed framework may be able to reduce the AI overhead, since operation of a given AI agent in one domain can be extended to different privacy domains, provided that the privacy requirements are addressed within each domain.

To handle varying network functional and performance requirements, programmable nodes and devices can be used. We evaluate the performance and cost of a programmable network in multiple configurations. We show that network programmability leads to a more flexible network without compromising performance, as well as helps to reduce the Total Cost of Ownership of the system. Programmability can also be utilized to adapt the UE protocol stack, in order to flexibly support new use cases for, e.g., dedicated networks.

The underlying infrastructure layer of the Hexa-X E2E architecture [HEX-D13] needs to be adaptable to varying NF workloads, new functionality, and dynamic placement of NFs across the multi-cloud continuum. We propose a two-level hierarchical orchestration solution, where domain-internal dynamicity is not fully exposed externally but internally so that network functions can be executed in a multi-domain multi-cloud environment on-demand basis. A top-level orchestrator decides candidate domains for NFs, to be created or to be moved. Final deployment details are left for domain specific logics.

Flexible networks 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.

The D2D mesh network is an ad hoc network consisting of temporary local nodes (e.g., D2D devices, other temporary access points nodes). The main benefit with a D2D mesh network is that it can extend the coverage in a flexible and scalable manner. To manage the ad hoc network, a Management and Orchestration architecture concept is proposed.

Another important topic for a flexible network is the ability of the network to utilize the available spectrum and achieve a reliable connection. For this we propose a new multi-connectivity solution for 6G, which combines features from CA and DC. For full global coverage, we investigate an NTN deployment with a functional split of the RAN, where the DU is located in a UAV and the CU in a LEO based satellite system. We also investigate the need for inter-satellite-link hops to achieve full global coverage, using several different hop schemes. We find that it is possible to achieve full global coverage with a simplified scheme with just a limited latency increase.

The 6G architecture should enable **Efficient networks**. With this we mean that 6G should be more efficient in terms of, e.g., performance, (signalling) overhead, scalability as well as resource and power consumption compared to previous generations.

Independent NFs enable a new RAN and CN architecture, using an SBA approach. In this deliverable, we describe a method for how to characterize and design independent and self-sustained network functions. Further on, 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. To simplify cloud native CN RAN implementations as discussed in [HEX-D51], we here propose to extend a Service Based Interface with a direct interface from the RAN to the CN NFs, instead of relaying the CP messages via the AMF. This can reduce the latency and avoid single point of failure (for the AMF). Further on, to handle the new 6G requirements such as low latency and reliability, it is also important to optimize and simplify the RAN UP, taking into consideration new advances in Layer 4 transport protocols such as TCP and QUIC. The simulations show that PDCP in-order delivery and RLC retransmissions have a small impact on the endpoint applications showing that TCP can handle packet loss and out of order delivery. In addition to this, we also investigate a Compute-as-a-service solution, based on the ETSI RRS architecture and the so-called Compute Federation constituting of geographically distributed compute nodes. The solution is to introduce an information field characterizing the transportation of computing task execution data.

The efficiency of the network can also be translated to energy efficiency and Total Cost of Ownership. In this deliverable we have developed an initial methodology on how to evaluate the Total Cost of Ownership and the final results will be provided in deliverable D5.3.

In addition to the above, we also develop a set of **architectural KPIs** that we believe are needed to ensure a successful 6G architecture. These KPIs are derived from the architectural principles we defined in [HEX-D51] and from our WP5 objectives (see section 1.1 and Table 1-1). As can be seen from the summary of the KPIs in Table 6-1, several of our enablers described in the document can help to fulfil the KPIs.

The document also introduces the so called "**quantified targets**" for the "Network evolution and expansion towards 6G" objective. For two out of the four quantified targets there are some initial evaluations that show that we can reach the targets while for the remaining two quantified targets we have developed a methodology on how to evaluate the quantified target.

9 References

[23.041] 3GPP TS 23.041, "Technical realization of Cell Broadcast Service (CBS) (Release 18)", v18.0.0, Jun. 2022.

[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.

[23.501] 3GPP TS 23.501, "System architecture for the 5G System (5GS); Stage 2 (Release 17)", v17.2.0, Sep. 2021.

[23.502] 3GPP TS 23.502, "Procedures for the 5G System (5GS)", v17.2.1, Sep. 2021.

