AMAZING-6G
Amazing Large-Scale Trials and Pilots for Verticals in 6G

Deliverable 3.1

Technology Enablers – Intermediate Report

Co-funded by the European Union

6GSNS

AMAZING-6G project has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101192035.

Project Details

Call	HORIZON-JU-SNS-2024
Topic	HORIZON-JU-SNS-2024-STREAM-D-01-01
Project start date	01/01/2025
Duration	36 months
GA No	101192035

Deliverable Details

Deliverable WP:	WP3
Deliverable Identifier:	D3.1
Deliverable Title:	Technology Enablers – Intermediate Report
Editor(s):	Haibin Zhang (TNO), Maria Raftopoulou (TNO), Nina Slamnik-Krijestorac (IMEC), Giada Landi (NXW), Paulo Alexandre Duarte (CAPG)
Author(s):	Haibin Zhang (TNO), Maria Raftopoulou (TNO), Ewout Brandsma (TNO), Anthony Pages (TNO), Belma Turkovic (TNO), Yohan Toh (TNO), Ramon de Souza Schwartz (TNO), Stan van Nieuwamerongen (TNO), Pascal Heijnen (TNO), Sarah Lim Choi Keung (TNO), Daniele Ronzani (HPE), Nicola di Pietro (HPE), Robert Gdowski (ISRD), Maria Safianowska (ISRD), Daniele Brevi (LINKS), Edoardo Bonetto (LINKS), Andreas Georgakopoulos (WINGS), Raul Filipe Barbosa (CAPG), Paulo Alexandre Duarte (CAPG), Bruno Miguel Mendes (CAPG), Rui Ferreira (CAPG), Adriano Gois (CAPG), Yi Ma (UoS), Xu Chunmei (UoS), Dionysia Triantafyllopoulou (TUC), Shahab Ehsanfar (TUC), Giada Landi (NXW), Gabriele Scivoletto (NXW), Enrico Alberti (NXW), Nina Slamnik-Krijestorac (IMEC), Raul Cuervo Bello (IMEC), Xhulio Limani (IMEC), Zisis Maleas (CERTH), Georgios Barboutidis (CERTH), Evgenios Sochopoulos (CERTH), Christina Politi (UPAT), Panagiotis Papaioannou (UPAT), Antti Heikkinen (VTT), Mikko Uitto (VTT), Dragos Anca (SIMTEL), Mihai Neagu (SIMTEL), Cristian Petrache (ORO), Spyros Batistatos (P-NET), Christos Tranoris (P-NET), Vaia Kalokidou (P-NET), Eirini Tserga (ThPA), Christos Papadopoulos (ThPA), Dionysis Deligiannopoulos (ACRO), Mauro Agus (TIM), Alessandro Trogolo (TIM)

Reviewer(s):	François Carrez (UoS), Dionysia Triantafyllopoulou (TUC), Haesik Kim (VTT)
Submission Date:	31/10/2025
Dissemination Level:	PU

Disclaimer

The information and views set out in this deliverable are those of the author(s) and do not necessarily reflect 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.

Executive Summary

This deliverable presents the intermediate results of Work Package 3 (WP3) “Technology Enablers”. Considering the time of this deliverable (M10) in relation to the whole duration of WP3 (M1-M33), this deliverable focuses on identification, design and some initial development/implementation of the technology enablers. Eventual fine-tuning of the designs and final implementations will be reported in Deliverable D3.2 (M33).

The technology enablers are identified based on the needs of the AMAZING-6G use cases, with the aim that the identified enablers may also be used for other vertical use cases with similar requirements. In order for their integration in the AMAZING-6G trials and pilots, the identified technology enablers are particularly those where there are innovation opportunities and in which the project partners have strong expertise.

The identified technology enablers are grouped into the following four (4) categories. Some of the enablers are further split into multiple sub-enablers with similar but still different technical focuses:

- Communication enablers, including radio, transport and core network enablers
  - Communication resource management
  - Network slicing
  - Integrated sensing and communication (ISAC)
  - Public-private network integration
  - Multiple-RAT connectivity
- Compute as a Service enablers (CaaS), including the enablers in the compute continuum
  - Compute resource management
  - Compute continuum
  - Cloud-native service design
  - Creation of Kubernetes clusters on demand
- Application enablers and AI , including network exposure APIs and AI-driven application services
  - AI-as-a-Service framework
  - Digital Twin framework
  - OpenAPIs for Network Exposure
- IoT and localization enablers, including IoT platforms, IoT devices and localization technologies
  - Advanced Localization and Positioning Systems
  - IoT Connectivity and Infrastructure
  - Data Ingestion and Telemetry
  - IoT Service Platforms and Management
  - IoT Contextual Awareness Systems
  - Remote Control and Operation (Actuation)

Note that some of the technology enablers are relevant to each other or even overlap. For example, “Digital Twin” (at both application and network levels) is identified as a separate technology in the category of “Application enablers and AI”, while it’s also relevant for the collection of IoT data; ISAC is listed as one of the communication enablers, but the sensing part of ISAC is out of the scope of communication and may be related to IoT or Internet of Sense. The development progress of these enablers will be followed closely to ensure necessary coordination among the relevant enablers and involved partners.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

TERM	DESCRIPTION
3GPP	3rd Generation Partnership Project
5GA	5G Architecture
5GC	5G Core
5GTN	5G Test Network
5QI	5G QoS Identifier
AI	Artificial Intelligence
AIAAS	AI-as-a-Service
AGV	Automated Guided Vehicle
AMF	Access and Mobility Management Function
ANW P5G	Aruba Networking Private 5G
AP	Access Point
API	Application Programming Interface
AUSF	Authentication Server Function
AR	Augmented Reality
AS	Application Server
ATSSS	Access Traffic Steering, Switching, and Splitting
BLE	Bluetooth Low Energy
CAAS	Compute as a Service
CAPIF	Common API Framework
CI/CD	Continuous Integration / Continuous Deployment
CL	Closed Loop
CLI	Command Line Interface
CN	Core Network
CN-RAN	Cloud-native RAN
CNF	Containerized network function

CPE	Customer Premises Equipment
CPU	Central Processing Unit
CRD	Custom Resource Definition
COTS	Commercial Off-The-Shelf
CSI	Channel State Information
CU-CP	Centralized Unit - Control Plane
CU-UP	Centralized Unit - User Plane
COAP	Constrained Application Protocol
DT	Digital Twin
DU	Distributed Unit
E2E	End-to-End
EAP	Extensible Authentication Protocol
EAP-AKA	EAP– Authentication and Key Agreement
EC	European Commission
EFN	Energy Footprint Notification
EU	European Union
ERP	Enterprise Resource Planning
ETSI	European Telecommunications Standards Institute
EMS	Energy Management Systems
EDGEAPP	Edge Network Application
FAR	Forwarding Action Rule
FL	Federated Learning
FR1	Frequency Range 1
FR2	Frequency Range 2
FRMCS	Future Railway Mobile Communication System
GDBR	General Data Protection Regulation
GPDA	Generalized Probability Data Association

GPS	Global Positioning System
GPU	Graphics Processing Unit
GNB	Next Generation Node B
GNSS	Global Navigation Satellite System
GTP-U	GPRS Tunneling Protocol - User Plane
GSMA	Global System for Mobile Communications Association
GUI	Graphical User Interface
HARQ	Hybrid Automatic Repeat Request
HD	Hard Disk
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
HVAC	Heating, Ventilation, and Air Conditioning
HW	Hardware
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IP	Internet Protocol
IOT	Internet of Things
ISAC	Integrated Sensing and Communication
IBN	Intent-Based Networking
IME	Intent Management Entity
K8S	Kubernetes
K8SAAS	Kubernetes-as-a-Service
KPI	Key Performance Indicator
KVI	Key Value Indicator
LCX	Leaky Coaxial Cable
LIDAR	Light Detection and Ranging
LLM	Large Language Model

LMF	Location Management Function
LTE	Long-Term Evolution
LXC	Linux Containers
LORAWAN	Long Range Wide Area Network
MAC	Medium Access Control
MANO	Management and Orchestration
MC	Mission Critical
MCX	Mission Critical Services
MEC	Multi-access Edge Computing
MIMO	Multiple Input Multiple Output
ML	Machine Learning
MLOPS	Machine Learning Operations
MPTCP	Multipath Transmission Control Protocol
MQTT	Message Queuing Telemetry Transport
MSC	Message Sequence Chart
N3IWF	Non-3GPP Interworking Function
NDT	Network Digital Twin
NEF	Network Exposure Function
NF	Network Function
NFV	Network Function Virtualization
NFVI	Network Function Virtualization Infrastructure
NGAP	Next Generation Application Protocol
NR	New Radio
NRF	Network Repository Function
NSSAI	Network Slice Selection Assistance Information
NPN	Non-Public Network
NSSF	Network Slice Selection Function

NTN	Non-Terrestrial Network
NUC	Next Unit of Computing
NLP	Natural Language Processing
OAI	OpenAirInterface
OBU	On-Board Unit
ODA	Open Digital Architecture
OFH	Open Fronthaul
OPC UA	Open Platform Communications Unified Architecture
ONAP	Open Network Automation Platform
OSS	Operations Support System
PCF	Policy Control Function
PDU	Protocol Data Unit
PFCP	Packet Forwarding Control Protocol
PLC	Programmable Logic Controller
PLMN	Public Land Mobile Network
PPDR	Public Protection and Disaster Relief
PNI-NPN	Public Network-Integrated NPN
PRB	Physical Resource Block
PTP	Precision Time Protocol
PDU	Power Distribution Unit
PPDR	Public Protection and Disaster Response
QAM	Quadrature Amplitude Modulation
QoD	Quality on Demand
QoS	Quality of Service
QCI	QoS Class Identifier
QER	QoS Enforcement Rule
RAN	Radio Access Network

RAT	Radio Access Technology
RBAC	Role-Based Access Control
REC	Renewable Energy Community
REDCAP	Reduced Capability
RFS	Resource Facing Service
RIC	RAN Intelligent Controller
RRC	Radio Resource Control
RSU	Road Side Unit
RTK	Real-Time Kinematic
RAM	Random Access Memory
REST	Representational State Transfer
RTT	Round Trip Time
SB-EMS	Smart Building Energy Management System
SBA	Service-Based Architecture
SBI	Service-Based Interface
SCADA	Supervisory Control and Data Acquisition
SDN	Software-Defined Networking
SDR	Software-Defined Radio
SEAL	Service Enabler Architecture Layer
SIM	Subscriber Identity Module
SLA	Service Level Agreement
SLO	Service Level Objective
SMF	Session Management Function
SMS	Short Message Service
SW	Software
TF	Terraform
TN	Transport Network

TMF	TeleManagement Forum
UC	Use Case
UDR	Unified Data Repository
UE	User Equipment
UGV	Unmanned Ground Vehicle
UPF	User Plane Function
URLLC	Ultra-Reliable Low Latency Communication
V2X	Vehicle-to-Everything
VNF	Virtual Network Function
VRU	Vulnerable Road User
VR	Virtual Reality
VM	Virtual Machine
YAML	Yet Another Markup Language
ZSM	Zero-touch Network and Service Management
WEBRTC	Web Real-Time Communication

1 Introduction

This deliverable aims to present the intermediate results of Work Package 3 (WP3) "Technology Enablers". The major objectives of WP3 include:

- To identify relevant technology enablers, based on the use cases-based requirements (KPIs and KVIs) defined in Task T2.1 and the system architecture designed in Task T2.2.
- To design and develop the identified technology enablers.
- To facilitate WPs 4-6 in the integration of developed technology enablers in the trials and pilots.

Considering the time of this deliverable (M10) and the whole duration of WP3 (M1-M33), this deliverable focuses on identification, design and some initial development/implementation of the technology enablers. Eventual fine-tuning of the designs and final implementations will be reported in Deliverable D3.2 (M33).

1.1 Scope of AMAZING-6G technology enablers

AMAZING-6G addresses the following four (4) categories of technology enablers, for implementing trials and pilots:

- Communication enablers (the scope of Task T3.1), including radio, transport and core network enablers.
- Compute as a Service enablers (CaaS, the scope of Task T3.2), including the enablers in the compute continuum.
- Application enablers and AI (the scope of Task 3.3), including network exposure APIs and AI-driven application services.
- IoT and localization enablers (the scope of Task 3.4), including IoT platforms, IoT devices and localization technologies.

Figure 1-1 illustrates the scope of the AMAZING-6G technology enablers and their association with WP3 tasks. Note that Figure 1-1 is not aimed for the illustration of AMAZING-6G architectures. Interested readers are referred to Deliverable D2.1 for the initial AMAZING-6G architectures.

The technology enablers are identified based on the needs of the AMAZING-6G use cases, with the aim that the identified enablers may also be used for other vertical use cases with similar requirements. In order for their integration in the AMAZING-6G trials and pilots, the identified technology enablers are particularly those **where there are innovation opportunities and in which the project partners have strong expertise**. Thanks to the complementary expertise of the project partners, a sufficient broad spectrum of enablers can be covered for each of the four (4) categories.

Note that, next to the identified technology enablers, there are also other technology components which will be needed in the aimed trials and pilots. Non-complete examples of such technology components are macro- or small-cells, core networks, backhaul/fronthaul, user devices, MIMO technology, traffic scheduling, etc. These are either ready-to-be-used or easy-to-be-implemented, in any case not the focus of the AMAZING-6G innovation study (the performance of these components themselves and the influence of their configurations on the use cases performance). Such technology components will be part of the integration process in WP4/5/6 and involved in the overall use case associated tests in WP7, but not included as AMAZING-6G technology enablers in the WP3 work.

Some of the identified technology enablers may be composed of multiple "sub-enablers", each corresponding to a distinct technique with the same or similar purpose(s). This has been made possible, due to the fact that different partners are involved and different test facilities are used for different use cases, which share the same or similar needs among each other regarding one particular technology enabler, but have different constraints posed by the used test facilities. The association of project partners with the use cases is shown in Section 1.2, while the list of technology enablers (and their eventual sub-enablers) is shown in Section 1.3.

Due to current and predicted **hardware and software unavailability**, there are a small number of (sub-)enablers which are identified as important for one or more of the AMAZING-6G use cases but not aimed for implementation during the project. The consortium will monitor the availabilities of such hardware and software. In case of changes during the project, the involved partners will evaluate the feasibility of implementing the associated (sub-)enabler(s) by considering the potential impact (time and budget-wise) on other implementation activities. Such eventual changes will be reflected in Deliverable D3.2.

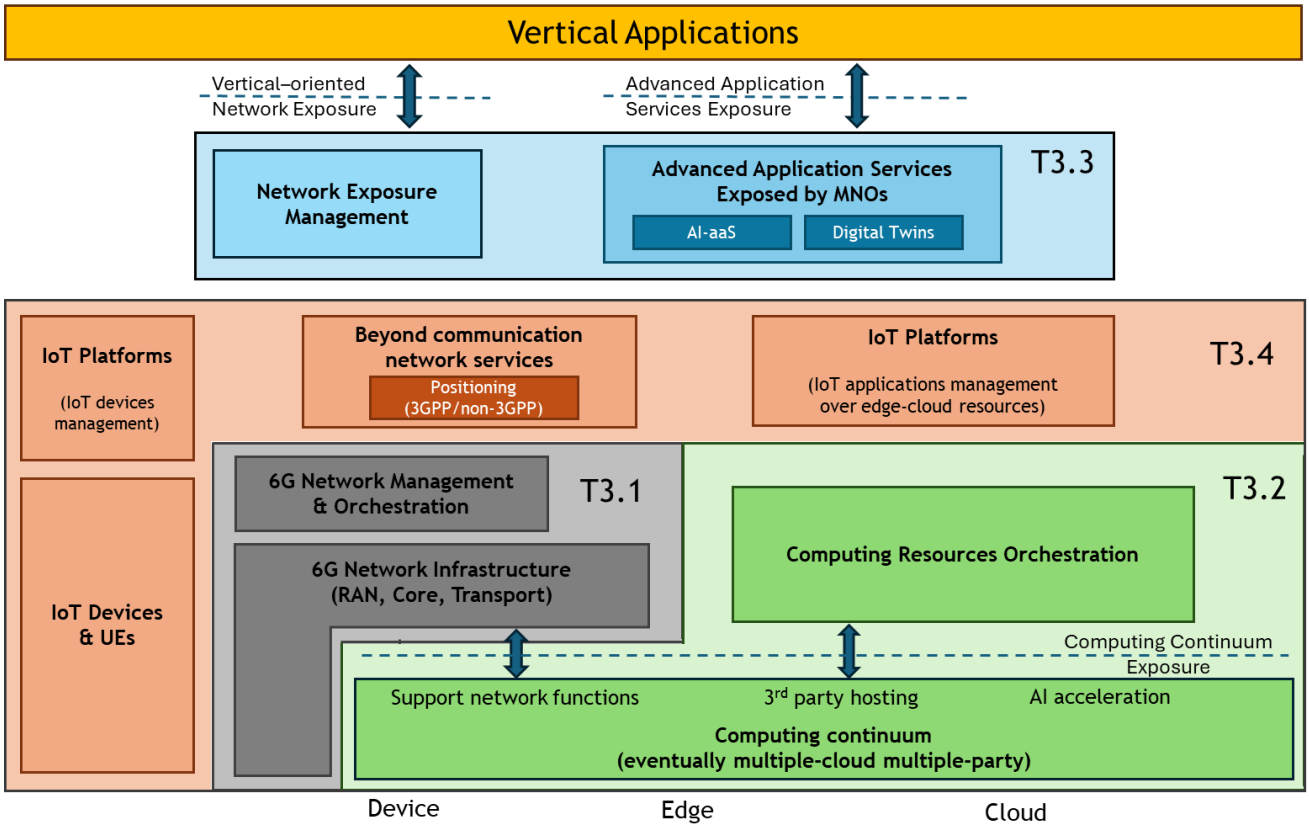

Figure 1-1 Scope of AMAZING-6G technology enablers

1.2 Overview UC-partner association

Table 1-1 shows the association of the project partners with the AMAZING-6G use cases (UCs), including the high-level information of the technology enabler categories each partner is working on and the associated UC(s). The readers may refer to Deliverable D2.1 for the numbering of the UCs and their detailed specification. It's admitted that, for a few UCs (P2, P3, P4, T3) not all the four technology enabler categories are addressed in the project, due to practical constraints (involved partners, budget and test facility limits).

Table 1-1 UC-partner association

Use cases	Communication enablers	CaaS enablers	Application enablers and AI	IoT and localization enablers
H1	TNO	TNO	TNO	TNO
H2	TNO	TNO	TNO	TNO
P1	PNET, UPAT	PNET, UPAT	UPAT, WINGS	WINGS

P2	VTT		VTT	
P3	VTT		VTT	
P4	VTT		VTT	
P5	ORO	ORO	ORO	ORO
E1	CAPG	CAPG, NXW	CAPG, NXW	NXW
E2	TNO	TNO	TNO	TNO, CAPG
E3	ORO, CAPG	ORO	ORO, CAPG	ORO, SIM
T1	HPE, LINKS, TIM	LINKS, NXW	LINKS, NXW	LINKS, NXW
T2	HPE, LINKS, TIM	LINKS, NXW	LINKS, NXW	LINKS, NXW
T3	TUC, UoS		TUC, UoS	TUC
T4	IMEC, TUC, ISRD	IMEC, ISRD	IMEC, TUC	TUC
T5	ACRO, CERTH, ThPA	CERTH, ThPA	CERTH, ThPA, WINGS	CERTH, ThPA

1.3 List of technology enablers and their use case association

Table 1-2 shows the list of enablers and applied use cases. As mentioned above, for some of the enablers we have further specified multiple sub-enablers. More details of the enablers and their use case relevance are described in the following chapters.

Table 1-2 List of enablers, sub-enablers and applied use cases

Technology enablers	Sub-enablers	Applied use cases
Communication enablers		
Communication resource management	Zero-touch network and service management	E1, E3, T4, T5
	Network programmability	T4
	Automated gNB configuration	P1, T4, T5
	Network performance monitoring and control	P1, P2, P4, E1, E3, T1, T2, T4, T5
	Identification and selection of backhauling	P1, T5

Network slicing		H1, H2, P3, E1, E2, E3, T1, T2, T3, T4
Integrated sensing and communication		T3
Public-private network integration		E2, T1, T2, T5
Multiple-RAT connectivity	Sidelink	H1, H2, T3
	Wi-Fi integration	H1, H2, P2, P4, T5
	NTN integration	P2, E2
Compute-as-a-Service enablers		
Compute resource management	Intelligent service and resource orchestration in extreme edge/edge/cloud compute continuum	P1, E1, T1, T2, T4, T5
	Intent-based service and resource management	E1, T4
	Real-time monitoring of compute resources and energy consumption in compute continuum	T1, T2, T4
Compute continuum	Distributed user-edge-cloud compute continuum	H1, H2, E1, E2, T1, T2, T4
	Multi-party cloud	H1, H2, E2
Cloud-native service design		E1, T4
Creation of Kubernetes clusters on demand		P1
Application enablers and AI		
AI-as-a-Service framework	AI-as-a-Service	P1, E1, E3, T2
	ML models catalogue	P1, E1, E3, T5
	MLOps	E1, T5
Digital Twin framework		T1, T2, T5
OpenAPIs for Network Exposure	TM Forum API	P1

	CAMARA APIs for Quality on Demand	H1, H2, P2, P3, P4, E2, T1, T2, T4
	CAMARA APIs for Energy management	P2, P3, P4, T2
	CAMARA APIs for Edge/Cloud	H1, H2, E2, T1, T2, T4
IoT and localization enablers		
Advanced Localization and Positioning Systems	Hybrid Localization (5G + GNSS)	E2, T1, T2, T5
	Cooperative Positioning	T1, T2
	GNSS and RTK Positioning	E2
IoT Connectivity and Infrastructure	IoT Gateways	P1, E3, T1, T2
	5G Redcap IoT	H1, H2
	IoT Sensors and Devices	P1, E3, T1
Data Ingestion and Telemetry	Data Collection from sensors and devices	E1, E3, T3, T4,
	Monitoring Platform	P1, E1
IoT Service Platforms and Management	IoT Sensors Orchestration (	P1, E1, T1, T2
	- IoT Resource Orchestration - IoT Resource Registry - IoT Service Registry	P1, E1, E2, T1, T2
	)	P1, E1, T1, T2
	Service and Resource Lifecycle Manager	P1, E1, T1, T2
	Message queue fabric	P1, E1, T1, T2
	IoT Exposure Function	P1, E1, T1, T2, T5
IoT Contextual Awareness Systems		E3, T4
Remote Control and Operation (Actuation)	Real-Time Actuation/Teleoperation	P5, E3, T5

1.4 Structure of the deliverable

The rest of the deliverable is organised as follows. Chapters 2-5 are dedicated to each of the technology enabler categories:

- Chapter 2 - Communication enablers;
- Chapter 3 - Compute-as-a-Service (CaaS) enablers;
- Chapter 4 - Application enablers and AI;
- Chapter 5 - IoT and localization.

For each of the identified technology enablers, the following content is given in their respective sections: a brief description of the enabler and (if applicable) their sub-enablers , use case relevant and associated partners, overall design and implementation plan, and eventually preliminary implementation results.

Concluding remarks are given in Chapter 6.

2 Communication enablers

In this chapter, we address the communication enablers as identified in Table 1-2. These enablers are driven by the communication needs of the AMAZING-6G use cases, and may be also applicable for other vertical use cases with similar needs:

- Communication resource management, which enables real-time network performance monitoring and control, as well as efficient use of various communication resources in the network.
- Network slicing, which enables the same network infrastructure to be shared among different users and different types of services, ensuring resource isolation among the slices.
- Integrated sensing and communication (ISAC), which enables the same B5G/6G network to provide both communication and sensing capabilities.
- Public-private network integration, which enables seamless movement of users between private and public network infrastructure.
- Multiple-RAT connectivity, which ensures network coverage and resilience even in difficult-to-reach areas.

2.1 Communication resource management

The demand for network resource consumption is continuously growing due to the proliferation of IoT devices, user equipment (UE), and other connected endpoints. To address this demand, network management must evolve to become more flexible and resilient, enabling dynamic resource allocation, service scalability, and operational flexibility. This evolution is supported by programmability through Software-Defined Networking (SDN) and Network Function Virtualization (NFV), which also underpin critical services like ultra-reliable low-latency communications (URLLC) for latency-sensitive applications. Challenging use cases such as autonomous driving highlight the importance of this approach: Ultra-reliable teleoperation services, where human operators can remotely intervene to protect vulnerable road users (VRUs), demand stringent <30ms end-to-end (E2E) latency, dynamic resource allocation across heterogeneous networks, and seamless service continuity during high-mobility transitions. While programmability provides flexibility, it also increases complexity: traditional MANO frameworks and human operators cannot keep pace with rapidly changing conditions.

This enabler aims to autonomously manage various communication resources in different parts (radio, backhaul, core) of the network, to fulfill the KPI and KVI requirements of AMAZING-6G use cases.

2.1.1 Description of the enabler

This enabler consists of the following sub-enablers, addressing different aspects of communication resource management and applicable for different AMAZING-6G use cases: Zero-touch Network and Service Management (ZSM), network programmability, automated gNB configuration, network performance monitoring and control, (dynamic) identification and selection of backhauling.

Zero-touch Network and Service Management (ZSM)

ZSM aims to enable autonomous networks capable of self-managing, based on service-level policies and rules, without human intervention. It is driven by intent-based input (from operators, verticals or customers), closed-loop automation, AI/ML-driven decision-making, and policy-driven orchestration. It outlines a service-based architecture (SBA) composed of functional domains and management services that interact through well-defined APIs. It can support critical functions such as policy configuration, subscriber onboarding, network slice orchestration, and management of containerized network functions (CNFs) and virtual network functions (VNFs). This sub-enabler is particularly tailored to address vertical domains where automation, reliability, and scalability are essential.

ZSM provides the following core capabilities:

- Intent Interpretation and Policy Enforcement: high level goals are converted into executable policies.

- Network slice Lifecycle Management: dynamically instantiates, scales and terminates slices across domains to serve vertical use cases with specific SLAs (e.g., for smart grid fault detection or V2X communication).
- Policy and subscriber management: applied dynamic access, QoS, and security rules tailored to subscribers, devices or applications.
- Orchestration of VNFs/CNFs automatically deploys and manages the lifecycle of those virtualized and cloud-native functions on cloud, edge, or hybrid infrastructure.
- Closed-Loop Service Assurance: continuously monitors KPIs, applies AI/ML mechanisms to detect anomalies, service degradation, and triggers automated remediation if some anomalies or degradation is detected.

Figure 2-1 illustrates the standardised framework. Each domain hosts modular management functions for orchestrating resources and services (e.g., E2E orchestration, analytics, intelligence, and monitoring). These operate in closed-loop cycles, enabling networks to continuously sense, analyse, and adapt. To support interoperability, each domain exposes capabilities through a Domain Integration Fabric, responsible for communication, authentication, and data exchange. Data services ensure persistent storage and exchange of telemetry and metadata, providing consistent datasets for analytics and orchestration. Together, these components create a federated architecture where multiple domains can collaborate seamlessly while remaining independently manageable.

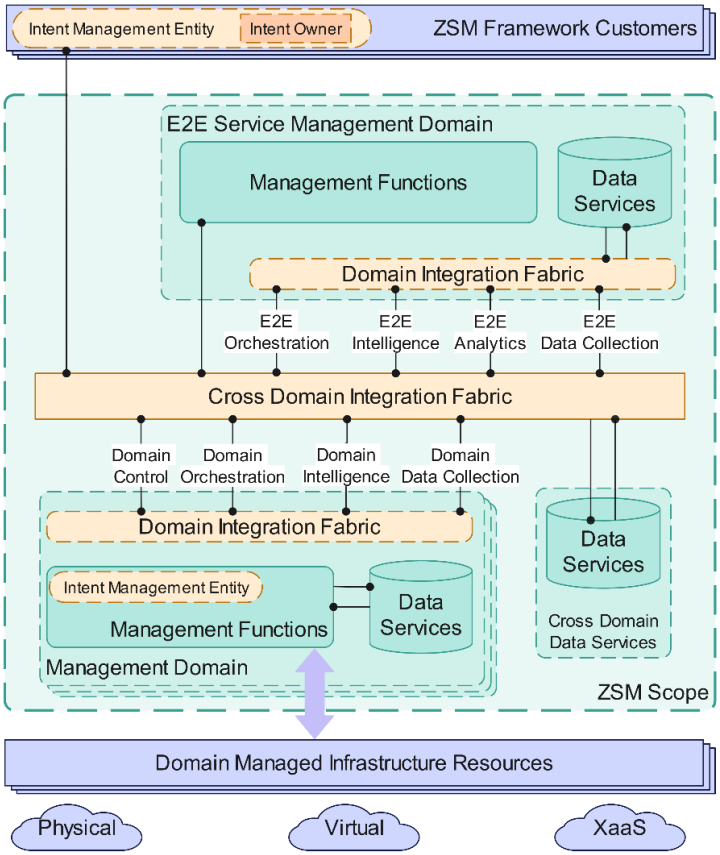

Figure 2-1 Zero-touch network and Service Management Framework reference architecture by ETSI

The ZSM paradigm, based on this standardized architecture, enables the practical management of heterogeneous and dynamic B5G/6G networks. In the energy sector, ZSM allows dynamic provisioning of edge resources for real-time grid monitoring and SLA assurance. In transportation, it supports V2X and teleoperation services, ensuring ultra-low latency, predictive orchestration, and autonomous reconfiguration. Several industry-standard frameworks complement the ZSM approach, including OSM (NFV orchestration across multi-cloud environments), Nephio (Kubernetes-native intent automation), and ONAP (policy control, orchestration, and closed-loop analytics).

Network programmability

Many emerging applications still face difficulties in unusual and unpredictable situations, such as highly dynamic environments, adverse operating conditions that degrade sensor performance, ambiguous contextual information, or circumstances in which on-board intelligence cannot guarantee safe or optimal decisions. To function reliably in such scenarios, the supporting network must fulfil stringent performance requirements in terms of i) latency, ii) bandwidth, and iii) reliability to ensure uninterrupted connectivity. Legacy networks, built on fixed-function hardware and static configurations, struggle to meet such demands. The absence of flexible resource allocation and adaptive network reconfiguration e.g., migrating Network Functions (NFs) from the cloud to the edge, limits the ability of current infrastructures to handle changing conditions, mobility, and fluctuating workloads. Network programmability addresses these limitations by rethinking the network as a software-based platform rather than a set of hardware devices. Programmability means that network behavior, from packet forwarding decisions to resource allocation (radio or computing), can be modified via (standardized) software interfaces in real time without any physical intervention. This paradigm allows the network to dynamically instantiate services where needed (NF at the edge or cloud), scale resources based on actual demand, and reconfigure network behavior to meet service requirements (change resource allocation).

Table 2-1 Network Programmability technologies

Technology	Purpose	Implementation in Network Domains	Benefits for teleoperation (UC T4)
Software Defined Networks (SDN)	Separate control plane from data plane, enables centralized network control	Control plane centralization across RAN, TN, and CN; Standardized interfaces	Real-time path optimization, dynamic traffic routing, sub-20ms latency guarantee
Network Function Virtualization (NFV)	Transforms hardware network functions into software applications	NFs deployment in edge/cloud locations, containerized network services	Dynamic UPF placement at edge, scalable traffic shaping, on-demand firewall deployment
Virtualization Technologies	Provide abstraction layer for resource pooling and isolation	VMs, containers, pods across all network domains; Kubernetes orchestration	Isolated teleoperation sessions, guaranteed resource allocation, rapid service deployment

The deployment of network programmability relies on the use of complementary technologies, namely Software Defined Networks (SDN) and Network Function Virtualization (NFV). SDN establishes programmable control by separating the control plane from the data plane, creating a logical architecture in which centralized controllers have the global state of the network and sends the control messages through standardized interfaces (such as Openflow [5]). In Table 2-1, we summarize how NFV and SDN work synergistically to guarantee the specific network requirements for vertical services within 5G SA networks. A specific example of a vertical service shown in the table is associated with the T4 use case (teleoperation services). For example, to start a teleoperation session, the SDN controller can dynamically configure the optimal data flow paths taking into account the current network conditions, by configuring the end-to-end path based on real-time network performance. On the other hand, NFV transforms network services from hardware devices into software applications, enabling functions such as traffic shaping, firewalls, congestion mechanisms, virtual switches, and routers. NFV allows these functions to be instantiated exactly where needed using virtualization techniques. Virtualization technologies (e.g., Docker [6], LXC [7], Kubernetes [8]) provide the fundamental level of abstraction that allows network resources to be grouped, shared, and dynamically allocated, while maintaining isolation between services. Through virtualization, physical data centers become pools of computing resources capable of hosting multiple network functions. Therefore, with proper resource orchestration, these network resources can be used more flexibly to accommodate large number of vertical services.

Automated gNB configuration

Automated gNB configuration represents a fundamental paradigm shift in the management of the Radio Access Network (RAN). It moves away from the traditional, manual, command-line interface (CLI) based configuration of individual 5G/6G base stations (gNBs) towards a software-driven, API-centric, and declarative automation model. This enables the flexibility and efficiency required by the future autonomous networks and 6G. In AMAZING-6G, this will facilitate:

- Extreme Dynamic Resource Sharing: 6G envisions a fabric of interconnected networks (non-terrestrial, terrestrial, private) supporting countless heterogeneous devices and applications. Manual configuration cannot scale. Automated, API-driven configuration is the only way to rapidly reassign spectrum and other radio resources.
- Massive Scale and Complexity: 6G networks will be vastly more complex, with integrated sensing, AI-native operation, and sub-networks. Automation through APIs is essential and this will only be feasible if the gNBs are automation-ready.
- Ultra-Low Latency Service Deployment: The ability to reconfigure the RAN near-instantaneously via API calls is prerequisite for providing guaranteed latency and reliability for critical applications like PPDR that will be showcased in AMAZING-6G.
- Energy Efficiency: Automated configuration allows gNBs to dynamically scale power, shut down cells, or adjust parameters in real-time based on traffic load, moving far beyond today's static power-saving features.

Following are the main characteristics of this sub-enabler in AMAZING-6G:

- **Abstraction**: The complex, low-level technical parameters of a gNB (e.g., frequency, bandwidth, PLMN, slice definitions, TDD patterns) are abstracted into a high-level profile simplifying and automating the procedure by specifying the desired end-state and declaring the intent.
- **API-driven control**: This will act as the universal control plane for any gNB providing a single, consistent interface for any authorized system, orchestration platform or controller to interact with the network resources.

Network performance monitoring and control

The possibility to control and monitor the network and its performance is a horizontal enabler that applies to a wide range of use cases. The objective of B5G/6G networks is to deliver ultra-reliable, low-latency, and high-capacity communications tailored to diverse application requirements or vertical use cases. To effectively reach this objective, continuous and ideally real-time performance monitoring is crucial to gain insights into the network's behavior at any given moment.

Network slicing is a key technology of 5G/6G (see Section 2.2), and it comes with Service Level Agreements (SLAs) that require continuous monitoring of network status, including latency, reliability, bandwidth utilization, jitter, and other metrics. This implies the adoption of new monitoring technologies, since traditional approaches are no longer sufficient to provide the necessary level of granularity and all the information needed for fine-grained network control. This involves not only network-related measurements but also computation-related indicators. Tools such as Prometheus, API-based technologies, Grafana and others offer different perspectives on monitored metrics, which can also be collected by third-party systems for analytics or predictive modeling.

In case of O-RAN, network performance monitoring, and control in the RAN should be extended, for example through specialized xApps running on the Near-Real-Time (Near-RT) RAN Intelligent Controller (RIC). xApp continuously collects KPIs, such as throughput, PRB utilization or number of RRC connections, via standardized E2 interfaces. The gathered data is then processed and exposed either to other co-located xApps or to external entities through northbound interfaces. This enables both internal and external consumers to gain real-time visibility into the network state, while also allowing automated control loops to be implemented.

The large amount of collected data requires AI/ ML-based processing capable of automatically detecting deviations from standard network behavior and acting in real time to maintain performance or adapt operations in accordance with general policies. Another important building block for monitoring and control is the concept of the Network Digital Twin (see Chapter 4).

From a practical point of view, the current 5G networks are often managed by a proprietary Element Management system, that constantly collects performance monitoring data in each gNB. It allows full control and monitoring of the network parameters and performance. This will be extended to have more comprehensive monitoring and control of the whole network towards end-to-end QoS control.

Identification and selection of backhauling

2In emergency use cases where a standalone B5G/6G network is deployed on demand, there is a requirement for identifying and ultimately selecting backhaul connectivity options among the available ones, to extend coverage and enhance services provided to the personnel operating in the emergency. The aim is to find possible backhaul solutions to achieve connectivity from the deployed area to the public Internet, which might be either pre-existing infrastructure (e.g. fiber), mmWave links that can be deployed on demand, existing 5G/6G connections from other vendors or non-terrestrial network connection if such an option is available. Evidently such a mechanism could be very beneficial for both fast network provisioning and for resiliency purposes in case of backhaul failure. This mechanism assumes the existence of redundant connectivity and an observability mechanism that reports on the performance of these options so that the sub-enabler can identify and select the suitable option.

2.1.2 Use case association and contributing partners

Table 2-2 shows the association of AMAZING-6G use cases with the communication resource management sub-enablers.

Table 2-2 Mapping between the communication resource management sub-enablers and the use cases

	ZSM	Network Programmability	Automated gNB configuration	Network performance monitoring and control	Identification and selection of backhauling
P1			PNET, UPAT	PNET, UPAT	PNET, UPAT
P2				VTT	
P4				VTT	
E1	CAPG			CAPG	
E3	CAPG			ORO	
T1				HPE, LINKS, TIM	
T2				HPE, LINKS, TIM	
T4	IMEC	IMEC, ISRD	ISRD	IMEC, ISRD	
T5	CERTH, ThPA		ACRO	ACRO, ThPA	ACRO

For the **P1 use case**, the following 3 sub-enablers are relevant:

- **Automated gNB Configuration**. The sub-enabler is extremely important for P1 as it enables rapid deployment of temporary or mobile gNBs in disaster areas (e.g., portable/mobile 6G cells and drones) that can be reliably configured and offer optimal coverage and capacity for AR/VR streaming to PPDR control centers with reduced manual intervention, which is crucial in emergency response scenarios. This means that the PPDR actors will need to deploy the mobile/portable cell in the affected area, in order to ensure coverage and high-performance network availability to support their scenario with fast provisioning and configuration of suitable gNB parameters.
- **Network Performance Monitoring and Control**. We need real-time analytics and closed-loop control of KPIs (latency, throughput, reliability) with AI-driven network management, hence advanced network observability is required that guarantees ultra-reliable low-latency communication for mission-critical data and possible AR/VR feeds. Furthermore in order to monitor and control the slice deployment that serves the rescuers and their PPDR campaign, slice performance monitoring ensures the PPDR slice maintains priority across operators and/or private /public network deployment. Advanced monitoring may allow proactive detection of congestion or failures, triggering automatic resource scaling and network re-configuration.
- **Identification and Selection of Backhauling**. In the P1 case, the availability of backhaul network to interconnect an isolated cell is not guaranteed. However, the high-performance slice that may be provisioned assumes high quality transport network availability especially when interconnection with cloud resources and/or remote (central) control center is assumed. Intelligent selection of backhaul links among available ones based on performance requirements and backhaul availability is vital for interconnecting the mobile private network with the central infrastructure. Furthermore, in emergencies terrestrial backhaul may be damaged—automated selection ensures fallback to satellite. Especially once an AR enabling service is established, maintaining continuous connectivity for AR/VR control centers even in disrupted infrastructures is important for the rescuers. A high performance PPDR slice assumes an optimized backhaul selection to meet the needs of AR/VR video streams.

For the **P2 and P4 use cases**, the sub-enabler **Network Performance Monitoring and Control** is applicable:

- The P2 use case aims to demonstrate and investigate the B5G/6G mission critical services interoperability relying with other non-3GPP systems across national networks. Due to this, operability, reliability and QoS can be ensured in mission critical operations such as in P4. This requires accurate and real-time network performance monitoring and control in order to select best available wireless network to meet the QoS requirements. To achieve this, the testbed in conjunction with the network infrastructure must be able to form and maintain accurate time synchronization between the nodes of interest and to collect real-time performance indicators from the system. These indicators may contain not only QoS (e.g., delay, packet loss, throughput) but also ones related to energy and processing factors, such as momentary power consumption and CPU/GPU usage which can help on maintaining the battery-powered devices online longer.
- The P4 use case aims to demonstrate arctic area search and rescue operation in northern Finland. The areas of natural disasters, such as avalanches, may lack of decent mobile connectivity vital for communication between first responders and remote control site instructing the operations. Due to this and similarly as in P2, the availability of non-3GPP networks may form a better option for reliable connection, but its identification and selection requires advanced network monitoring and control system for mobile devices in the field relying only on wireless connectivity (aka tactical bubble). Furthermore, some of the on-site services may have strict QoS requirements for example in terms of end-to-end latency, which means that these services should be prioritized by using slicing or multi-connectivity techniques. As on-site devices are onboarded to eSleds (or similar battery-powered devices) energy monitoring alongside with network is essential for longer operation times.