[23.799] 3GPP TR 23.799, "Study on Architecture for Next Generation System", v14.0.0, Dec 2016.

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

[28.809] 3GPP TR 28.809, "Study on enhancement of Management Data Analytics (MDA) (Release 17)", v17.0.0, Apr. 2021.

[29.500] 3GPP TS 29.500, "5G System; Technical Realization of Service Based Architecture; Stage 3", v17.6.0, Mar. 2022.

[29.893] 3GPP TR 29.893, "Study on IETF QUIC Transport for 5GC Service Based Interfaces", v1.7.0, Feb. 2022.

[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.

[37.320] 3GPP TS 37.320 "Radio measurement collection for Minimization of Drive Tests (MDT); Overall description; Stage 2", v16.8.0, April 2022.

[37.340] 3GPP TS 37.340 "Evolved Universal Terrestrial Radio Access (E-UTRA) and NR; Multi-connectivity", v16.2.0, July 2020.

[37.817] 3GPP TR 37.817, "Study on enhancement for data collection for NR and ENDC", v17.0.0, Apr. 2022.

[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.

[38.331] 3GPP TS 38.331 "NR; Radio Resource Control (RRC); Protocol specification", Rel-16, Version 16.6.0, Sep. 2021.

[38.413] 3GPP TS 38.413, "NG-RAN; NG Application Protocol (NFAP)", v17.1.1, Jun. 2022.

[802.11s] IEEE 802.11s, "Overview of the Amendment for Wireless Local Area Mesh Networking," Online: https://www.ieee802.org/802_tutorials/06-November/802.11s_Tutorial_r5.pdf.

[6GFlag20] 6G Flagship, "White Paper on 6G Networking, 6G Research Visions, No 6.", Jun. 2020.

[ABG+21] S. T. Arzo, R. Bassoli, F. Granelli, and F. H. P. Fitzek, "Multi-Agent Based Autonomic Network Management Architecture," in IEEE Transactions on Network and Service Management, doi: 10.1109/TNSM.2021.3059752.

[AIA21] K. M. Ahmed, A. Imteaj, and M. H. Amini, "Federated Deep Learning for Heterogeneous Edge Computing," 2021 20th IEEE International Conference on

Machine Learning and Applications (ICMLA), pp. 1146-1152, 2021, doi: 10.1109/ICMLA52953.2021.00187.

[BAH+18] K. Bogineni, A. Akhavain, T. Herbert, D. Farinacci, A. Rodriquez-Natal, G. Carofiglio, J. Auge, L. Muscariello, P. Camarillo, and S. Homma, "Optimized Mobile User Plane Solutions for 5G; draft-bogineni-dmm-optimized-mobile-user-plane-01.txt', IETF Internet-Draft, June 2018, [Online]. Available: https://tools.ietf.org/html/draft-bogineni-dmm-optimized-mobile-user-plane-01.

[BBG+20] S. Bonafini, R. Bassoli, F. Granelli, F. H. P. Fitzek, and C. Sacchi, "Virtual Baseband Unit Splitting Exploiting Small Satellite Platforms," 2020 IEEE Aerospace Conference, 2020, pp. 1-14, doi: 10.1109/AERO47225.2020.9172316.

[BDG+14] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and. Walker, "P4: Programming protocol-independent packet processors," ACM SIGCOMM Computer Communication Review, vol. 44, no. 3, pp. 87–95, 2014.

[BGG+20] D. Bega, M. Gramaglia, A. Garcia-Saavedra, M. Fiore, A. Banchs, and X. Costa-Perez, "Network Slicing Meets Artificial Intelligence: An AI-Based Framework for Slice Management," in IEEE Communications Magazine, vol. 58, no. 6, pp. 32-38, June 2020, doi: 10.1109/MCOM.001.1900653.

[BGS+20] R. Bassoli, F. Granelli, C. Sacchi, S. Bonafini, and F. H. P. Fitzek, "CubeSat-Based 5G Cloud Radio Access Networks: A Novel Paradigm for On-Demand Anytime/Anywhere Connectivity," in IEEE Vehicular Technology Magazine, vol. 15, no. 2, pp. 39-47, June 2020, doi: 10.1109/MVT.2020.2979056.

[BJ15] M. Boucadair and C. Jacquenet, "Introducing Automation in Service Delivery Procedures: An Overview.," in Handbook of Research on redesigning the future of internet architectures, Hershey, PA, USA: Information Science Reference, an imprint of IGI Global, 2015.