For the **E1 and E3 use cases**, the sub-enablers **ZSM** and **Network Performance Monitoring and Control** are applicable:

- The E1 use case on Renewable Energy Communities (REC) adopts a Service and a Resource Orchestrator to deploy cloud-native application components of Energy Management Systems (EMS) across a computing continuum involving cloud resources, devices, and edge nodes within smart buildings. Effective communication resource management ensures seamless data exchange between smart buildings and edge/cloud platforms, supporting real-time telemetry collection, remote actuation, and collaborative optimization across RECs. Differentiated QoS levels and dynamic resource provisioning prioritize traffic related to comfort management, energy control, and user interaction, while dedicated network slices isolate energy-related data to maintain security, low latency, and reliability for critical control loops. The setup integrates a ZSM which acts as a decision-making layer for translating service intents into network configurations. This enables orchestration framework to automatically associate the most pre-configured network slice with each service request, configure the appropriate QoS profiles and interpret real-time network monitoring to adjust resources properly.
- In the context of E3, the enabler supports reliable and scalable communication between field-level edge devices, the B5G/6G communication layer, and the cloud-based orchestration platform. Using B5G/6G connectivity with cost-efficient 5G RedCap devices, telemetry from inverters and IoT gateways is transported with high reliability and low latency, while actuation commands are delivered securely and in real time. To meet mission-critical requirements, the enabler provides support for Ultra-Reliable Low-Latency Communications (URLLC). Network slicing isolates traffic such as inverter reconfiguration commands from less time-sensitive telemetry flows. Continuous monitoring of KPIs (latency, jitter, packet loss), combined with closed-loop assurance and edge fallback mechanisms, guarantees uninterrupted control and resilience even in degraded conditions. In addition, orchestration, automation, and assurance functions integrate AI/ML-based forecasting and optimization into the overall control loop, enabling solar energy systems to adapt dynamically to changing operational and environmental conditions while meeting performance requirements.

For the **T1 and T2 use cases**, the sub-enabler **Network Performance Monitoring and Control** is applicable:

- The use case T1 on “Protection of vulnerable road users” integrates a Resource Orchestrator to coordinate the deployment of containerized application components for roads monitoring, collection of data from vehicles and real-time assessment of risks for pedestrians and vulnerable road users. The orchestration logic dynamically distributes the tasks among Road Side Units (RSU) equipped with solar panels and edge nodes. The placement and task offloading decisions take input from the real-time monitoring, in order to jointly consider availability of computing resources, power consumption and RSU battery level. Application components and tasks migration procedures guarantee service continuity.
- The use case T2 on “Improving urban safety with UGV monitoring” implements the same sub-enabler as in T1, but handling workloads that can be deployed in a computing continuum extended to Unmanned Ground Vehicles (UGV). The real-time monitoring collects data related to usage of computing resources, power consumption and battery level of the UGVs to feed placement and offloading decisions.

For the **T4 use case**, the sub-enablers **ZSM, Network Programmability, Automated gNB configuration** and **Network Performance Monitoring and Control** are applicable: The use case addresses situations where autonomous driving cannot safely continue (e.g., adverse weather, construction zones, complex traffic) by seamlessly transferring control to a remote operator. The ZSM framework contributes to this by providing autonomous, intent-based orchestration and management of network and compute resources, enabling consistent QoS, ultra-low latency, and resilient connectivity for mission-critical teleoperation. Concerning network programmability, real-time path optimization, dynamic traffic routing, sub-20ms latency guarantee are possible, due to the flexibility in deploying application services

and UPF instances at optimal locations (e.g., edge). In addition, network programmability also enables isolated teleoperation sessions, guaranteed resource allocation, and rapid service deployment.

For the **T5 use case**, the sub-enablers **ZSM, Automated gNB configuration, Network Performance Monitoring and Control are applicable** and **Identification and Selection of Backhauling:** The T5 use case aims to demonstrate efficient and safe real-life, end-to-end port processes by utilizing B5G network, namely the teleoperation of an Ship-to-Shore (STS) crane in the Port of Thessaloniki. A STS crane is a sophisticated piece of equipment that is used to load and unload containers in port terminals, operating in a safety-critical environment, which involves workers, ships, vehicles, and valuable cargo. To remotely operate an STS crane over a dedicated B5G, it requires absolute reliability, ultra-low latency, and continuous service assurance. Network performance monitoring and control is essential to guarantee safe, real-time operation by maintaining strict latency and throughput requirements, ensuring high-quality video and possibly haptic feedback, while enabling rapid detection of anomalies before they impact crane movements. At the same time, the identification and selection of backhaul links among the available ones is critical to deliver the low-latency, high-capacity, redundant, and secure connectivity that keeps operations running smoothly end to end.

To manage this complexity at scale, ZSM acts as the glue that keeps the system reliable, adaptive, and cost-efficient. By automating provisioning, monitoring, optimization, and self-healing, ZSM ensures SLA compliance, rapid fault recovery, and seamless adaptation to dynamic port conditions such as vessel arrivals or variable crane workloads. Together, these enablers ensure that STS crane teleoperation is not only technically feasible but also safe, resilient, and economically sustainable in modern smart ports.

2.1.3 Design, development and implementation

In this section, we first illustrate the overall high-level design concepts for the sub-enablers, followed by testbed specific (associated with different use cases) design, development and implementation.

High-level design for ZSM: Figure 2-2 shows a high-level design for ZSM. The architecture follows the ETSI ZSM framework and is structured into multiple layers to enable intent-driven, closed-loop automation across heterogeneous network domains. At the top, the ZSM layer interacts with vertical intent systems, business management platforms, or automated customers. This allows high-level goals, expressed as intents, to be ingested into the framework. The E2E Service Management layer provides the core intelligence of the system, including modules for intent and policy management, orchestration and analytics, closed-loop assurance, and AI/ML-driven intelligence (GenAI/AIaaS). This layer translates abstract intents into actionable workflows, ensures that services are deployed according to SLA requirements, and triggers automated remediation actions when anomalies or degradations are detected. The Exposure APIs layer provides interoperability and programmability, leveraging frameworks such as CAMARA, CAPIF, and EDGE APP. These APIs allow vertical services and external applications to dynamically request QoS upgrades, slice selection, or access network insights without needing telco-specific knowledge. The Domain Management layer coordinates resources across RAN, Core, Cloud, and Edge domains. This includes both physical and virtual infrastructure, where orchestration mechanisms (e.g., Nephio, NFV MANO, Kubernetes-based controllers) ensure that network functions and applications are deployed, scaled, and optimized consistently across multiple environments. Finally, the architecture anchors into the underlying infrastructure, where IoT devices, RSUs, UGVs, smart buildings, and other domain-specific assets connect. This layer represents the operational environment where the E2E orchestration and assurance loops are enforced. By combining intent-based management, domain programmability, and exposure APIs, the design enables autonomous service provisioning, SLA-compliant resource allocation, and continuous assurance across use cases such as PPDR, renewable energy communities, smart ports, and teleoperation.

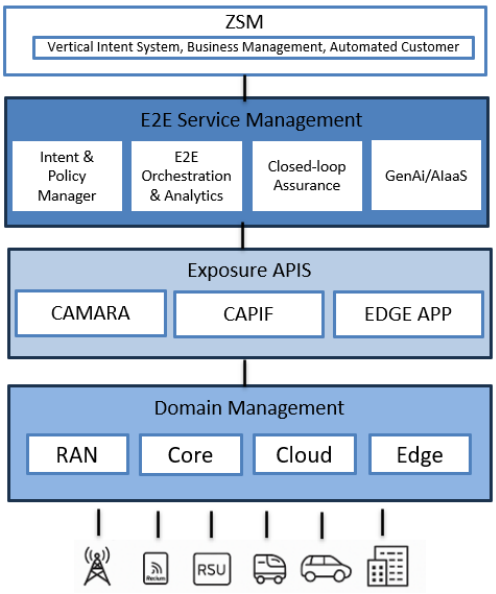

Figure 2-2 High-level design for ZSM

High-level design for Network Programmability: In Figure 2-3, we illustrate the high-level design of a fully programmable end-to-end B5G network, embodying the principles of software-based network and service design, which bring flexibility in resource and service provisioning. In particular, the core network, built on Service-Based Architecture (SBA) principles, consists entirely of cloud-native NFs that communicate through standardized service-based interfaces (SBIs). NFs such as the Access and Mobility Management Function (AMF), Session Management Function (SMF), and User Plane Function (UPF) can be dynamically instantiated, scaled, and relocated between edge and/or central clouds based on the service demands. The UPF, particularly critical for the data plane, can be deployed at the edge to minimize end-to-end latency, automatically reconfiguring the data plane to route traffic through the optimal UPF instance.

The transport network leverages programmable switches and routers that can be configured through SDN controllers to establish latency-guaranteed paths between the RAN and edge (or cloud) computing locations. In the RAN, programmability enables dynamic spectrum allocation, beamforming configurations or PRBs allocation, allowing the RAN to adapt its behavior based on the requirements. In the case of O-RAN (e.g. used in a subset of activities associated with the T4 use case), this is achieved primarily through the Near-Real-Time RIC (RAN Intelligent Controller), which enables closed-loop control on sub-second timescales. The Near-RT RIC hosts xApps, which ingest network performance data (KPIs) and issue configuration updates toward the RAN functions over the E2 interface. This introduces a structured way of decoupling policy and control from vendor-specific network elements, ensuring agility, interoperability, and automation in managing complex multi-slice environments.

Hence, network programmability serves as the foundational layer that provides both the mechanisms for dynamic resource allocation and the standardized interfaces that enable communication between network domains and functions. On top of that, advanced features like ZSM and Network Slicing will be used in AMAZING-6G to deliver the sophisticated orchestration and service differentiation required for mission-critical services.

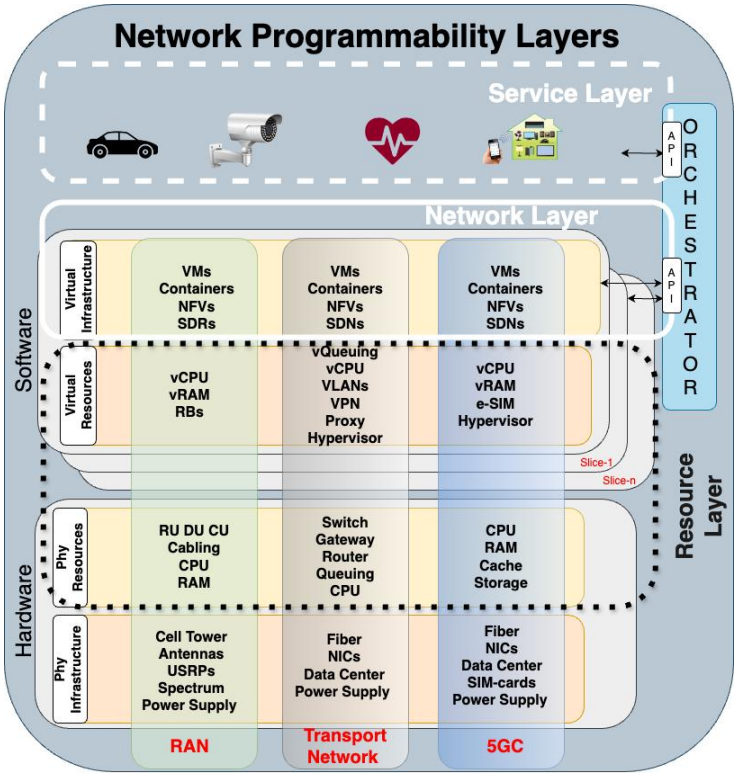

Figure 2-3 High-level design of Network Programmability, i.e., fully programmable end-to-end Beyond 5G networks

High-level design for Automated gNB Configuration: A high level design for the sub-enabler is shown in Figure 2-4. It comprises interfaces with gNB and 5G core in a way that the integration of the gNB configuration into a larger automated workflow will be explored. An orchestrator could automatically reconfigure gNBs in real-time based on triggers like changing network demand, energy prices, or specific vertical application requirements (e.g., instantly requiring ultra-low latency for an emergency service). Furthermore, network slices and RAN configurations are to be deployed and updated in real time for the deployment of the various services and use cases while an AI engine can analyze network performance data, make decisions on optimal parameters, and push new configurations to the gNBs automatically, creating a self-optimizing network.

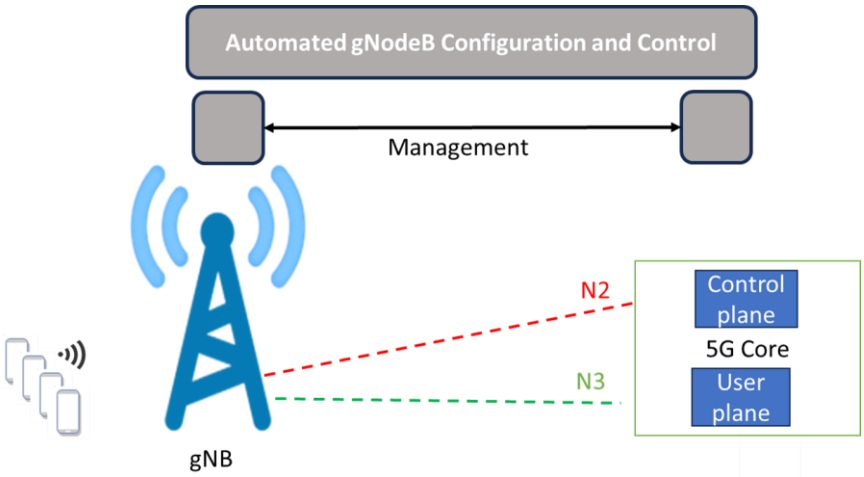

Figure 2-4 5G/6G gNodeB configuration enabler

High-level design for Network Performance Monitoring and Control: In Figure 2-5 a layered architecture for Network Performance Monitoring and Control is presented.

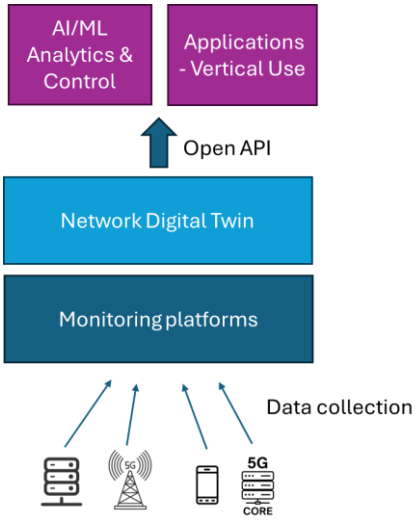

Figure 2-5 High-level design for Network Performance Monitoring and Control

At the foundation, data is collected from different elements of the network through monitoring platforms that leverage open-source initiatives and, whenever possible, standardized 3GPP APIs. The collected data can be visualized using existing or custom-built tools to support human-driven data analysis.

All monitoring data can be stored within a Network Digital Twin (DT), creating a real-time digital representation of the network state. Both historical and real-time data inside the DT can be accessed through open APIs. On one side, these data can feed advanced analytics modules powered by AI/ML for anomaly detection, trend prediction, and SLA compliance verification. On the other side, applications and vertical use cases increasingly require access to such data in order to optimize their own behaviors and performance. While the DT is an important component of this architecture, it is not a mandatory element, as applications can also access data directly from the monitoring platforms. The DT, however, provides several benefits, enabling comprehensive simulations, enhancing cybersecurity, and supporting more standardized interfaces, among others.

Overall, this design enables operators and third-party applications to access consistent, real-time network intelligence, while also supporting closed-loop automation mechanisms to optimize QoS across diverse vertical domains.

High-level design for Identification and Selection of Backhauling: As shown in the Figure 2-6, this specific sub-enabler enhances the availability and reliability of B5G (private) networks by surveying the available connectivity options for backhauling where and when an ideal fixed backhauling may not be available (e.g. deploying a temporary network in case of a disaster). As shown in the figure, this is performed by examining the available solutions against the deployed network requirements in terms of bandwidth, latency, reliability and cost. The most efficient backhaul solution will be selected and used.

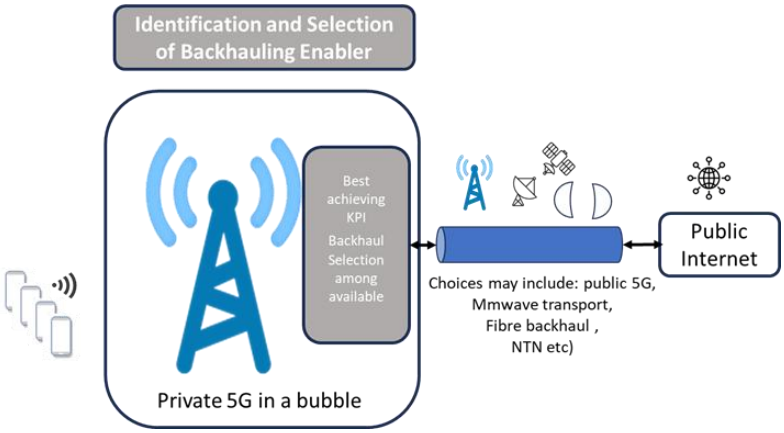

Figure 2-6 Identification and selection of backhauling

In the rest of this subsection, we describe use case specific design, development and implementation for the enabler.

Design, development and implementation for the P1 use case: Figure 2-7 illustrates the interconnectivity for the P1 use case with emphasis on communication resource management. More specifically,

- Automated gNB configuration: Allowing automated configuration and deployment of base stations which meets the requirements of the use case.
- Network performance monitoring and control: Almost real time monitoring of the deployed network. By identifying possible issues modifications can be applied to the network to have more efficient use of resources providing the first responders the best possible coverage.
- Identification and selection of backhauling: If available automated selection of backhauling among existing candidates ensuring connectivity with the public Internet.

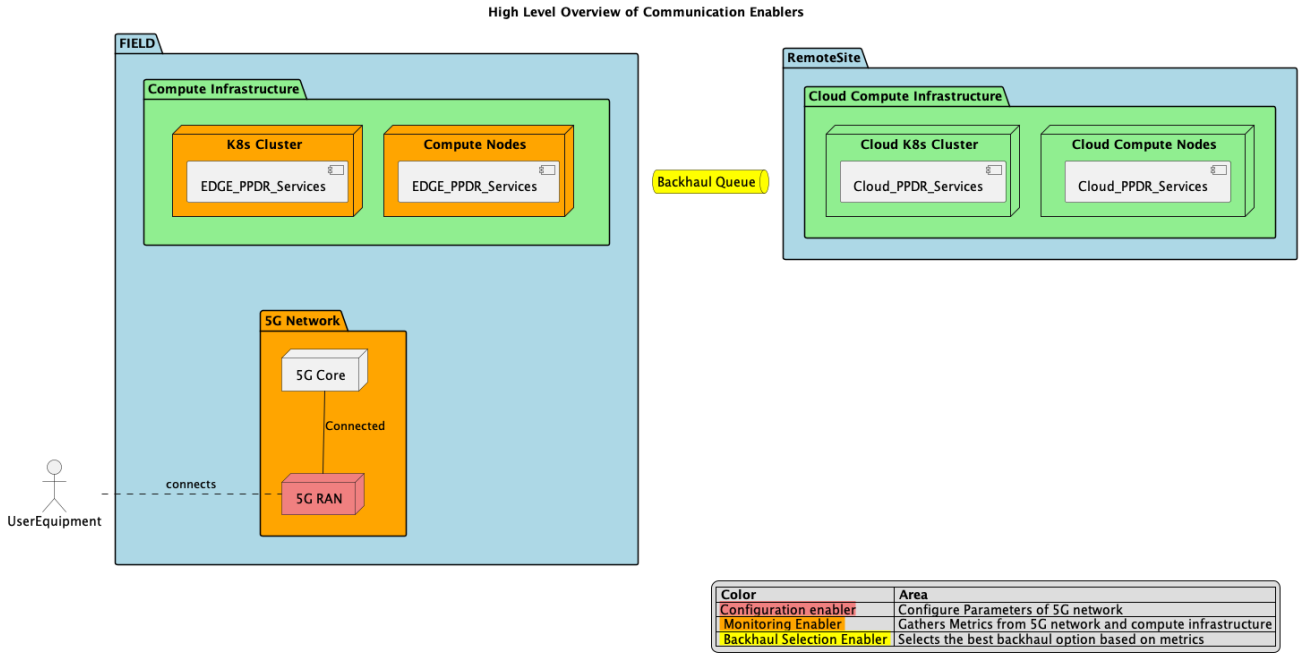

Figure 2-7 P1 related communication resource management sub-enablers

When a service request is received, the system will deploy a K8S cluster (as described for the corresponding sub-enabler) and a core network will be deployed and configured. Afterwards the gNB configuration process begins as shown in Figure 2-8. The system will communicate with the Radio

Resource Management System (RRMS) through standard TMF APIs. The RRMS will communicate with corresponding gNB agents which are pre-deployed in the gNBs allowing for configuration and control by setting various operational parameters and defining the core network where each gNB is to be connected. The various interactions among the components can be seen in the figure. When creating a network for the first time, the system will set the various parameters of the gNB configuration, like frequency, slices, PLMN, TDD configuration or Tracking Area Code (TAC), while at the same time instructing the gNB where to find a core network to be connected to. Once this is done, the gNB will start operating and connect to the desired network. If needed, during the life-cycle of the gNB the system can start, stop and restart the gNB as needed depending on the requirements and responding to any situational changes. Furthermore, configuration changes can also be applied, which can vary from changing simple parameters to directing the gNB to connect to another core network if needed.

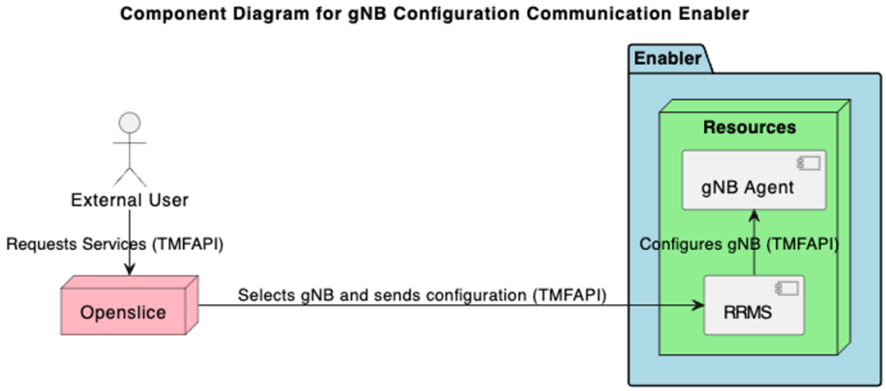

Figure 2-8 Automated gNodeB configuration sub-enabler for the P1 use case

Regarding the sub-enabler "Network performance monitoring and control", the Patras 5G testbed (https://wiki.patras5g.eu/) will provide continuous monitoring and observability of the whole infrastructure, including compute resources, network resources and deployed 5G networks. The monitoring architecture overview is described in Figure 2-9. Interested parties can view various dashboards of the metrics gathered. Operating under the Role Based Access Architecture (RBAC), different verticals can have access to data that correspond to their experiments. The provided solution is Grafana, an open source widely recognized and accepted for information visualization. For data storage a time series database, Prometheus, is used, which is a standard approach for keeping such measurements data. To gather the metrics, Prometheus probes and gets info from various Prometheus exporters installed in the Patras 5G infrastructure. These can include built-in solutions, like K8s clusters or various network applications, monitoring tools that can export in certain formats (like NETDATA or custom-made ones). Furthermore, raw Prometheus data can also be provided if required. In this way other entities can use APIs to get specific metrics from the Patras 5G infrastructure in almost real time.

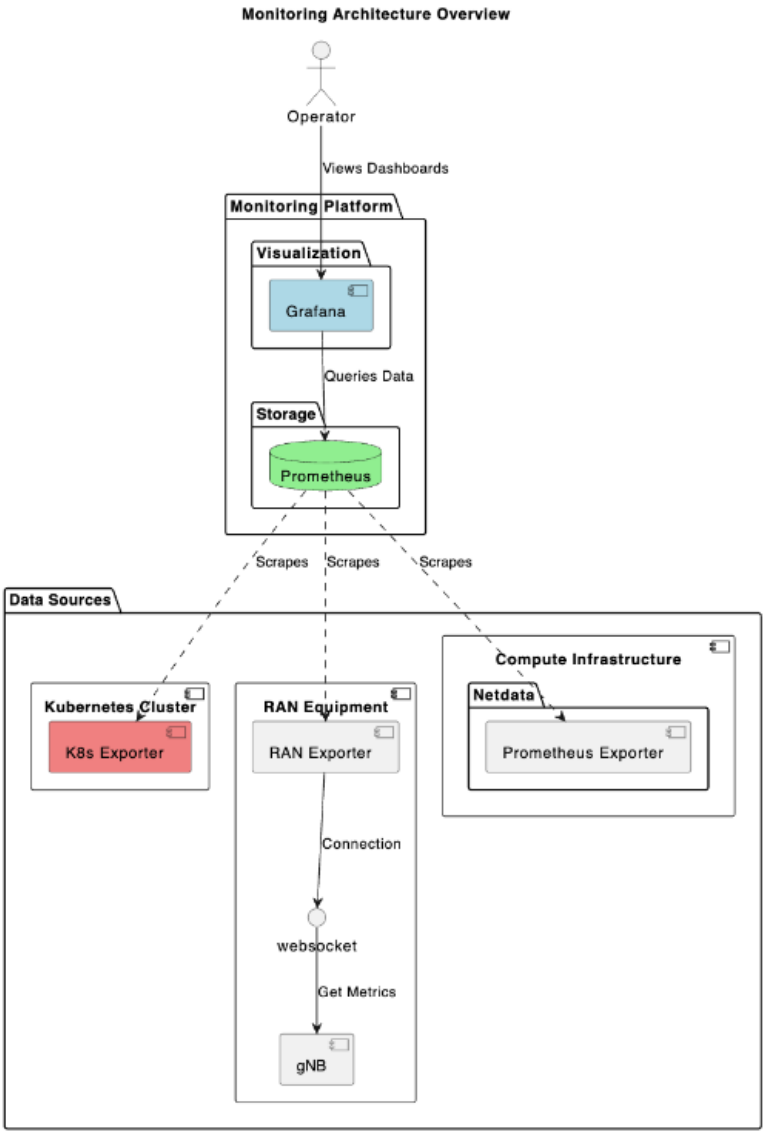

Figure 2-9 Network performance monitoring and control for the P1 use case

The sub-enabler "Identification and selection of backhauling" serves P1 in the case when getting access to a Remote Control Center and/or other services from the public Internet. Specifically, in a PPDR campaign like in P1, it would be highly beneficial for the campaign efforts when no backhaul connectivity is guaranteed. For such a purpose, the sub-enabler will be developed to scan all the backhaul connectivity options and select the most appropriate one based on PPDR QoS requirements. Such options might include a Non-Terrestrial Networks (NTN), a pre-existing fiber infrastructure or an existing public 5G/6G network that may be operational (after a public disaster). End-to end (E2E) slice delivery will be provided, deployed over multiple private and public B5G/6G networks, and extended from the (remote) emergency site to the Central Control Centre.

The first implementation results will be documented and presented in Deliverable D4.1, in association with the use case, scheduled for submission at M17.

Design, development and implementation for the P2 and P4 use cases: The P2 use case aims to demonstrate and investigate the B5G mission critical services interoperability with other non-3GPP systems across national networks and has strong synergies in the implementation with the P4 use case. The monitoring system for network performance and control is essential part of the use cases in order to provide best available QoS when network slicing, multiple connections and/or networks are available in

order to illustrate the benefits of network selection as well as QoS statistics. Illustration of the implementation architecture is presented in Figure 2-10.

The implemented system will be running as a part of the Finnish B5G test network in Oulu in which several measurement nodes will be placed to monitor essential parts of the network. In the mobile network context these nodes are usually the clients (UE) and edge servers. Other options, such as intermediate routers as nodes in core network or BSs, can be also used. GPS- or PTP-based time synchronization will be used for accurate time synchronization between the nodes, which is essential especially for latency measurements. Furthermore, the components in the test network is currently harvested with accurate energy measurement equipment, which can be used for real-time power consumption measurements. The main measured KPIs across Finnish use cases include throughput (performance), latency (operability & performance), packet loss (reliability) and optionally power consumption (operability & sustainability). Qosium software and Carlo Gavazzi devices are the current options to be used in the implementation and Grafana for visualization. Control mechanism of the network(s) can be done by scripting according to the measurement information.

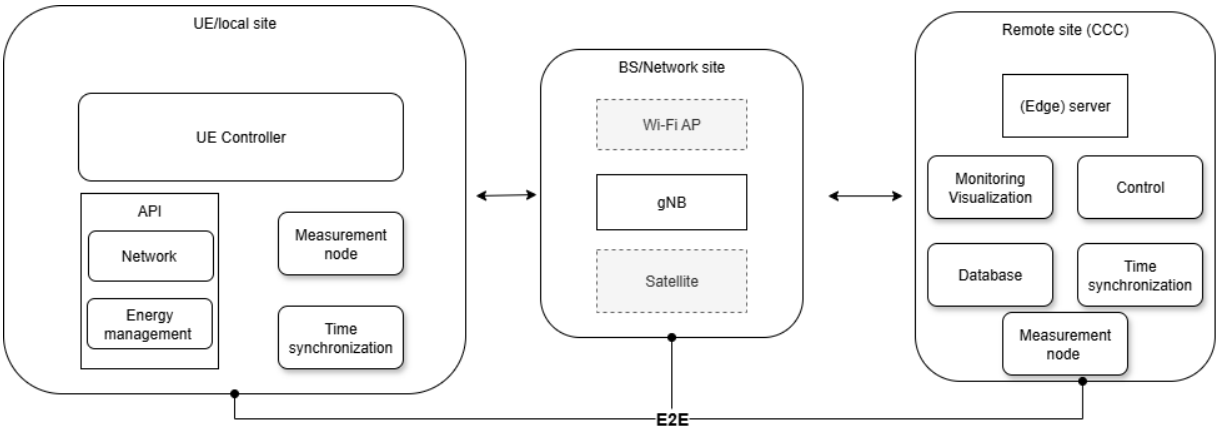

Figure 2-10 QoS assessment architecture for P2 and P4 use cases

The P4 use case aims to demonstrate arctic area search and rescue operation in northern Finland. The monitoring system for network performance and control is essential part of the use case in order to provide best available QoS even in low mobile coverage areas. The implementation of network performance monitoring and control will follow the outline mentioned above. As the first responder in the field is driving with an eSled, all the required monitoring HW and SW are onboarded with the vehicle and powered by the eSled or an external battery bank. The outdoor testing environment requires time synchronization using GPS or AccuBeat atomic clock. The implementation is roughly divided into two phases similarly as for the P2 use case.

The implementation of the monitoring system is carried out in two phases and reported accordingly to the two deliverable deadlines of WP4. The first phase, reported in D4.1, included the deployment and integration of the measurement nodes with the system as well as preliminary testing in the Finnish B5G test network. The second phase, reported in D3.2 and D4.2, includes complete integration and testing with the actual use case setup and final components. As the first pre-trial will take place during the winter of 2026, the first version of the measurement system should be available by then.

Design, development and implementation for the E1 use case: For the E1 use case, the ZSM framework is designed to operate as the central intelligence for intent-based orchestration of energy services. Figure 2-11 shows the high-level architecture, where the ZSM receives intents from operators, service platforms, or automated applications and translates them into network and service configurations.

The process begins with intent acquisition, including interfaces for intent expression, pre-processing, and monitoring. Intents are then processed by the understanding and classification module, which

interprets and maps them to the energy management domain. The validation module ensures that intents are consistent with policies, capabilities, and service constraints.

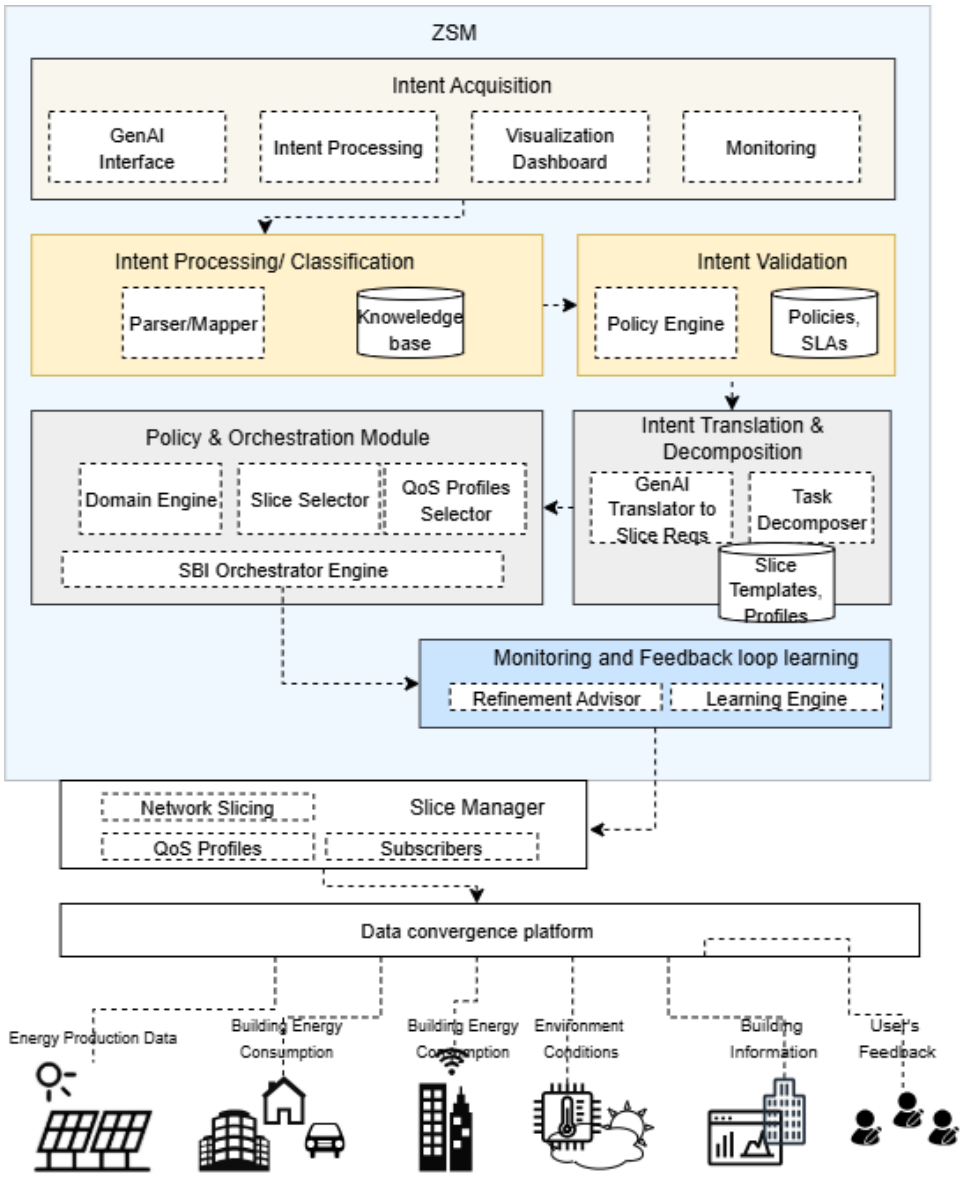

Figure 2-11 Service and Resource Orchestration solution for E1

After validation, intents are passed to the translation and decomposition module, where they are converted into service templates and decomposed into concrete tasks. Importantly, intents are also leveraged to select the most suitable network slice according to IoT traffic requirements and the prevailing network conditions. This allows the framework to dynamically apply differentiated QoS profiles for subscribers, ensuring tailored service delivery.

The policy and orchestration module then compiles these tasks and interacts with the orchestration layer to apply the necessary configurations. Through the slice manager, it not only manages subscribers and QoS but also triggers specific operations to configure network slicing in alignment with generated goals. Furthermore, the framework will support integration with multiple orchestrators, enabling cross-domain adaptability.

Finally, the data convergent platform provides the telemetry required for monitoring and closed-loop assurance. This monitoring data helps operators understand current network metrics, identify areas for

improvement, and feed insights back into ZSM's learning mechanisms, ensuring continuous refinement of policies, templates, and orchestration strategies over time

The first implementation results will be documented and presented in Deliverable D5.1, associated with the use case, scheduled for submission at M17.

Design, development and implementation for the E3 use case: The E3 use case is based on a modular and scalable architecture, as presented in Figure 2-12, that supports the intelligent monitoring, control, and optimization of solar energy systems using B5G/6G connectivity and distributed computing resources. The generalized solution is composed of three main layers: (1) field-level edge devices with 5G RedCap connectivity, responsible for real-time telemetry collection and local actuation; (2) a B5G/6G communication layer that ensures ultra-reliable, low-latency, and secure data transport; and (3) a cloud-based orchestration platform that hosts forecasting models and enables system-wide control and optimization. The design ensures that solar power plants operate effectively even under fluctuating network conditions, with edge-based fallback mechanisms and secure data pipelines. Central to the system is the integration of AI/ML algorithms for predictive energy forecasting and inverter control, alongside network slicing and dynamic QoS management for mission-critical commands. Figure 2-13 shows the Cloud-based Orchestration Platform for the E3 Use Case.

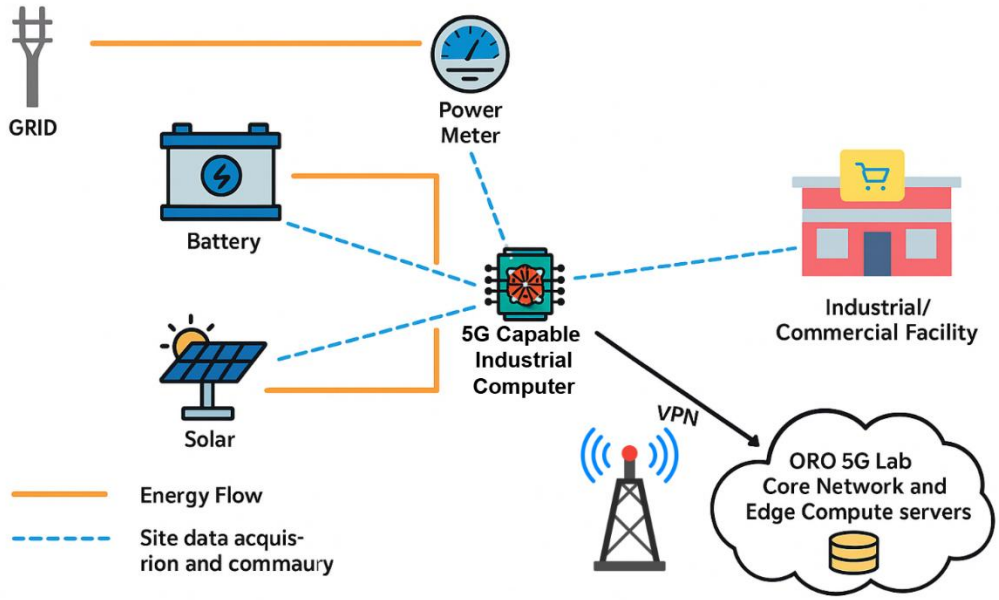

Figure 2-12 Modular and scalable architecture for E3 design

The development and implementation of the E3 solution will proceed in two phases. In Phase 1 (by M18), the focus will be on building a preliminary prototype, including the deployment of edge devices at selected test sites, initial 5G RedCap integration, and setup of a minimum viable cloud orchestration platform. Basic data flows, device discovery, telemetry aggregation, and rule-based edge control will be validated in a controlled environment. In Phase 2 (final implementation by M33), the system will be scaled and finalized with full support for AI-based forecasting, remote actuation from national energy authorities, performance monitoring, and integration with WP3 AIaaS components. This phase will also include trials under real-world operational conditions, validating end-to-end functionality, network slicing, high availability, and predictive optimization workflows.

The first implementation results will be documented and presented in Deliverable D5.1, scheduled for submission at M17.

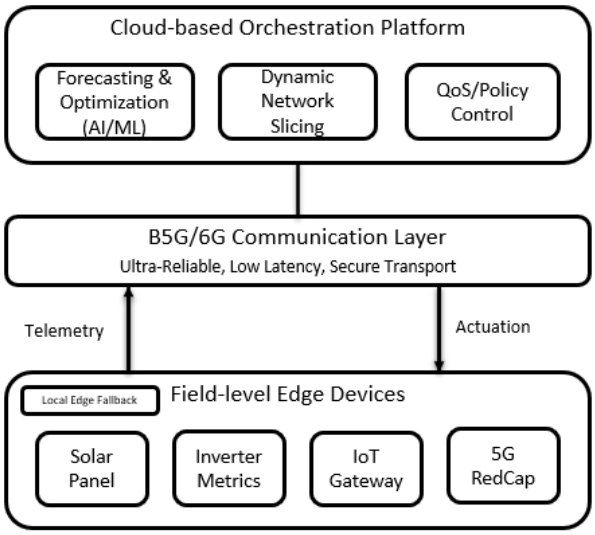

Figure 2-13 Cloud-based Orchestration Platform for the E3 Use Case

Design, development and implementation for the T1 and T2 use cases: The T1 and T2 use cases will implement a monitoring system on a private network that allows data to be collected in a scalable way and make it available through a Network Digital Twin. The baseline will be the private network built for the ENVELOPE project (https://envelope-project.eu/) that will be fully operational by the end of 2025. The monitoring part will be ready for phase 1 (M18) including a first version of the Network Digital Twin. In M33 a fully operational version including APIs for the interaction with the orchestration framework and applications will be ready. In addition, T1 and T2 will benefit from the integration into the end-to-end system of the metric gathering tools offered by the HPE ANW P5G core network. Metrics from the core network can be retrieved and fetched to the Monitoring Platform and the Network Digital Twin either through the integrated Prometheus query system or directly via REST APIs. These metrics fall into two main categories: general system-level metrics and core-specific metrics. System-level metrics provide insights into the health and performance of the physical or virtual nodes hosting the core components, including CPU usage, memory consumption, disk activity, I/O operations, and internal process statuses. Core-specific metrics, on the other hand, are related to the functionalities of the Core itself and are associated with various Network Functions (NFs) such as AMF, AUSF, NRF, PCF, SMF, UDR, and UPF. These metrics offer detailed visibility of signaling activity, session management, authentication workflows, and user data handling.