[BKA21] A. Bayazeed, K. Khorzom, and M. Aljnidi, "A survey of self-coordination in self-organising network," Computer Networks, Vol. 196, pp 108222, 2021, ISSN 1389-1286.

[BMZ+20] J. Baranda; J. Mangues-Bafalluy, E. Zeydan, L. Vettori, R. Martinez, X. Li, A. Garcia-Saavedra, C. F. Chiasserini, C. Casetti, K. Tomakh, O. Kolodiazhnyi, and C. J. Bernardos, "On the Integration of AI/ML-based scaling operations in the 5Growth platform," IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 105-109, 2020, doi: 10.1109/NFV-SDN50289.2020.9289863.

[CAMARA22] CAMARA: Telco Global API Alliance - Future Networks, 2022. https://www.gsma.com/futurenetworks/ip_services/understanding-5g/camara-telco-global-api-alliance/.

[CAMARA22a] CAMARA - The Telco Global API Alliance. 2022, https://camaraproject.github.io/index.html.

[CCD+18] Y. Cheng, N. Cardwell, N. Dukkipati, and P. Jha, "The RACK-TLP Loss Detection Algorithm for TCP", RFC 8985, DOI 10.17487/RFC8985, February 2021.

[CMC99] M. Corson, J. Macker, and G. Cirincione, "Internet-based mobile ad hoc networking," IEEE Internet Computing, vol. 3 no. 4) pp. 63-70, 1999. doi: 10.1109/4236.780962.

[CPRI15] CPRI, "Common public radio interface: Interface specification", V7.0, Tech. Rep., 2015 Available online: http://www.cpri.info/downloads/CPRI_v_7_0_2015-10-09.pdf.

[DLH19] Y. Dang, Q. Lin, and P. Huang, "AIOps: Real-World Challenges and Research Innovations," 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), pp. 4-5, 2019, doi: 10.1109/ICSE-Companion.2019.00023.

[DS21] A. Dusia and A. S. Sethi, "Software-Defined Architecture for Infrastructure-less Mobile Ad Hoc Networks," 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 742-747, 2021.

[EWS+22] M. Ericson, S. Wänstedt, M. Saimler, H. Flinck, G. Kunzmann, P. Vlacheas, D. Rapone, A. de la Olivia, C. J. Bernardos, R. Bassoli, F. H.P. Fitzek, G. Nardini, M. Filippou, M. Mueck, "Setting 6G Architecture in Motion -the Hexa-X approach", 2022 EUCNC, Grenoble, Available at: https://zenodo.org/record/6638245#.YvNeBfhBwuV.

[EUAI21] Council of the European Union, "Proposal for a regulation of the European parliament and of the council laying down harmonized rules of artificial intelligence (Artificial intelligence act) and amending certain union legislative acts", Brussels, 21.4.2021.

[EUAI21a] Council of the European Union, "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 - Presidency compromise text Interinstitutional File 2021/0106(COD)", Brussels, 29.11.2021, available at https://data.consilium.europa.eu/doc/document/ST-14278-2021-INIT/en/pdf.

[FBB+80] W. Findeisen, F. N. Bailey, M. Brdys, K. Malinowski, P. Tatjewski, and A. Wozniak, "Control and coordination in hierarchical systems," J. Wiley, New York, 1980.

[FCI01] Federal Chief Information Officer Council - Enterprise Interoperability and Emerging Information Technology Committee (EIEITC) - Federal Architecture Working Group (FAWG). "A Practical Guide to Federal Enterprise Architecture," February 2001, [Online]. Available: https://www.gao.gov/assets/588407.pdf.

[FKS20] D. Falanga, K. Kleber, D. Scaramuzza, Dynamic obstacle avoidance for quadrotors with event cameras, Science Robotics, Vol 5, Issue 40, 25 March 2020, [Online]. Available: https://www.science.org/doi/10.1126/scirobotics.aaz9712.

[FMM+96] S. Floyd, J. Mahdavi, M. Mathis, and D. A. Romanow, "TCP Selective Acknowledgment Options". 1996, RFC2018.

[FMR+20] N. Foster, N. McKeown, J. Rexford, G. Parulkar, L. Peterson, and O. Sunay, "Using deep programmability to put network owners in control" https://www.cs.princeton.edu/~jrex/papers/Pronto20.pdf.

[Gha20] Z. Ghadialy, "Understanding the TCO of a Mobile Network", 26 October 2020, The 3G4G Blog, [Online]. Available: https://blog.3g4g.co.uk/2020/10/understanding-tco-of-mobile-network.html.

[GKT21] N. Gritli, F. Khendek, and M. Toeroe, "Decomposition and Propagation of Intents for Network Slice Design," 2021 IEEE 4th 5G World Forum (5GWF), 2021, pp. 165-170, doi: 10.1109/5GWF52925.2021.00036.

[GSA21] Global mobile Supplier Association (GSA). “5G Market Snapshot 2021 – end of Year,” December 2021, [Online]. Available: https://gsacom.com/paper/5g-market-snapshot-2021-end-of-year/.

[GSM18] Global System for Mobile Communications Association (GSMA) “Road to 5G: Introduction and Migration,” April 2018, [Online]. Available: https://www.gsma.com/futurenetworks/wp-content/uploads/2018/04/Road-to-5G-Introduction-and-Migration_FINAL.pdf.

[GSM19] Global System for Mobile Communications Association (GSMA), “5G Implementation Guidelines: NSA Option 3,” March 2019, [Online]. Available: https://www.gsma.com/futurenetworks/wiki/5g-implementation-guidelines/.

[GSM19a] 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/.

[GSM21] Global System for Mobile Communications Association (GSMA), “Mobile Networks for Industry Verticals: Spectrum Best Practice,” July 2019, [Online]. Available: https://www.gsma.com/spectrum/wp-content/uploads/2021/07/Mobile-Networks-Industry-Verticals.pdf.

[HCJ15] N. Han, Y. Chung, and M. Jo, “Green data centers for cloud-assisted mobile ad hoc networks in 5G”. IEEE Network, vol. 29, no. 2, pp.70-76, 2015.

[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]. Available: https://hexa-x.eu/wp-content/uploads/2022/04/Hexa-X_D1.2_Edited.pdf.

[HEX-D13] Hexa-X Deliverable D1.3, “Targets and requirements for 6G – initial E2E architecture”, Mar. 2022, [Online]. Available: https://hexa-x.eu/wp-content/uploads/2022/03/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-D42] Hexa-X Deliverable D4.2, “AI-driven communication & computation co-design: initial solutions”, Jun. 2022, Online: https://hexa-x.eu/wp-content/uploads/2022/07/Hexa-X_D4.2_v1.0.pdf.

[HEX-D51] Hexa-X Deliverable D5.1, “Initial 6G architectural components and enablers”, Dec. 2021, Online: https://hexa-x.eu/wp-content/uploads/2022/03/Hexa-X_D5.1_full_version_v1.1.pdf.

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

[HEX-D71] Hexa-X Deliverable D7.1, “Gap analysis and technical work plan for special-purpose functionality”, Jun. 2021, [Online]. Available: https://hexa-x.eu/wp-content/uploads/2021/06/Hexa-X_D7.1.pdf.

[HH19] Huawei, HiSilicon, R1-1911858, 3GPP TSG RAN WG1 Meeting #99, “Discussion on performance evaluation for NTN”, Reno, USA, November 18 – 22, 2019.

[HHH+21] H. Harkous, B. A. Hosn, M. He, M. Jarschel, R. Pries, and W. Kellerer, “Towards performance-aware management of p4-based cloud environments,” in 2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV- SDN), 2021, pp. 87–90.

[HJH+21] H. Harkous, M. Jarschel, M. He, R. Pries, and W. Kellerer, “P8: P4 with predictable packet processing performance,” IEEE Transactions on Network and Service Management, vol. 18, no. 3, pp. 2846–2859, 2021.

[HRX08] S. Ha, I. Rhee, and L. Xu, “CUBIC: a new TCP-friendly high-speed TCP variant,” SIGOPS Oper. Syst. Rev. vol. 42, no. 5, pp. 64–74, July 2008.

[IKC20] C. -L. I, S. Kuklinskí, and T. Chen, “A Perspective of O-RAN Integration with MEC, SON, and Network Slicing in the 5G Era”, in IEEE Network, vol. 34, no. 6, pp. 3-4, November/December 2020, doi: 10.1109/MNET.2020.9277891.

[Inf20] Infinera, “Low Latency – How Low Can You Go?” White paper, [Online]. Available: https://www.infinera.com/wp-content/uploads/Low-Latency-How-Low-Can-You-Go-0188-WP-RevB-0920.pdf.