In the current phase of development, the focus is on enabling efficient metric collection from the HPE ANW P5G core network, integrating the core network's monitoring tools with the Monitoring Platform and the Network Digital Twin. Namely, HPE's core solution exposes a set of metrics via Prometheus, accessible either through predefined queries or directly via REST APIs. To integrate these metrics into the monitoring platform, a custom library must be developed. This library will be responsible for periodically querying the available endpoints, adapting the frequency and scope of requests based on the specific needs of the monitoring system.

Design, development and implementation for the T4 use case: Designing and deploying 5G SA networks around network programmability means treating every layer i.e., RAN, TN, and 5GC, as a set of software-driven, dynamically configurable systems rather than static appliances. The network architecture must expose programmable control points and standard interfaces so that orchestration platforms can manipulate resources, instantiate NFs, and steer traffic flows according to service requirements.

In Figure 2-14 we show the design of a 5G SA network deployed with network programmability. In the RAN, communication between components in an O-RAN deployment is organized around open and standardized interfaces that enable interoperability, fine-grained control, and programmability. The

architecture disaggregates the RAN into RU, DU, and CU, each of which can be supplied by different vendors or deployed in different locations, provided they comply with interface specifications. Through the F1-C interface, the DU and CU coordinate UE context, bearer setup, and configuration updates, while F1-U carries user traffic toward the core. Beyond the RU/DU/CU split, O-RAN introduces the near-RT RIC, which supervises functions requiring control-loop latencies of 10 ms to 1 s, such as handover optimization, interference management, and slice-aware scheduling. It communicates with DUs and CUs through the E2 interface, exposing telemetry and control points in a vendor-neutral format. xApps running on the near-RT RIC subscribe to E2 metrics and issue policies or commands to adjust schedulers, power control, or beam management for specific slices or services. By relying on interfaces such as OFH, F1, E1, E2, A1, and O1, the O-RAN architecture enables a fully programmable RAN. Each function can be deployed, scaled, or optimized independently, while RICs and xApps orchestrate radio resources and policies in alignment with slice requirements, traffic profiles, and SLAs.

In the 5G Core, communication between NFs is governed by the SBA, which defines how control and user-plane components exchange information through well-specified interfaces. Each NF is implemented as a virtualized or containerized software element that exposes RESTful APIs over the SBI, enabling other functions to discover, authenticate, and invoke its services dynamically. A central component of the SBA is the NRF, which maintains a registry of available NFs, their supported services, and reachable endpoints. When an NF (for example, the AMF) needs to interact with another (such as the SMF or NSSF), it queries the NRF to obtain the target's profile and then establishes a secure HTTP/2 session, typically over TLS, to invoke the required operation through its SBI endpoint.

The AMF, as a control-plane NF, relies on multiple interfaces to carry out its tasks. Toward the RAN, the AMF communicates over the N2 interface using the Next-Generation Application Protocol (NGAP) to manage registration, authentication, mobility, and context transfer for UE. When a session needs to be created or modified, the AMF interacts with the SMF via the N11 interface, passing information about the UE, the requested data network, and session parameters. The SMF, in turn, coordinates with the UPF over the N4 interface, delivering rules for packet forwarding, QoS enforcement, and traffic steering in the user plane. This separation ensures that data-plane elements remain simple and optimized for high-throughput processing, while policy and session logic remain centralized in the control plane.

The NSSF is reached through the SBI by the AMF or SMF whenever slice selection or admission is required. Over the N22 reference point, the NSSF provides information about which slice instances are available for a given UE or service request. This allows the AMF to bind the UE to the proper slice and instruct the SMF to allocate corresponding session resources.

On the user-plane side, the UPF is the anchor for data flows. It connects upstream to the SMF via N4, downstream to the RAN through the N3 interface, and outward to external data networks via N6. The N3 interface transports encapsulated user packets between the gNodeB and the UPF with minimal overhead, while N6 provides standard IP connectivity toward the internet or enterprise networks. When low latency is required, the UPF may be instantiated at the edge, and its location is determined during session establishment based on policies delivered by the SMF over N4.

This set of interfaces i.e., SBI and N2, N3, N4, N6, N11, N22, creates a flexible topology where each NF can be deployed, scaled, or relocated independently without disrupting the rest of the system. This modular and interface-driven design is at the heart of implementing a programmable 5G Core capable of supporting diverse services while maintaining strict performance and isolation requirements.

Furthermore, the transport network (TN) binds these domains together. SDN-enabled routers and switches establish deterministic paths between RAN and core or edge locations, reacting in real time to congestion or link failures. Controllers can adjust queueing, re-route sessions, or reserve bandwidth for mission-critical services. This dynamic approach eliminates the rigidity of fixed hardware paths and allows new slices or services to be provisioned without re-engineering the transport fabric.

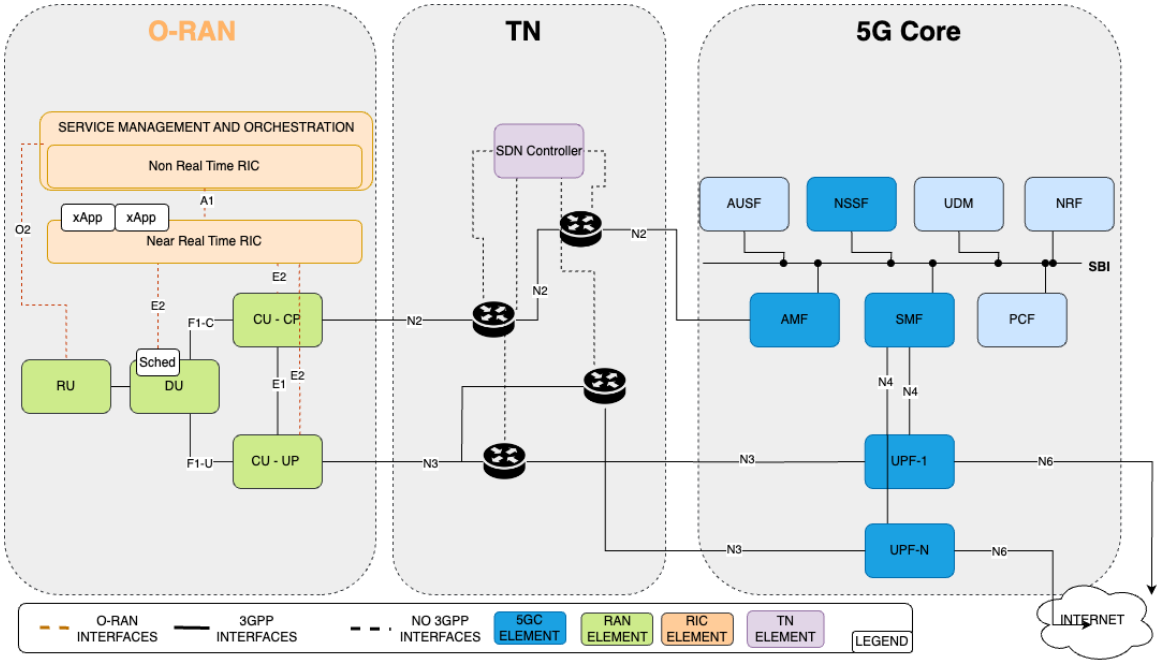

Figure 2-14 Network Programmability to deploy 5G SA Networks

Apart from the network programmability design principles, which are embodied in the network and service design in the case of network deployment needed for T4, ZSM will also be leveraged to improve resource and service orchestration, which are essential for meeting T4 service requirements. For instance, maintaining end-to-end (E2E) latency below 30 ms is essential for vehicular services such as teleoperation, where, for instance, remote control commands need to be transferred from the control centre to the onboard equipment on the teleoperated vehicle. The ZSM framework addresses these requirements by providing continuous monitoring and coordinated management of multiple Network Functions Virtualization Infrastructure (NFVI) domains within a unified environment. By integrating orchestration modules, monitoring tools, and NFVI domains under a single control system, ZSM ensures consistent service quality even under challenging conditions. The ZSM framework's flexibility was already demonstrated in the previous projects such as SNS JU TrialsNet, where IMEC and NXW managed diverse NFVI deployments in Belgium and Italy, and further validated during the IMEC-ORO trials in Romania, maintaining stable performance across varying load scenarios.

In the experiment performed on the 5G SA infrastructure in Romania, hosted services were subjected to a number of concurrent users exceeding the use case limits to evaluate the resilience of the ZSM framework under stress. Latency at node ztm2-vm rose from 17.468 ms to 497.045 ms, but once the orchestrator initiated a redeployment to a higher-performing K8S POD, it quickly dropped to 18.878 ms as seen in Figure 2-15. These results demonstrate the framework's effectiveness in maintaining stable QoS even during high-load conditions. Figure 2-16 illustrates the progression of this behavior, showing E2E latency rising under stress at node ztm2-vm until the orchestrator migrated the service to node ml-vm, which had greater resource availability, restoring the expected QoS. The traces capture the orchestrator's decision records, detailing both the KPI values of the selected node and the corresponding actions taken to preserve service quality whenever latency exceeded the 30 ms tolerance threshold for this service. In the scope of AMAZING-6G, and in particular T4 use case, this ZSM framework will be extended with more advanced decision-making logic, employing various machine learning algorithms to improve the effectiveness of the rule-based one that was used as a baseline.

For the T4 use case, a first version of both Network programmability and ZSM enablers is under development. Initial testing is already taking place in IMEC's testbed environment (as shown in Figure 2-15 and Figure 2-16), and will continue during the first year. The enablers will be integrated in the TUC

testbed during the second year of the project, along with other preliminary compatibility and integration tests. Full end-to-end trials will take place in the final year of the project.

id integer	node_id integer	node_name character varying	latency double precision	timestamp double precision
515136	224	ztm2-vm	17.468	1756286593.0232673
515137	224	ztm2-vm	17.468	1756286593.0232673
515138	224	ztm2-vm	17.468	1756286593.0232673
515139	224	ztm2-vm	497.045	1756286594.032876
515140	224	ztm2-vm	497.045	1756286594.1478348
515141	224	ztm2-vm	497.045	1756286594.1478348
515142	224	ztm2-vm	497.045	1756286594.1478348
515143	224	ztm2-vm	497.045	1756286595.2655272
515144	224	ztm2-vm	497.045	1756286595.2655272
515145	224	ztm2-vm	497.045	1756286595.2655272
515146	224	ztm2-vm	497.045	1756286596.36779
515147	224	ztm2-vm	497.045	1756286596.36779
515148	224	ztm2-vm	497.045	1756286596.36779
515149	224	ztm2-vm	497.045	1756286597.4919448
515150	224	ztm2-vm	497.045	1756286597.4919448
515151	224	ztm2-vm	497.045	1756286597.4919448
515152	224	ztm2-vm	497.045	1756286598.604006
515153	224	ztm2-vm	497.045	1756286598.604006
515154	224	ztm2-vm	497.045	1756286598.604006
515155	223	ml-vm	18.878	1756286599.170688
515156	223	ml-vm	18.878	1756286599.170688
515157	223	ml-vm	18.878	1756286599.170688
515158	223	ml-vm	18.878	1756286600.190877
515159	223	ml-vm	14.201	1756286600.7606382

E2E latency increases in node ztm2-vm

Switch to node ml-vm and E2E latency decreases

Figure 2-15 Extract of ZSM orchestrator decision traces showing service migration actions triggered by E2E latency variations

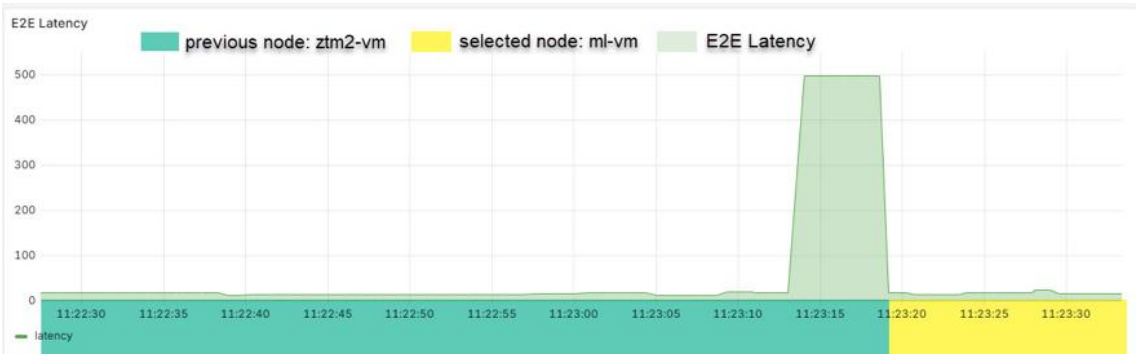

Figure 2-16 E2E latency variations and the ZSM orchestrator decisions showing node selection changes

Design, development and implementation for the T5 use case: The **ZSM** framework acts as the central intelligence for the entire private 5G network, automating complex tasks and ensuring the network meets the stringent Service Level Agreements (SLAs) for the STS crane teleoperation.

- Intent-Driven Orchestration Layer: The highest layer of the ZSM design will be intent-driven. The crane operator or a port management system will define a high-level intent, such as "Ensure real-time teleoperation for STS crane #3." This intent will be translated into specific network requirements, like "latency < 5ms," "jitter < 1ms," and "throughput > 100 Mbps" for the video and possible haptic feedback streams.
- Automation Engine: This is the core of the ZSM system that consists of:
    - Analysis: Based on the information gathered by various metrics, this will be input adjacent to the requirements given by the user,

- Decision: Based on the requirement given by the user, a decision-making component determines the necessary action. This could be a simple configuration change, a more complex resource allocation, or a trigger for a backhaul link switch.
- Execution: The system automatically executes the decided action without human intervention. This could involve re-routing traffic, adjusting radio power, changing QAM pattern, or activating a redundant backhaul link.

Network performance monitoring and control: This sub-enabler provides the real-time visibility and granular control required to meet the strict performance requirements of the crane's teleoperation. It serves as the "eyes and hands" for the ZSM closed-loop automation.

- Real-Time Metric Collection: The system will collect data from various sources:
- On-Crane Sensors: Telemetry data from the crane itself, including its position, speed, and any haptic feedback information.
- 5G Network Probes: Embedded probes and software agents within the 5G gNodeB and core will measure key performance indicators (KPIs) like end-to-end latency, jitter, packet loss, and throughput for specific traffic flows (e.g., the crane control signal).
- Predictive Analytics: Beyond simple monitoring, the system will use historical and real-time data to predict network issues. For instance, based on weather patterns, cargo traffic, or the time of day, it can anticipate network congestion and proactively adjust resources.
- Control and Remediation Mechanisms: This component will work in tandem with the ZSM execution layer to implement changes.
- Dynamic QoS Adjustment: The system can prioritize traffic based on its type. Crane control signals will receive the highest priority, followed by haptic feedback and then high-resolution video streams.
- Anomaly Detection and Alerting: Any deviation from the defined SLAs will trigger an immediate alarm, which the ZSM system can use to initiate an automated remediation action.

Identification and selection of backhauling: The backhaul links, which connect the private 5G network to the wider internet or a port's central data center, are a critical single point of failure. This sub-enabler ensures a resilient backhaul.

- Multi-Link Redundancy: The design will incorporate at least two different backhaul links to the core network. One can be the primary link (e.g., a high-capacity fiber optic or copper connection), while the other serves as a redundant link (e.g., a cellular link).
- Intelligent Link Selection: A dedicated component analyzes the link status and makes real-time decisions on which backhaul to use.
- Automated Failover: In the event of a primary backhaul link failure, the system will automatically and seamlessly switch all critical traffic to the redundant link without any human intervention. This failover must be fast enough to not impact the crane's operation.

Also, the core is responsible for the overall management and control of the private 5G network. It handles critical functions such as user authentication, session management, and ensuring that data traffic is routed correctly. By having the 5G core on-premise, the system minimizes latency and operates independently of public networks, which is essential for the secure, real-time applications required by this use case. The Acromove P5G app further contributes to this role by providing a centralized interface to manage and restart components. This capability allows for basic control and troubleshooting, which is essential for maintaining network stability and performance.

For the T5 use case, a first version of the enabler is under development. Initial testing will take place in lab environment in the 1st year. After this period, deployments in the field will follow in the 2nd year for conducting some preliminary experiments. In the 3rd year, final trials will take place.

2.2 Network slicing

2.2.1 Description of the enabler

Network slicing in cellular systems is based on the concept of running multiple logical networks on a shared physical infrastructure. Each slice represents a virtualized, end-to-end network instance tailored to a specific service profile or customer requirement—for example, enhanced mobile broadband, massive IoT, or URLLC communications. Slices are isolated from one another in terms of control, data handling, and resource allocation, which allows operators to guarantee differentiated SLAs on the same RAN, Core and Transport infrastructure. The orchestration layer ensures lifecycle management of these slices, from creation to scaling and termination, while enforcing performance and security boundaries between them.

The core network, in addition to playing a central role in managing UE access to slices, plays an important role in creation of E2E slices. In the core network, slicing manifests through the allocation of dedicated or shared Network Functions (NFs). For example, one slice may deploy its own SMF (Session Management Function) and UPF (User Plane Function) to achieve traffic isolation, while another slice shares with other slices the AMF (Access and Mobility Management Function) for efficiency. Network exposure functions (NEF) and policy control (PCF) can be tailored per slice to enforce differentiated policies, QoS profiles, and charging schemes. This approach guarantees that each slice has its own logical control over mobility, session management, and service policies—preventing cross-slice interference and enabling customized SLAs.

Transport slicing ensures that the backhaul links between RAN and core network domains are logically partitioned. For this purpose slice-specific forwarding paths can be created with guaranteed bandwidth, latency, and jitter characteristics. Slice-aware transport mechanisms ensure that the end-to-end SLA is upheld, regardless of congestion in other parts of the network. Isolation in transport not only secures performance guarantees but also prevents slice degradation caused by traffic bursts in other slices.

Within the RAN domain, slicing extends beyond logical partitioning to include dedicated Network Functions. For example, a slice may be assigned its own CU-UP (Centralized Unit - User Plane) instance to isolate user-plane data processing, while sharing control-plane components such as CU-CP and DU with other slices. This enables fine-grained separation of traffic flows and optimizes resources for latency-sensitive or high-throughput services. Dedicated CU-UP instances also simplify monitoring, troubleshooting, and SLA enforcement, as performance metrics can be tied to a specific slice rather than aggregated across the network. A fundamental mechanism of RAN slicing is radio resource partitioning. At the scheduler level, slices can be allocated dedicated or dynamic Physical Resource Block (PRB) quotas. This ensures that each slice receives a guaranteed share of spectrum resources under high load conditions while still allowing statistical multiplexing when capacity is available. By configuring PRB resources quota per slice, operators can enforce strict slice guarantees per slice. On the user equipment (UE) side, slicing enables devices to attach simultaneously to multiple slices or to be restricted to a specific slice depending on subscription, application requirements, or enterprise policy. Slice selection is driven by parameters such as the Single Network Slice Selection Assistance Information (S-NSSAI), which guides the UE to establish connectivity with the appropriate slice. A single device can thus run consumer services on an eMBB slice while enterprise or mission-critical applications use a URLLC slice. This ensures that application-specific performance requirements are met without over-provisioning resources across all services.

When combined, slicing across the RAN, core network, transport network, and UE domains creates a consistent end-to-end logical separation. This guarantees that a slice behaves like a fully dedicated network, even though it shares the same physical infrastructure with others. The orchestration and management layer ensures coordination across these domains, aligning resources, policies, and monitoring tools to deliver differentiated services reliably and securely.

2.2.2 Use case association and contributing partners

The following table presents the mapping between network slicing and the use cases.

Table 2-3 Mapping between network slicing and the use cases

	Network slicing
H1, H2	TNO
P3	VTT
E1	CAPG, ORO
E2	TNO
E3	CAPG, ORO
T1, T2	LINKS, HPE
T3	TUC
T4	IMEC, ISRD, TUC

In the **H1 and H2 use cases**, medical devices such as connected patches and pacemakers rely on continuous and secure connectivity to transmit patient data to hospital data centers. Network slicing enables the creation of isolated, high-priority slices that guarantee the required QoS for these health-critical applications. This ensures low latency, high reliability, and strong security boundaries, which are essential for timely detection of anomalies, remote diagnostics, and rapid intervention in case of emergencies. By isolating medical traffic from other network services, slicing also mitigates risks of congestion and cyberattacks, thereby enhancing patient safety and data integrity.

In the **P3 use case**, network slicing is part of the emergency private B5G/6G communication on-the-move for guaranteeing service(s) QoS in the search and rescue area. It facilitates the protection of the critical data flows such as audio-video communication between the first responders or lidar data providing the forms of terrain, which are vital for secure and efficient rescue operations.

For the **E1 use case**, which involves managing multiple sensors within Renewable Energy Communities (RECs), network slicing ensures that critical data — such as energy telemetry, control commands, and user feedback — is transmitted with guaranteed latency and reliability. By isolating traffic related to energy control and building automation from general-purpose data, the system avoids congestion and secures QoS for sensitive operations, such as real-time HVAC adjustments or coordinated energy distribution. In E1 the computing continuum involves the Smart Buildings IoT platforms, devices and extreme edge nodes as well as edge platforms deployed in the smart buildings, up to cloud resources used to run applications at the REC level.