[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.

[ISB+14] O. Iacoboaiea, B. Sayrac, S. Ben Jemaa, and P. Bianchi, “SON Coordination for parameter conflict resolution: A reinforcement learning framework,” 2014 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), 2014, pp. 196-201, doi: 10.1109/WCNCW.2014.6934885.

[JIT+16] M. Jaber, M. A. Imran, R. Tafazolli, and A. Tukmanov, “5G backhaul challenges and emerging research directions: A survey,” IEEE Access, vol. 4, pp. 1743–1766, 2016.

[JT87] J. Jubin and J. Tornow, “The DARPA packet radio network protocols,” Proceedings Of The IEEE, vol. 75, no. 1, pp. 21-32. doi: 10.1109/proc.1987.13702, 1987.

[JTS17] A. Jain, V. Tokekar, and S. Shrivastava, “Security Enhancement in MANETs Using Fuzzy-Based Trust Computation Against Black Hole Attacks,” Information and Communication Technology, pp. 39-47, 2017.

[KDN+21] S. Kannan, G. Dhiman, Y. Natarajan, A. Sharma, S. Mohanty, M. Soni, U. Easwaran, H. Ghorbani, A. Asheralieva, and M. Gheisari, “Ubiquitous Vehicular Ad-Hoc Network Computing Using Deep Neural Network with IoT-Based Bat Agents for Traffic Management”. Electronics, vol. 10, no. 7) pp.785, 2021.

[KKT+21] S. Kukliński, R. Kołakowski, L Tomaszewski, L. Sanabria-Russo, C. Verikoukis, C-. T. Phan, L. Zanzi, F. Devoti, A Ksentini, C. Tselios, G. Tsolis, and H. Chergui, "MonB5G: AI/ML-Capable Distributed Orchestration and Management Framework for Network Slices," 2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom), 2021, pp. 29-34, doi: 10.1109/MeditCom49071.2021.9647681.

[KR12] J. F. Kurose and K. W. Ross., “Computer Networking: A Top-Down Approach” (6th Edition), Pearson, 2012.

[KRV+22] B. R. Krishna, M. H. Reddy, P. S. Vaishnavi, and S. V. Reddy, "Traffic Flow Forecast using Time Series Analysis based on Machine Learning," 2022 6th International

Conference on Computing Methodologies and Communication (ICCMC), 2022, pp. 943-947, doi: 10.1109/ICCMC53470.2022.9753812.

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

[KY20] M. Khan and K. Yau, “Route Selection in 5G-based Flying Ad-hoc Networks using Reinforcement Learning,” 2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), 2020.

[LDY+22] G. Liu, H. Dong, Z. Yan, X. Zhou, and S. Shimizu, “B4SDC: A Blockchain System for Security Data Collection in MANETs”. IEEE Transactions on Big Data, vol. 8, no. 3, pp.739-752, 2022.

[LLG+19] W. Li, Y. Lemieux, J. Gao, Z. Zhao, and Y. Han, "Service Mesh: Challenges, State of the Art, and Future Research Opportunities," 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE), 2019, pp. 122-1225, doi: 10.1109/SOSE.2019.00026.

[LSL+22] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge Artificial Intelligence for 6G: Vision, Enabling Technologies, and Applications," in IEEE Journal on Selected Areas in Communications, vol. 40, no. 1, pp. 5-36, Jan. 2022, doi: 10.1109/JSAC.2021.3126076.

[MBC22] M. Moussaoui, E. Bertin and N. Crespi, "5G shortcomings and Beyond-5G/6G requirements," 2022 1st International Conference on 6G Networking (6GNet), 2022, pp. 1-8, doi: 10.1109/6GNet54646.2022.9830439.

[MGG+18] J. Moysen, M. Garcia-Lozano, L. Giupponi, and S. Ruiz, “Conflict Resolution in Mobile Networks: A Self-Coordination Framework Based on Non-Dominated Solutions and Machine Learning for Data Analytics [Application Notes],” in IEEE Computational Intelligence Magazine, vol. 13, no. 2, pp. 52-64, May 2018, doi: 10.1109/MCI.2018.2807038.

[NEP] https://nephio.org

[NFV21] ETSI GS NFV 006, “Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Architectural Framework Specification”, V2.1.1, Jan. 2021.

[Nok22] Nokia, “AirScale Cloud RAN,” White paper, [Online]. Available: https://www.nokia.com/networks/mobile-networks/airscale-radio-access/cloud-ran/.

[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.

[NSV+20] G. Nardini, G. Stea, A. Virdis, D. Sabella, and P. Thakkar, "Using Simu5G as a Realtime Network Emulator to Test MEC Apps in an End-To-End 5G Testbed", PiMRC 2020, London, UK, 1-3 September 2020.

[ONA] https://www.onap.org

[ORA21] O-RAN, “O-RAN AI/ML workflow description and requirements v1.02”, Jul. 2021.

[ORA21a] O-RAN, “O-RAN Non-RT RIC Functional Architecture Technical Report v1.01”, Jul. 2021.

[ORA] https://www.o-ran.org/

[Pao17] M. Paolini, “How much can operators save with Cloud RAN?” Senza Fili and Mavenir, 2017, [Online]. Available: https://www.mavenir.com/app/uploads/2020/01/SenzaFili-Mavenir-TCO-WP.pdf.

[Pao21] M. Paolini, “Move as Fast as You Can to the 5GC”, Senza Fili and Mavenir, 2021, [Online]. Available: https://senzafili.com/publications/move-to-the-5gc/.

[Qua21] RP-213599, 3GPP TSG RAN Meeting #94e, “Study on Artificial Intelligence (AI)/Machine Learning (ML) for NR Air Interface”, Dec. 6 – 17, 2021.

[QYC+19] M. Qin, Q, Yang, N. Cheng, J. Li, W. Wu, R. R. Rao, and X. Shen, “Learning-Aided Multiple Time-Scale SON Function Coordination in Ultra-Dense Small-Cell Networks,” in IEEE Transactions on Wireless Communications, vol. 18, no. 4, pp. 2080-2092, April 2019, doi: 10.1109/TWC.2019.2898002.

[RFC2960] IETF, “Stream Control Transmission Protocol,” Request for comment 2960, October 2000, [Online]. Available: https://datatracker.ietf.org/doc/html/rfc2960.

[RGF+21] G.A. Reina, A. Gruzdev, P. Foley, Patrick, O. Perepelkina, M. Sharma, I. Davidyuk, I. Trushkin, M. Radionov, A. Mokrov, D. Agapov, J. Martin, B. Edwards, M.J. Sheller, S. Pati, P. Moorthy, W. Narayana S. Shih-han, P. Shah, and S. Bakas, “OpenFL: An open-source framework for Federated Learning”, arXiv, 2021, [Online]. Available: https://arxiv.org/abs/2105.06413.

[Ste97] W. R. Stevens. “TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms,”. 1997, RFC2001.

[SP800-207] NIST Special Publication 800-207, "Zero Trust Architecture," [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf.

[SSC+17] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, and A. Banchs, "Mobile traffic forecasting for maximizing 5G network slicing resource utilization," IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, pp. 1-9, 2017, doi: 10.1109/INFOCOM.2017.8057230.

[ST20] K. Samdanis and T. Taleb, “The Road beyond 5G: A Vision and Insight of the Key Technologies,” in IEEE Network Magazine, vol. 34, no. 2, pp. 135-141, Mar. 2020.

[TMF21] TM Forum, “GB998 Open Digital Architecture (ODA) Concepts & Principles”, v2.1.0, TM Forum, March 2021.

[VBT+22] L. Velasco, S. Barzegar, F. Tabatabaeimehr, and M. Ruiz, "Intent-based networking and its application to optical networks [Invited Tutorial]," in Journal of Optical Communications and Networking, vol. 14, no. 1, pp. A11-A22, January 2022, doi: 10.1364/JOCN.438255.

[XHH+21] L. Xinyun, L. Huidan, Y. Hang, C. Zilan, C. Bangdi, and Y. Yi, "IoT Data Acquisition Node For Deep Learning Time Series Prediction," 2021 2nd International Conference on Big Data Analytics and Practices (IBDAP), 2021, pp. 107-111, doi: 10.1109/IBDAP52511.2021.9552096.

[YLC+19] Q. Yang, Y. Liu, T, Chen, and Y. Tong, “Federated Machine Learning: Concept and Applications,” ACM Trans. Intell. Syst. Technol. vol. 10, no, 2, Article 12, pp. 1-19, March 2019, [Online]. Available: https://doi.org/10.1145/3298981.