In the **E2 use case**, drones are deployed to perform high-precision wind blade inspections, requiring reliable connectivity, low latency, and guaranteed throughput for real-time data transmission and control. Network slicing enables the creation of a dedicated slice tailored to the specific QoS requirements of the drone operation, ensuring service continuity and performance even in dynamic or congested network environments. This isolation is critical to avoid interference from other concurrent services and to maintain the integrity of mission-critical inspection tasks. Furthermore, slicing supports

mobility management and edge integration, which are essential for seamless drone operation across large wind farms.

In the **E3 use case**, dedicated network slices enable uninterrupted, low-latency transmission of telemetry and control signals between solar inverters, edge devices, and the cloud platform. This is especially crucial when responding to real-time grid operator commands or performing edge-based actuation with sub-second reaction times. By allocating a separate slice for energy operations, E3 ensures that performance remains unaffected even during high network load or degraded conditions.

In the **T1 use case**, the application for the protection of vulnerable road users runs over distributed edge and extreme edge nodes, in particular using RSUs with computing capabilities and equipped with batteries and solar panels. For this kind of scenario, there are needs of two different types of slices:

- URLLC: this type of slice should provide low latency communication to manage all the messages related to the sensing of the environment and the subsequent warnings to visually impaired users.
- eMBB: this type of slice is activated on demand when the AI-based sensing application running on the RSU is moved to the edge. This implies that the RSU must send the streaming of its sensors (e.g., camera and LiDAR) to the edge requiring the use of a large amount of bandwidth. This is automatically triggered when the network detects that the battery of the RSU is reaching a critical point, and its consumption should be reduced by moving the computationally intensive application to the edge.

In the **T2 use case**, the application for urban video surveillance is distributed among edge nodes and 6G-connected UGV devices equipped with sensors, cameras, and computing resources. The application is split into containers for several tasks, including control of UGV navigation, image recognition, and object detection. Tasks with lower latency constraints can be offloaded to the edge nodes, based on the battery level. The network automatically detects the need to move some computationally intensive applications from the robot to the edge. Depending on the application that should be moved, different slices can be allocated to support low-latency messages (URLLC e.g. for command flow) or sensors flows (eMBB).

Network slicing is also essential for the **T3 use case**, because it guarantees that the communication between trains, trackside infrastructure, and the ISAC microservice operates with ultra-low latency, high reliability, and sufficient throughput. Since obstacle detection and warning require immediate response to ensure safety, a dedicated slice isolates these critical functions from other traffic in the network, preventing congestion or delays. This ensures that sensor data, ISAC analysis results, and driver notifications are consistently delivered within the strict timing requirements of railway safety operations.

In the case of the **T4 use case**, when an autonomous vehicle runs into a scenario which its onboard software cannot address e.g., unmapped route, dense fog, or a confusing traffic signal, it must give control of the vehicle to a remote driver (human operator). To enable this, the network has to be able to deliver: end-to-end control messages with delay less than 20 ms, uplink throughput of at least 25 Mb/s for multiple video streams coming from the vehicle, and a reliable connection that stays alive 99.999 % of the time. On a busy B5G/6G network already carrying smartphone traffic, video streaming, and IoT data, even a short surge of background data can push latency over the limit or drop video frames, undermining safe teleoperation. Network slicing transforms this problem into something that the network can program and control. Furthermore, thanks to network programmability, the SDN (and NFV) controller manages the resources of each slice e.g., radio spectrum, queues in the switch, dedicated UPF, computing resources allocated, based on the type of service and network conditions in real time. Therefore, on a busy B5G/6G network, if the data related to teleoperation goes through a dedicated slice, the network requirements could be guaranteed.

2.2.3 Design, development and implementation

Figure 2-17 illustrates a high-level overall design of network slicing, which shows that slicing may be realized in the different part of the network: UE, radio network, transport and core network. For different AMAZING-6G use cases, slicing will be realized in different ways as described below. Unlike legacy networks that rely on fixed-function hardware and static configurations, 5GSA is designed around

network programmability, enabling network behavior to be modified in real time through software commands without physical intervention. For instance, in dense urban areas, the RAN can allocate more PRBs and adjust beamforming, while the transport network can reroute traffic around congestion to maintain sub-20ms latency. The 5GC can dynamically deploy or move UPF instances closer to the user, reducing delay and packet loss. Network slicing builds upon this programmable network infrastructure by enabling the creation of multiple, isolated, end-to-end virtual networks over the same physical infrastructure. While QoS mechanisms are used within the network to manage and give priority to individual data flows, the slice itself defines a broader context: a network with a certain pool of resources designed to guarantee network requirements for a specific type of service. By combining network programmability with slicing, B5G/6G networks can ensure the SLAs required.

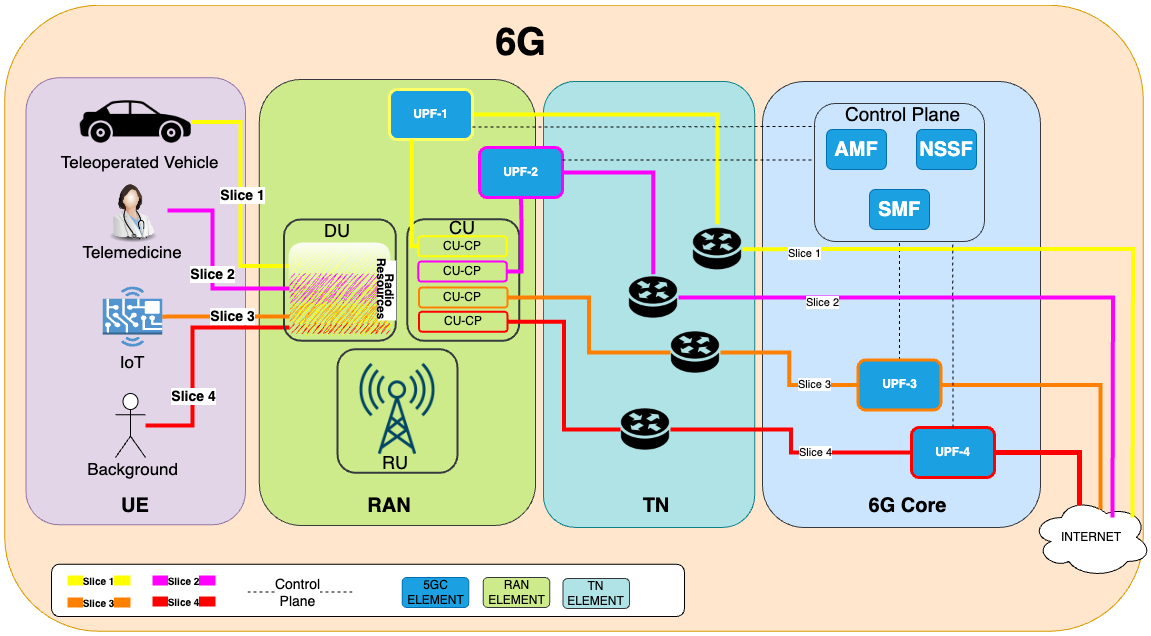

Figure 2-17 Network Slicing in the 6G Architecture

For the **H1 and H2 use cases**, network slicing will be used to support high uplink throughput and low end-to-end latency in order to guarantee the QoS requirements defined for the use cases. A first slice will be defined to handle the traffic between the medical device and the hospital data center, whereas a second slice will be defined for background traffic that is handled by the network. This configuration allows to demonstrate that the required QoS can be guaranteed for the medical-related slice, as well as investigates any potential performance degradation on the background traffic slice due to the finite amount of radio resources. Figure 2-18 shows the high-level design that will be used for the evaluation of network slicing in the H1 and H2 use cases. For the deployment of the radio network, the Amarisoft's Amari Callbox Classic edition will be used, which allows for configuration in a user-friendly manner and can operate as a 3GPP-compliant gNB. For the core network, the Open5GS will be used, which is an open-source implementation of the 5G Core and it is 3GPP Release-17 compliant. The network functions (e.g. AMF, UPF, etc.) will be hosted on different virtual machines to allow for control-user plane separation and can be scaled in terms of memory, processing and storage. Different devices (e.g. smartphones and development boards) are available and can be used as UEs for the evaluations. For the evaluations, the UEs, gNB and core network will be deployed at the TNO location in The Hague.

The results evaluating the performance of network slicing will be documented and presented in Deliverable D3.2.

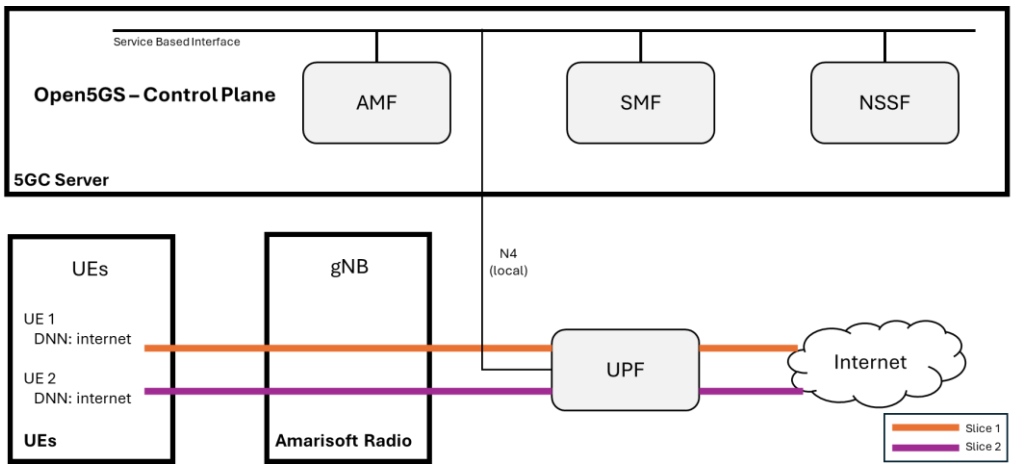

Figure 2-18 Network slicing design for use cases H1 and H2

The network slicing enabler for the **P3 use case** will guarantee the QoS for emergency services in the search and rescue area. The design and implementation in the first step follow a static configuration where the subscriber resources are configured in the Open5GS core and additionally 5G QCI can be allocated to specific applications running in the UE. After gathering the results from the first step and deep diving into techniques allowing traffic prioritization, the possibilities of dynamic application-specific slicing will be investigated in the Finnish B5G test network. The ultimate goal is to form application-specific slice in the search and rescue area, which can be prioritized over other traffic.

The results evaluating the performance of network slicing will be documented and presented in Deliverable D3.2.

In the **E1 use case**, network slicing is designed to support the concurrent operation of heterogeneous IoT data streams, user control commands, and AI-based energy orchestration across multiple sensors. The network slicing enabler will define dedicated logical slices for (i) real-time control and actuation, (ii) aggregated telemetry used for AI inference and optimization at REC level, and (iii) non-critical services such as user interface updates and long-term analytics. This approach will ensure predictable performance for latency-sensitive comfort or energy control decisions, while maintaining scalability and interoperability across multiple independently managed buildings. IoT platforms for smart buildings include sensors like environmental and presence sensors, actuators for HVAC, lighting and blinds control, and smart appliances. Moreover, the buildings are equipped with other elements for energy monitoring and management, like energy meters to measure the power consumption of specific devices or the controllers of renewable energy sources, like solar panels. Extreme edge nodes may include devices with controllable computing capabilities, like IoT gateways, smart cameras, video clients, etc. Moreover, edge nodes like NUCs or local micro datacentres can be available. These elements are managed through Kubernetes or, for smaller devices, K3S platforms. On top of these edge platforms, the Smart Building Resource Orchestrator is in charge of deploying the applications on the various nodes, as previously described in Section 2.1.3.

The development of the slicing enabler for the E1 use case will begin by defining slice templates based on QoS profiles extracted from building-level telemetry and actuation needs. During Phase 1 (by M18), the focus will be on static slice provisioning for critical communication paths between edge devices and cloud-based AI services. The implementation will include slice monitoring agents at building gateways and slice control interfaces at the orchestration layer. In Phase 2 (up to M33), dynamic slice scaling and reallocation based on building usage profiles and occupancy patterns will be added. Real-time KPIs such as latency and throughput per slice will be continuously monitored, and intelligent adjustments will be tested as part of joint trials coordinated under WP5.

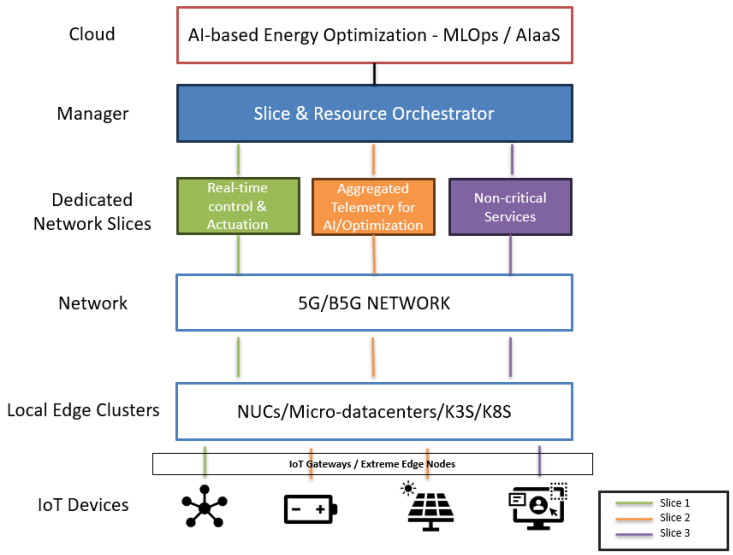

Figure 2-19 Network slicing design for the E1 use case

For the **E2 use case**, network slicing will be used to guarantee the required QoS for the traffic related to the drone operations, which goes over a network that handles other types of traffic too. Therefore, the design of network slicing for the E2 use case will consist of two slices: one for the drone-related traffic and one for other background traffic. This design is similar to the one used for the H1 and H2 use cases (and depicted in Figure 2-18) and allows for similar evaluation scenarios. Moreover, the same hardware will be used as for the H1 and H2 use cases. Differently to the design for the H1 and H2 use cases, where all network components are located at the TNO location in The Hague, The Netherlands, for the E2 use case, some components will be located at the Zephyros in Vlissingen, The Netherlands and the rest at the TNO location in The Hague. In particular, the UEs and the gNB (provided by the Amari Callbox) will be located at the Zephyros lab in Vlissingen and the core network (provided by Open5GS) will be located at the TNO location in The Hague. The connection between the two locations will be provided by a VPN. This design is chosen to more accurately replicate the scenario where the drone and access network are located offshore and the core network is located onshore.

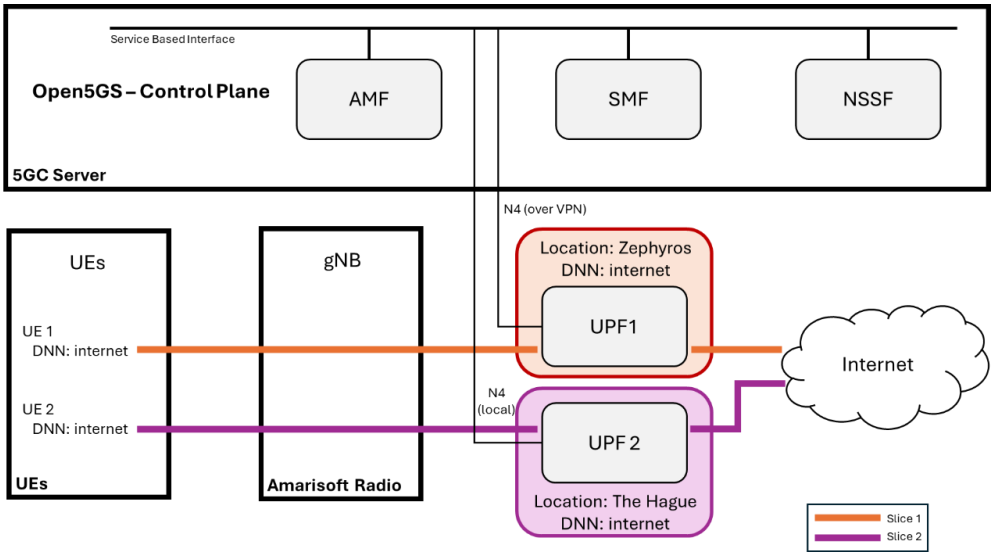

Figure 2-20 Network slicing design for the E2 use case

Additionally, a second design is considered for the E2 use case, as illustrated in Figure 2-20. Similarly to the previously explained design, the UEs and gNB will be located at the Zephyros Lab in Vlissingen and the core network at the TNO location in The Hague. However, to allow for lower latency, a second UPF

will be deployed at the Zephyros lab, specifically to serve the slice handling the drone-related traffic. All other background traffic from the second slice will be routed to the Internet via a different UPF, which will be deployed at the TNO location in The Hague. This design aims to showcase the capability of network slicing in achieving lower latency and replicates the scenario where an edge server (hosting the UPF) is deployed offshore.

The results evaluating the performance of network slicing will be documented and presented in Deliverable D3.2.

For the **E3 use case**, the network slicing enabler is focused on enabling ultra-reliable, low-latency connectivity for distributed solar infrastructure, where edge devices control inverters and transmit high-frequency telemetry to centralized platforms. The slicing design separates (i) mission-critical command/control traffic — such as curtailment or inverter tuning commands from national energy coordinators, from (ii) periodic telemetry updates, and (iii) cloud-to-edge model deployment and optimization instructions. This separation guarantees ultra-high actuation latency and high resilience even in degraded network conditions.

The first development phase (by M18) will involve the deployment of a dedicated URLLC-compatible slice between edge gateways and the central orchestration platform over the 5G RedCap infrastructure. This includes testing redundancy features and fallback mechanisms at the edge. Results from this initial configuration will be reported in Deliverable D5.1. In Phase 2 (by M33), slicing orchestration will be enhanced to include priority-based scheduling of control message and integration with OSS systems from ORO for real-time performance feedback. The slicing configuration will also be validated during full-scale solar production prediction trials and grid-integration demonstrations, in collaboration with WP2 and WP3.

In the **T1 and T2 use cases**, the computing continuum involves edge nodes, managed via Kubernetes, and extreme edge nodes or devices like RSUs (T1) or UGVs (T2) managed via K3S. On top of them, the Resource Orchestrator through a Resource Allocation Engine is responsible to handle all the resources in the different clusters, distributing the containers among the nodes taking into account the constraints on power consumption and battery level. Automatic task offloading from RSUs or UGVs towards the edge are managed through closed loops, with the objective of guaranteeing service continuity while optimizing the use of batteries and renewable energy sources on the devices.

Both for T1 and T2 the Control Plane of the Core Network will be properly configured with two distinct slices. The development and implementation work will entail, among other things, the integration between the Resource Orchestrator and the HPE ANW P5G core network, so to allow control-plane re-configurations and re-provisioning automated tasks. In T1 one slice will be dedicated to URLLC safety traffic, and the other handling all remaining traffic types. Each slice will be identified by a dedicated UPF, configured with its corresponding Slice/Service Type (SST) and, if needed, a Slice Differentiator (SD). To enable this setup, the SMF will be connected in advance to both the UPFs and configured to interact with them in the right way. All the setups will be ready for phase 1 (M18) and fully tested and operational for phase 2 (M33). In T2 the slices will be dynamically allocated depending on the robot's need. The plan will follow the same deadlines as for T1.

The Core Network used in T1 supports network slicing capabilities, where each slice instance is configured by assigning specific User Plane Functions (UPFs) to dedicated S-NSSAI values. These identifiers uniquely define each slice and determine how resources, including RAN elements, are allocated. Once the UE is onboarded, it can be assigned to one or more slices, as sketched in Figure 2-21. This flexible architecture allows the system to dynamically route traffic and deliver services based on slice-specific requirements, ensuring that URLLC and eMBB traffic are handled according to their respective performance needs.

In T2, the same slicing architecture is applied as for T1. The Core Network enables the configuration of slice instances by mapping UPFs to distinct S-NSSAI values. UEs, such as 6G-connected UGVs, are

assigned to slices during onboarding, allowing them to operate across multiple slices simultaneously. This flexibility is essential for managing the dynamic offloading of tasks from the UGV to the edge, where URLLC slices can support low-latency command flows and eMBB slices can handle high-bandwidth sensor data streams.