[ZZC20] J. Zhou, W. Zhao and S. Chen, "Dynamic Network Slice Scaling Assisted by Prediction in 5G Network," in IEEE Access, vol. 8, pp. 133700-133712, 2020, doi: 10.1109/ACCESS.2020.3010623.

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

A.2 AI regulation articles related to technical requirements & proposed AI system overview

The EU AI Regulation [EUAI] is available as a draft. It contains a number of Articles, among which technical requirements are introduced as summarized in the table below. An illustration follows which is introducing key entities meeting those technical requirements in accordance with the discussion in 3.2.2.

Table A-2 AI Regulation Articles related to Technical Requirements [EUAI]

Requirements	Summary as defined by AI Regulation
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…

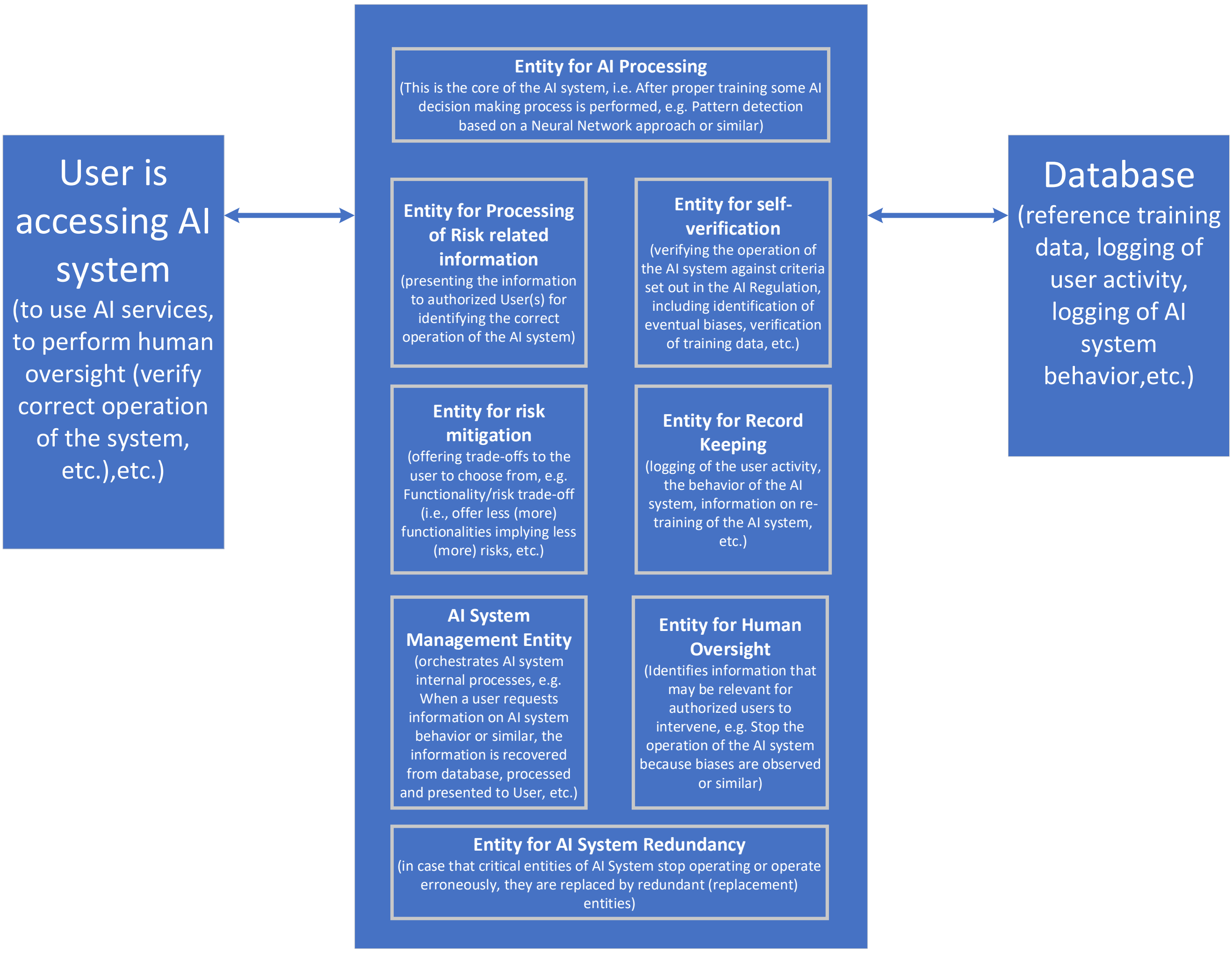

User is accessing AI system
(to use AI services, to perform human oversight (verify correct operation of the system, etc.),etc.)
Entity for AI Processing
(This is the core of the AI system, i.e. After proper training some AI decision making process is performed, e.g. Pattern detection based on a Neural Network approach or similar)
Entity for Processing of Risk related information
(presenting the information to authorized User(s) for identifying the correct operation of the AI system)
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.)
Entity for risk mitigation
(offering 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.)
Entity for Record Keeping
(logging of the user activity, the behavior of the AI system, information on re-training of the AI system, etc.)
AI System Management Entity
(orchestrates AI system internal processes, e.g. When a user requests information on AI system behavior or similar, the information is recovered from database, processed and presented to User, etc.)
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)
Entity for AI System Redundancy
(in case that critical entities of AI System stop operating or operate erroneously, they are replaced by redundant (replacement) entities)
Database
(reference training data, logging of user activity, logging of AI system behavior,etc.)

Figure 9-1: Components of a proposed AI system in support of the AI Regulation [EUAI].

A.3 Zero-Trust architecture for AI

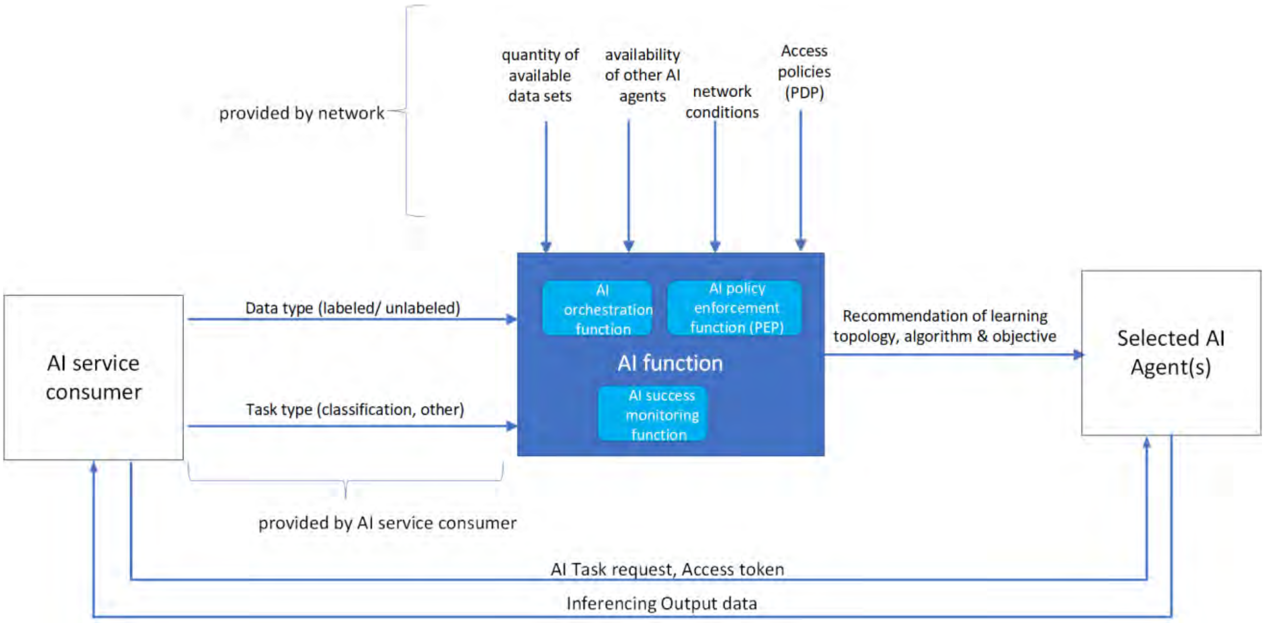

provided by network
quantity of available data sets
availability of other AI agents
network conditions
Access policies (PDP)
AI service consumer
Data type (labeled/ unlabeled)
Task type (classification, other)
AI orchestration function
AI policy enforcement function (PEP)
AI function
AI success monitoring function
Recommendation of learning topology, algorithm & objective
Selected AI Agent(s)
provided by AI service consumer
AI Task request, Access token
Inferencing Output data

Figure 9-2: Proposed Zero-Trust Architecture for AI.