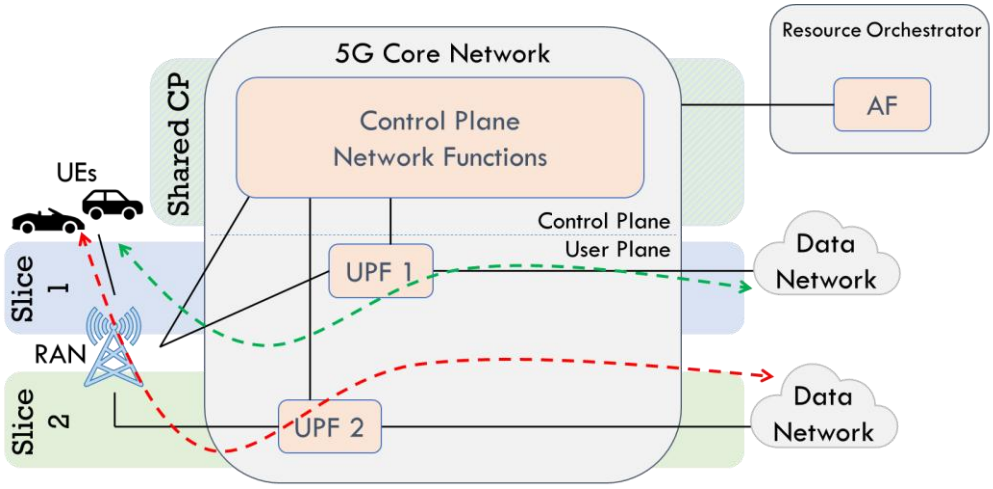

Figure 2-21 Network architecture with two slices with shared control plane and dedicated UPFs

The TUC testbed provides a static network slice on top of the public mobile network. This slice ensures dedicated resources for the **T3 (railway) use case**, isolating safety-critical traffic such as sensing signals, camera feeds, AI inference results, and train-to-train notifications from regular best-effort services. The static slice is geographically restricted to the area covered by the boundaries of the smart rail connectivity campus infrastructure and covers the whole of the 26kms of railway tracks where the demonstrations will be carried out. This ensures that the guaranteed QoS and resources are available precisely where they are needed, without unnecessary overhead outside the operational region.

The provider of the public network (Vodafone) issued a private Data Network Name (DNN) for the TUC slice, providing a logical separation of traffic flows. This allows the railway applications, e.g., ISAC data fusion (T3), to be cleanly separated from generic mobile services that are provided to the public via the same network infrastructure in the coverage area. The private slice also allows definition and application of their own (slice internal) policies.

The user plane traffic for the slice is anchored locally at the TUC-provided UPF (the UPF is physically located at the test track). This enables local breakout of traffic, minimizing latency by keeping the data path within the testbed domain, which is particularly critical for functions such as real-time obstacle detection, and ISAC data fusion.

While the user plane is terminated locally at the TUC UPF, the signalling and control plane interactions remain supported by the operator's public 5G core network. This hybrid approach enables seamless integration with the broader network and compatibility with existing mobility and authentication frameworks, while still achieving low-latency local data handling.

Within the static slice, multiple QoS profiles can be applied to match the requirements of each use case. For example, low-latency communication profiles (<15ms RRT) can be allocated to ISAC , while less stringent QoS classes can support monitoring or non-safety-related data exchange.

For the **T4 use case**: In real-world deployments, slices are preconfigured and assigned at session start, so users cannot switch slices dynamically. As a result, a user remains in the assigned slice for the duration of the session. Moreover, deploying an end-to-end slice requires synergy across all network domains. For a teleoperation use case, the RAN must prioritize the vehicle's traffic at the radio interface, the TN must maintain a deterministic low-latency route, and the 5GC must provision and manage a dedicated set of network functions for that slice.

To enable such a synergy, our approach consists of deploying a Cross-Domain Controller (CDC) as a centralized orchestration entity capable of coordinating all network domains i.e., RAN, TN, and 5GC, through dedicated connector interfaces (e.g., APIs, sockets), as illustrated in the following figure. The connectors harmonize the interaction between domain-specific controllers (e.g., RIC and TNC) and the CDC, enabling seamless interoperability even when each controller uses different message formats, data models, or protocols. Moreover, the connectors translate and normalize control and monitoring messages, ensuring that the CDC operates with a consistent internal data format while still communicating effectively with heterogeneous domain controllers.

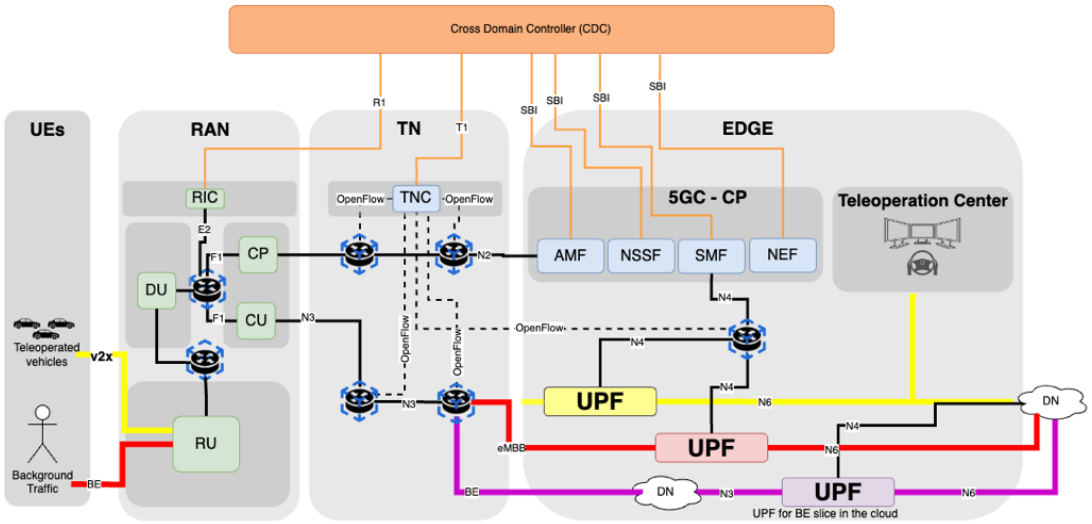

Figure 2-22 End-to-end Network Slicing architecture enabled by the Cross Domain Controller

In the RAN, the CDC interfaces with the RIC to adjust radio resources e.g., PRBs, scheduling policies, and beamforming configurations. Within the TN, it communicates with the TNC to configure deterministic, low-latency forwarding paths via OpenFlow-enabled programmable switches and routers. In the 5GC, it coordinates with control plane functions to dynamically instantiate, relocate, and scale UPF instances between central and edge locations.

With the deployment of the CDC, the operators can deploy multiple, isolated end-to-end slices tailored to different service requirements. Slice creation can be automated through pre-defined templates that encapsulate the necessary QoS parameters, routing policies, ensuring rapid and consistent instantiation across the domains (RAN, TN, and 5GC). Furthermore the CDC must support proactive, closed-loop resource management based on real-time monitoring and predictive analytics. When network monitoring tools indicate that SLA requirements are at risk, the CDC should dynamically reallocate network resources across the network domains. For example, when a surge in teleoperation demand is detected during peak traffic periods in dense urban environments, the CDC can trigger automated reallocation of radio resources, reconfiguration of transport paths, and scaling of UPF computing capacity. Conversely, resources can be released when demand subsides, improving overall network efficiency without compromising SLA compliance.

The implementation phase also involves extensive end-to-end validation to ensure that the system meets the stringent requirements of teleoperation. Real-life testing verifies the responsiveness of the RAN to PRB reallocation and beamforming changes, the speed and reliability of TN path switching under congestion, and the latency impact of relocating UPF instances during an active session. Performance testing confirms that the URLLC slice consistently delivers less than 20 ms end-to-end latency for control messages, more than 25 Mbps uplink throughput for multiple high-definition video streams, and 99.999% connection reliability under realistic background traffic conditions. The implementation phase includes: cross-domain programmability, seamless network slicing, dynamic resource scalability, and end-to-end validation, as illustrated in Table 2-4.

The dynamic network slicing will be implemented in the scope of **T4 use case**, by integrating IMEC's Open5GS-based 5G Core with the TUC testbed. The requirements and preliminary tests have been initiated prior to integration, which will take place during the second year of the project.

Table 2-4 Dynamic slice configuration for T4 use case

Phase	Goal	Key Actions	Need within T4
Cross-Domain Integration	Establish interoperability between CDC and all network domains	Deploy CDC with secure interfaces; install connectors to RIC, TNC, and 5GC control functions; harmonise message formats across heterogeneous controllers	Reliable bidirectional communication between CDC and domain controllers with unified data exchange format
Cross-Domain Programmability	Integrate programmable controls across RAN, TN, and 5GC	Deploy Cross-Domain actions to interact with the single controllers for dynamic resource allocation	CDC adjusts radio resource allocation and in RAN, reroutes TN paths, and relocates compute resource to edge node (UPF) near the teleoperated vehicle.
Slice Template Configuration	Define slice deployment parameters	Create slice templates including QoS profiles, routing policies, and UPF placement (edge or cloud)	Automated, consistent instantiation of isolated end-to-end slices across RAN, TN, and 5GC
Seamless Network Slicing	Support multiple, isolated end-to-end slices	Deploy a mechanism to achieve seamless network slicing across all the network architecture. Configure RIC to map UE IDs to URLLC slice; map it to TN slices; assign dedicated UPF at edge for slice	Teleoperated vehicle traffic is detected as part of the URLLC slice in each network domain.
Network monitoring mechanism	Adapt resource allocation to real-time demand	Implement closed-loop monitoring of KPIs network performance for each network domain	Data traffic is monitor in each Network domain to have a full network overview about the network conditions
Closed-Loop Automation and scaling	Enable proactive, real-time resource scaling	Integrate telemetry collection from all domains; implement SLA monitoring;	Automatic network adjustments to maintain SLA under

		configure automation logic for resource reallocation, TN path reconfiguration, and UPF scaling	changing demand and network conditions
End-to-End Validation	Verify functionality and SLA compliance	Test RAN responsiveness, TN path switching, and UPF relocation under load; measure latency, throughput, reliability	Under peak background load, URLLC slice maintains <20 ms latency, >25 Mbps uplink, and 99.999% reliability

2.3 Integrated sensing and communication (ISAC)

2.3.1 Description of the enabler

The ISAC enabler is a modular platform that integrates radio communication and radar-like sensing within a single radio system. It repurposes standard cellular base-station hardware and waveforms to operate as dual-purpose nodes, delivering high-throughput connectivity while simultaneously estimating range, velocity (Doppler), and angle of surrounding objects. The architecture supports monostatic and bistatic configurations, beam sweeping and digital beamforming, and can be extended with super-resolution methods (e.g., MUSIC/ESPRIT) for sub-beam angular and Doppler refinement.

Sensing outputs are produced from channel measurements and reference signals compliant with B5G/6G, enabling precise delay extraction, multipath analysis, and angular characterization without dedicated radar payloads. AI-assisted detection and multi-sensor data fusion (e.g., with cameras or inertial sources) are optional components that enhance classification and robustness.

The enabler is vendor-agnostic and scalable: it can be deployed on moving or fixed nodes, operates in standardized spectrum with large available bandwidths, and exposes clean interfaces for system integration (data, control, and KPI reporting). By sharing infrastructure between communication and sensing, it reduces duplication of hardware, simplifies deployment, and enables continuous situational awareness alongside reliable connectivity. While motivated by railway requirements in this project, the design principles are domain-agnostic.

2.3.2 Use case association and contributing partners

Table 2-5 Mapping between integrated sensing and communication and the use cases

	ISAC
T3	TUC, UoS

The ISAC enabler is only associated with the T3 use case in the project, and is tightly integrated with the leaky coaxial (LCX) cables for Future Railway Mobile Communication System (FRMCS), serving a shared purpose which is to increase safety in railway by enabling obstacle (animate and inanimate) detection, notifying in advance train drivers about incoming situations and propagating the information to following trains to minimize the scale of eventual accidents. A preliminary high-level design of the overall solution is available in the Context view Section 4.13 of D2.1.

ISAC is essential for this use case because it enables the integration of communication and sensing functionalities into a single system, allowing trains to detect obstacles on the tracks while maintaining reliable connectivity. By fusing sensing data from onboard modules, cameras, and trackside equipment with AI-based analysis, ISAC provides timely and accurate information to the train driver and following

trains. This ensures higher safety, reduces the risk of accidents, and minimizes delays, all while leveraging the same wireless infrastructure for both communication and sensing.

2.3.3 Design, development and implementation

The FR2-based ISAC design, as shown in Figure 2-23, leverages the mmWave bands around 28 GHz to transform standard base stations into dual-purpose units for both communication and radar-like sensing. At these frequencies, the short wavelength and large available bandwidth provide fine spatial resolution, enabling precise range and angle estimation that is not possible at FR1 bands. The design concept is to use a 5G FR2 transmission in standard protocol operation and simultaneously treat it as a monostatic radar (onboard the train) or bistatic radar where collaboration between two 5G FR2 base-stations are considered. In this mode, transmitted 5G waveforms are reflected by surrounding objects such as trains, trackside equipment, or obstacles, and the base station (or another base-station in the bistatic sensing case) collects these echoes for sensing. This approach allows seamless integration of radar functionality into communication infrastructure, reducing hardware duplication and enabling advanced railway safety applications.

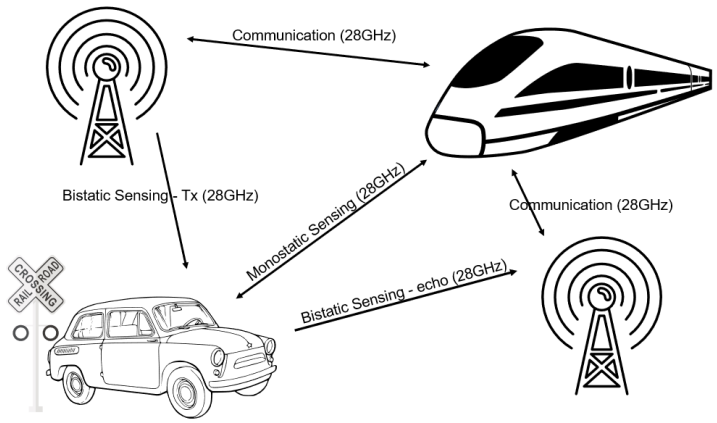

Figure 2-23 FR2-based ISAC for the T3 use case

The design builds on channel measurements and extraction in FR2 bands, capturing path delay, multipath components, Doppler signatures, and angular spreads in realistic railway settings. Delay analysis is used to determine achievable range resolution, while radar cross-section and link budget studies establish maximum detectable ranges. Doppler analysis quantifies velocity estimation accuracy for high-speed trains and allows fine discrimination of moving objects. Angular resolution is enhanced through beam sweeping and digital beamforming, and further refined by super-resolution approaches such as eigenvalue decomposition, MUSIC, or ESPRIT, which enable accurate angle and Doppler estimation beyond classical array limits. These analyses form the basis for evaluating sensing capabilities under 5G FR2 waveforms and protocols, ensuring compatibility with FRMCS infrastructure.

The development and implementation plan proceeds through the following steps:

- Perform 28 GHz channel sounding and measurement campaigns in railway-relevant environments.
- Perform 28 GHz channel emulation in railway-relevant environments considering two 5G FR2 base-stations collaboratively performing bi-static sensing.
- Extract channel impulse responses using 5G FR2 standard-compliant reference signals.
- Apply delay analysis to evaluate range resolution and maximum detection distance.
- Conduct Doppler analysis for velocity estimation of fast-moving trains.
- Enhance angular and Doppler resolution through super-resolution algorithms such as MUSIC or eigenvalue-based decomposition.
- Prototype algorithms for obstacle detection and infrastructure monitoring using FR2 base station data.

- Integrate the sensing outputs with communication functions in a monostatic or bistatic FR2 transmission.
- Validate sensing performance under FRMCS requirements for latency, coverage, and reliability.

By combining high-resolution mmWave radar techniques with super-resolution methods, this enabler turns the 5G FR2 base station into a powerful dual-use platform. It provides precise localization, obstacle detection, and environmental monitoring, while simultaneously delivering ultra-fast communication links for railway signaling and passenger services.

The results evaluating the performance of ISAC will be documented and presented in Deliverable D3.2.

2.4 Public-private network integration

2.4.1 Description of the enabler

This enabler aims to enable Non-Private Networks to interwork with private networks to extend coverage, up to the public network coverage, even if with lower performances, but enough to guarantee the minimum required for applications. Dedicated spectrum can be allocated to private network to avoid interference. The integration can be realised in different ways and architectures, for example with separated Core network or UPF, or sharing the core network allocating dedicated gNBs for public and private RAN (with dedicated spectrum) or also sharing the gNBs. The routing can be configured either via public or private IP addresses, according to the private network configuration.

In general, there are two types of non-public/private networks: SNPNs (Standalone Non-Public Networks) are private 5G networks that operate entirely independently of any public network, while PNI-NPNs (Public Network-Integrated NPNs) integrate with existing public mobile networks to provide private communication services. An PNI-NPN can either only use the radio network part of the public network or share both the radio network and core network control plane of the public network, via e.g. network slicing.

2.4.2 Use case association and contributing partners

Table 2-6 Mapping between public-private network integration and the use cases

	Public-private network integration
E2	TNO
T1, T2	HPE, TIM
T5	ACRO

In the **E2 use case**, connectivity needs to be maintained and guaranteed between the drone and the base station (located at the offshore platform), as well as stable and reliable connection between the base station and the core network, which can be located either offshore or onshore. Due to the nature of the use case, being offshore, it could be the case that there are no other deployed networks for providing connectivity at the offshore location. In this scenario, the deployment of a standalone private network will facilitate the connectivity of the drone and enable its communication with the servers (and digital twin) located onshore. Another relevant scenario where private networks are required is when connectivity at the offshore location is provided by one network operator. In this case, the different enterprises or organisations operating offshore can establish a customer relationship with the network operator and have their own private network based on the public network, operated and maintained by the network operator. Therefore, both SNPNs and PNI-NPNs can be seen as critical components to establish connectivity in offshore locations.

In the **T1 use case**, the availability of large-scale public network coverage allows users to interact with the service even outside the Private Network area. More in detail, visually impaired users will be assisted while navigating urban intersections moving from public to private network and vice versa depending on the connectivity needs.

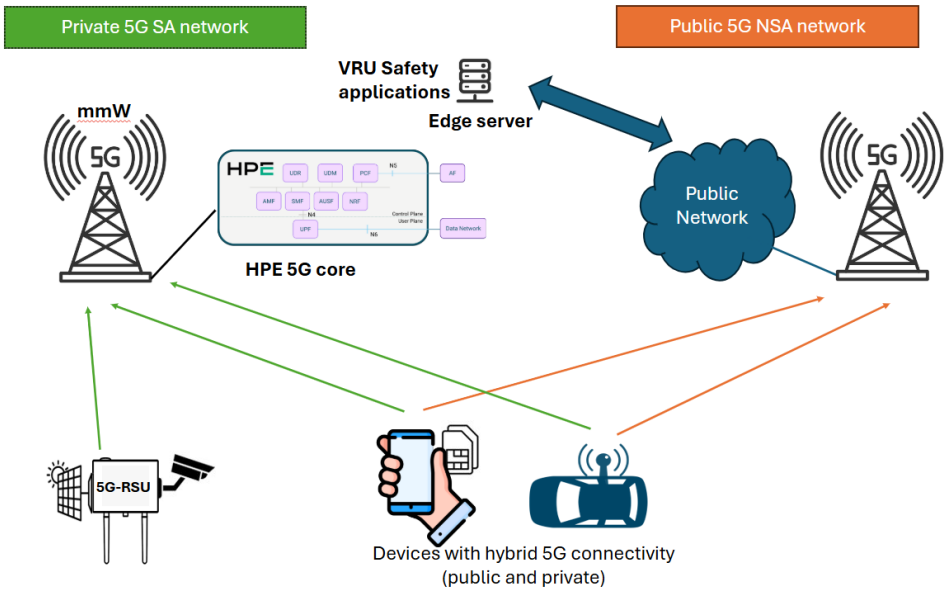

Figure 2-24 Public-private network integration for the T1 use case

As shown in Figure 2-24, the trial will test the dynamic transition between:

- **Private B5G network**: Deployed locally at critical points such as busy intersections, offering ultra-low latency and localized processing. The pilot plans to test mmW in order to assess the possibility of this technology for the considered use case.
- **Public 5G network**: Offering continuous, reliable coverage outside the immediate private network zones, it guarantees access to localized computing resources to maintain service performance.

The objectives are to test handover mechanisms and service continuity as users move from a private B5G zone into the public 5G network and vice versa and to ensure that user assistance (e.g., obstacle alerts, navigation cues) remains reliable and with acceptable latency across both networks. This will be done with dual-connectivity devices: In the smartphone case, a dual-SIM device will ensure the connectivity with both networks; In the case of cars' OBU, a dual model approach will be taken with a standard 5G device for the public network and a dedicate device for the mmW private connectivity. The RSU is a fixed element requiring, in certain situations, the streaming of large data flows (e.g. camera and LiDAR flows). For this reason, it will be connected to the private network that provides a large amount of available bandwidth.

In the **T2 use case**, UGVs will be dispatched to urban areas following citizen alerts to autonomously monitor the environment and detect security threats.

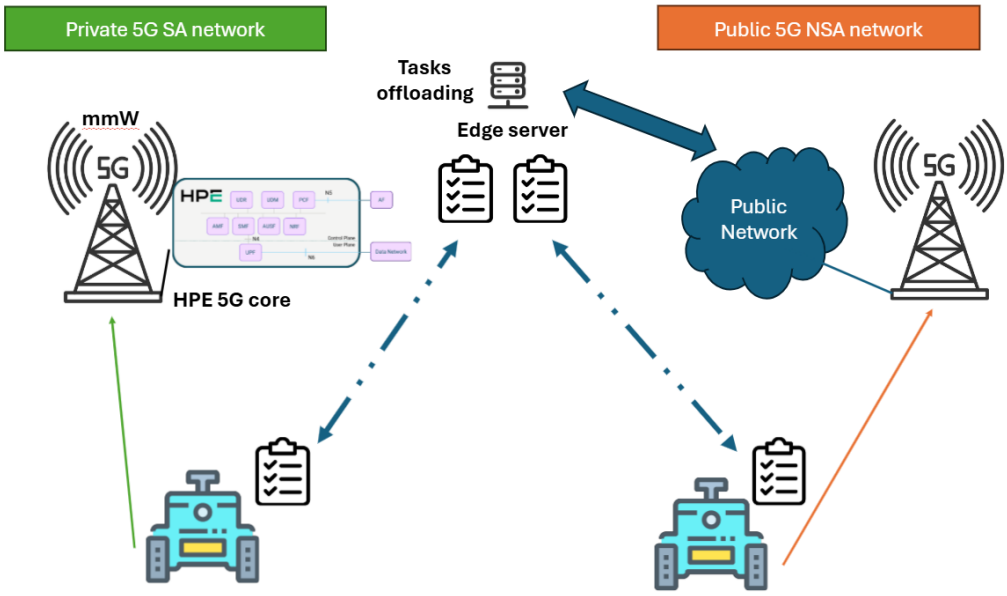

Figure 2-25 Public-private network integration for the T2 use case

As shown in Figure 2-25, the following networks will be involved in the T2 trials:

- **Private B5G network**: Essential for offloading critical driving functions (e.g., path planning, obstacle avoidance) that require ultra-low latency and highly deterministic communication. Even with private network support, achieving the strict performance needed for real-time driving control remains an open challenge and will be a key focus of experimental validation.
- **Public 5G network**: Offloading non-critical functions (e.g., image recognition, object detection) to nearby edge/cloud resources. Leveraging wide coverage, cost efficiency, and distributed intelligence of public 5G infrastructures

The objectives are to validate the split of functionalities according to network capabilities and to assess the real-world feasibility of UGV teleoperation and autonomy using private B5G support.

In the **T5 use case**, public–private network integration refers to seamless swap of backhaul connectivity of the private network between local connection and use of public cellular networks operated by telecom providers. This integration allows organizations to benefit from the security, customization, and dedicated resources of private networks while maintaining as backup the public infrastructure. For example, devices and assets while remaining securely connected within a private 5G or 6G network on-site can use different backhaul connectivity options when reaching not operation critical aspects of the wider infrastructure via the public network.

2.4.3 Design, development and implementation

In general, to enable the interconnection between public and private networks in a 5G environment, three design approaches are adopted:

- **Public Network Integrated - Non-Public Network (PNI-NPN)**: defined by 3GPP, refers to the actual integration of a private network with a public one. This design allows the private network to leverage part of the public infrastructures, keeping the dedicated functionalities and management.
- **Dual-sim**: allows a device to connect to two mobile networks simultaneously. It enables seamless switching or parallel use of public and private networks.
- **Roaming**: enables a mobile device to access services outside its home network by connecting to a visited network. In 5G, it supports seamless mobility across regions and operators

In the following, considering the use cases proposed in the project, the PNI-NPN and Dual-sim approaches have been implemented (roaming case is out of scope).

For the **E2 use case**, private networks will be used to establish connectivity at the offshore location and ensure a reliable connection for the drone communication. Different designs are being considered, depending on how the private network will be configured. Firstly, an SNPN is considered, where both the radio and core networks are deployed and operated by the private entity. For this design, the Amarisoft's Amari Callbox Classic edition will be used to act as a dedicated gNB for the private network. For the core network, the Open5GS will be used, which will also be dedicated to the private network. Both the radio and core networks will be deployed at the Zephyros lab in Vlissingen, The Netherlands. This design will showcase the benefits of an SNPN, e.g., optimized gNB deployment for the private network, and potentially wider frequency carriers as the radio network is not shared with other private networks.

For the scenario of a PNI-NPN, two different deployments are considered. For both deployments, it is assumed that a public network is operating at the offshore location and that the private network will use (part of) the infrastructure (i.e., gNB and core network) of the public network. Note that in the implementations, the public network will be emulated instead of using an actual existing public network. The first design is shown in Figure 2-26, which illustrates that both the gNB and core networks will be shared between the public and private networks. Similarly to the SNPN, the gNB will be provided by the Amari Callbox and the core network by the Open5GS. Both radio and core networks will be deployed at the Zephyros lab in Vlissingen (representing the offshore location). Regarding the emulation of the public network, different UEs will be deployed to generate background traffic, acting as traffic of the public network. Moreover, it can be considered that the location and beam direction of the gNB will be optimized for the background traffic to showcase the limitations of PNI-NPNs, which are not specifically designed and deployed to optimally serve the traffic of the private network.

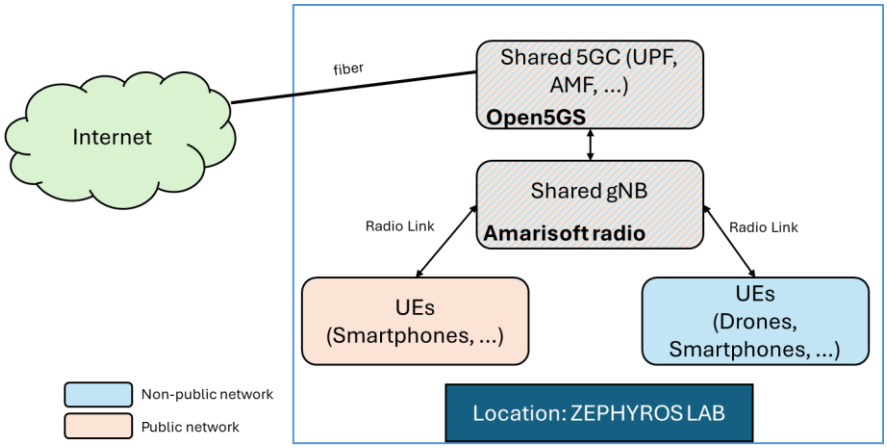

Figure 2-26 First design for public-private network integration for the E2 use case

The second design for the PNI-NPN is shown in Figure 2-27, where the same software, hardware and public network emulation method are used as in the previous design. Differently to the previous design, the shared core network control plane is now located at the TNO location in The Hague, The Netherlands, and two separate UPFs are deployed. One UPF is deployed at the TNO location in The Hague, meant to serve traffic from the public network, and the second UPF is deployed at the Zephyros Lab in Vlissingen, meant to serve traffic from the private network. This design aims to replicate the scenario where the public network operator deploys only its radio network offshore and the core network remains onshore. However, a dedicated UPF is still deployed offshore (at Zephyros lab) to serve the traffic from the private network in order to ensure lower latencies.

The results evaluating the performance of this enabler will be documented and presented in Deliverable D3.2.

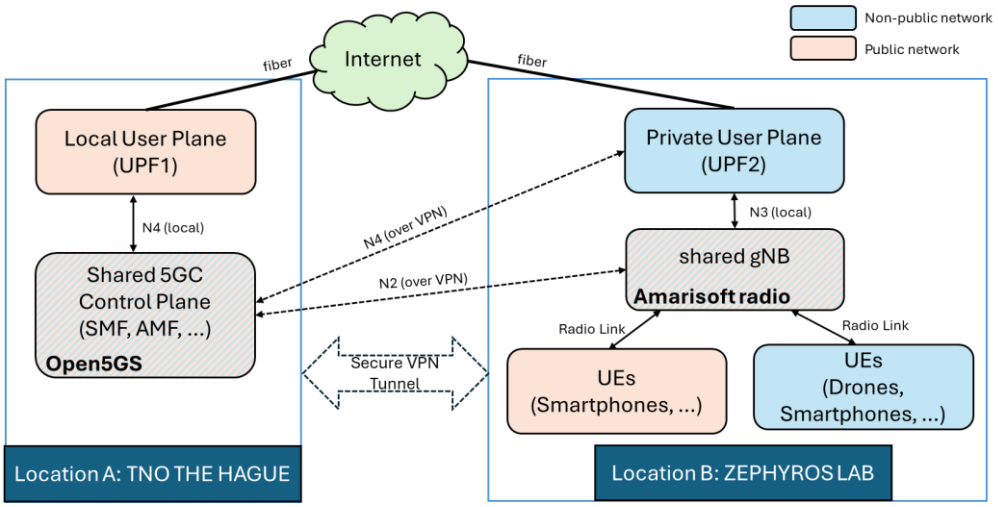

Figure 2-27 Second design for public-private network integration for the E2 use case

In the **T1 and T2 use cases**, the public-private architecture design will focus on integrating private B5G and public 5G networks to support advanced mobility and safety services in urban environments. The general goal of this architecture is to ensure seamless connectivity as users, devices and vehicles move between different network areas. Devices involved, such as smartphones and vehicles, will be equipped to operate across both networks, using dual connectivity.

The general design and implementation will follow a phased approach, beginning with the deployment in the Torino cluster of private infrastructures in the area already covered by public (commercial) 5G, as illustrated in Figure 2-28. In particular, the private 5G network will be deployed at selected locations, to provide low-latency and high-bandwidth communication, where specific capabilities (localized processing) are essential for UCs requirements. The virtualized private 5G core network will be provided by HPE Aruba Networking P5G, while the public one will be provided by TIM. The implementation plan will continue with the public-private network integration, followed by a UC specific handover scenario and the monitoring of key performance indicators.

In phase 1 the private network will be fully operational while the integration (for T1) and the comparison (for T2) with the public network will be performed for phase 2.

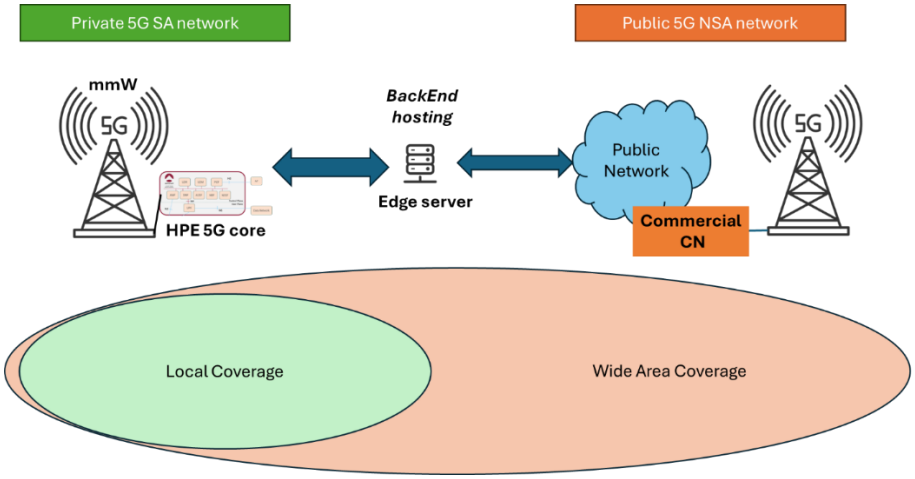

Figure 2-28 Torino Cluster Public private enabler architecture

In the **T5 use case**, the design and implementation of the network will adopt a phased approach to ensure smooth integration and reliable performance from the outset. The initial phase will focus on deploying private B5G infrastructures within areas of the port that are already covered by existing public (commercial) B5G services, enabling direct comparison, interoperability testing, and seamless

transition between the two domains. A private network will be established at selected strategic locations within the Port of Thessaloniki, tailored to support critical use cases such as crane teleoperation, asset tracking, and real-time monitoring. The private base station, supplied by Acromove, will serve as the core enabler of this infrastructure, providing dedicated, low-latency, and high-reliability connectivity for port operations. The implementation plan emphasizes close collaboration and information exchange between private and public stakeholders to ensure alignment on interoperability, security, and service continuity. In addition, the rollout will be continuously evaluated through the monitoring of key performance indicators (KPIs) such as latency, throughput, reliability, and coverage, enabling data-driven optimization at each stage and ensuring the system meets the demanding requirements of next-generation port operations.

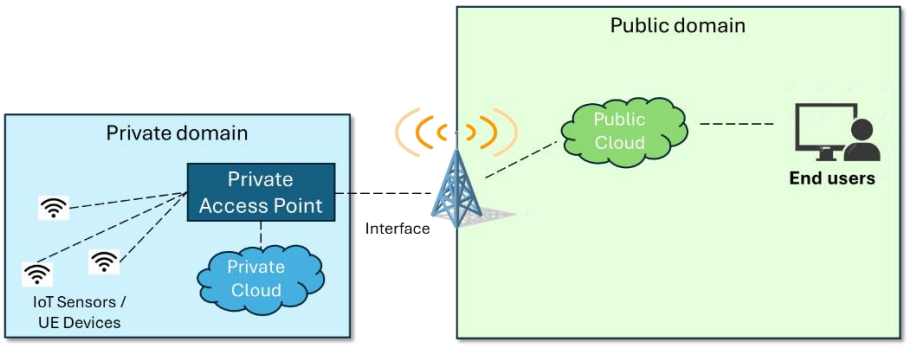

Figure 2-29 Public-private network integration for the T5 use case.

For T5, a first version of the enabler is under development. Initial testing will take place in lab environment in the 1st year. After this period, deployments in the field will follow in the 2nd year for conducting some preliminary experiments. In the 3rd year, final trials will take place.

2.5 Multi-RAT connectivity

Multi-RAT connectivity aims to ensure network coverage and resilience even in difficult-to-reach areas. The following options (sub-enablers) are considered: B5G sidelink, Wi-Fi integration and NTN integration. Each of these sub-enablers will be described separately below.

2.5.1 Description of the enabler

B5G sidelink: Starting from LTE, the sidelink interface has been introduced in 3GPP to allow for direct communication between devices and to extend network coverage through relaying. Sidelink is seen as an important feature and particularly useful for scenarios where coverage is limited, as it can extend the coverage, and for applications such as Vehicle-to-Everything (V2X), where it allows for low-latency communication. In 5G, sidelink is being enhanced to support further V2X use cases, enhance operations for device-to-device communication and for relaying to the network via another device, Industrial IoT, positioning, and operation in unlicensed spectrum. Note that the relaying to the network via another device is important for network coverage extension as well as for low-power devices, which due to their limited power cannot maintain a connection to the network.

Different protocol stacks for the user and control planes have been implemented, depending on whether the relaying functionality is in layer 1, 2 or 3. Particularly for V2X applications, a dedicated and licensed Intelligent Transport Systems (ITS) spectrum is used, as well as dedicated ITS higher protocol stacks. Additionally, two operational modes have been standardized in terms of how the radio resources are allocated to the sidelink between two UEs. In mode 1, the resource allocation is handled by the base station, which requires that both UEs maintain a connection to the base station. In mode 2, the resource allocation is performed without the support of the base station, and thus it is not required for both UEs to be connected to the base station.

Wi-Fi integration: The integration of Wi-Fi within a Multi-Radio Access Technology (Multi-RAT) framework is a key enabler for achieving seamless and efficient connectivity across heterogeneous wireless networks. In the context of 5G and beyond, Multi-RAT connectivity leverages both licensed (e.g., 5G NR) and unlicensed (e.g., Wi-Fi) spectrum to optimize user experience, network performance, and resource utilization.

Wi-Fi integration supports dynamic traffic management through mechanisms such as Access Traffic Steering, Switching, and Splitting (ATSSS), allowing traffic to be intelligently distributed across available RATs based on real-time network conditions, user policies, and service requirements. This enables enhanced reliability, throughput, and latency performance, particularly in dense urban or indoor environments where Wi-Fi coverage is prevalent.

Furthermore, standardized interfaces and protocols—such as those defined by 3GPP and IEEE—facilitate interoperability between cellular and Wi-Fi domains, enabling unified mobility management, session continuity, and coordinated handovers. The integration also supports advanced features like multipath transport (e.g., MPTCP) and policy-based routing, which are critical for enabling robust and flexible connectivity in future network architectures.

NTN integration: Non-Terrestrial Networks (NTNs) are a fundamental component of the evolving 6G architecture, designed to extend connectivity beyond the limitations of terrestrial infrastructure. By incorporating satellite systems, high-altitude platforms (HAP), and other aerial or spaceborne assets, NTNs enable coverage in remote, rural, maritime, and underserved regions, as well as in scenarios where terrestrial networks may be disrupted or unavailable. In the context of multi-RAT connectivity, NTNs contribute to a seamless and resilient communication environment by integrating with terrestrial cellular networks. This allows devices to maintain service continuity across heterogeneous access technologies, supporting mobility, global reach, and dynamic adaptation to varying network conditions. The technical realization of NTNs builds on 3GPP-defined architectures and protocols, starting from Release 17, and involves addressing challenges such as variable latency, Doppler effects, and intermittent coverage. NTNs also interface with core network functions to support mobility management, authentication, and quality of service enforcement. Their inclusion in the 6G ecosystem enhances the overall robustness, flexibility, and availability of future networks.

2.5.2 Use case association and contributing partners

Table 2-7 shows the association of AMAZING-6G use cases with the multi-RAT connectivity sub-enablers.

Table 2-7 Mapping between the Multi-RAT connectivity sub-enablers and the use cases.

	B5G sidelink	Wi-Fi integration	NTN integration
H1	TNO	TNO	
H2	TNO	TNO	
P2		VTT	VTT
P4		VTT	
E2			TNO
T3	TUC		
T5		CERTH, ThPA, Acro	

In the **H1 and H2 use cases**, it is critical to ensure reliable connectivity between the medical devices and the hospital data center. A big challenge for reliable connectivity is lack of coverage or a weak signal strength from the cellular network, as well as the fact that the medical devices are low-power devices. To enhance/extend coverage, B5G sidelink can be used. In particular, with B5G sidelink the medical device can connect to the B5G/6G network via a different device, e.g. a smartphone, which is in the vicinity of the medical device, and has a better connection to the B5G/6G network (and higher transmission power) than the medical device. Additionally, the Wi-Fi integration sub-enabler can be used to ensure continuous and reliable connectivity for medical devices when cellular coverage is weak, unstable or non-existent. In this setup, devices can seamlessly switch to trusted Wi-Fi access points that are securely anchored to the B5G/6G core network. This maintains consistent data flow to hospital systems, supports real-time monitoring, and enhances resilience against connectivity disruptions, which is critical for health-related applications.

The **P2 use case** leverages both Wi-Fi integration and NTN integration sub-enablers. Integrating these technologies with B5G ensures seamless interoperability of mission-critical services, providing robust connectivity even under challenging conditions. These sub-enablers enhance redundancy and guarantee service continuity, which is essential for mission-critical operations.

The **P4 use case** relies on Wi-Fi integration sub-enabler. The focus here is on ensuring reliable connectivity in extreme arctic environments, where infrastructure is sparse and conditions are harsh. Wi-Fi integration with B5G provides a practical and efficient solution for maintaining communication links during search and rescue missions.

In the P2 and P4 use cases, multiple networks are employed to increase resilience and guarantee reliable connectivity. In these cases, a vehicle is equipped with Multi-Radio Access Technology (Multi-RAT), including B5G, Wi-Fi, and NTN, ensuring network access and service availability even in rural or remote areas. P2 and P4 demonstrate how multi-network strategies can be adapted to meet specific operational requirements, ensuring resilient and dependable connectivity in diverse and demanding scenarios.

In the **E2 use case**, the drone-based wind blade inspections are operating offshore and reliable communication is required with the onshore location. However, in offshore environments, network coverage by terrestrial networks is challenging, e.g. due to the limited locations where base stations can be deployed. NTN is seen as a potential solution for extending coverage in locations where there is limited or no coverage by the terrestrial network. Thus, an NTN provides the necessary connectivity to ensure reliable communication and data transmission, supporting service continuity and operational safety.

Sidelink is considered for the **T3 use case** to enable direct communication between trains without relying solely on the network infrastructure. This ensures that critical safety messages, such as obstacle detection alerts or sudden braking events, can be shared immediately with following trains, even in areas with limited network coverage. By providing low-latency, reliable train-to-train communication, sidelink enhances situational awareness and contributes to safer and more efficient railway operations.

Wi-Fi integration is important for the **T5 use case**, because it complements B5G by handling non-critical traffic, allowing operators to balance cost with performance by assigning the right network for the right task, ensuring seamless connectivity across the port ecosystem. This allows the dedicated B5G network to be utilized exclusively for the safety-critical teleoperation of the STS crane, while Wi-Fi supports broader operational needs.

2.5.3 Design, development and implementation

In this section, we first illustrate the overall high-level design concepts for the sub-enablers, followed by testbed specific (associated with different use cases) design, development and implementation.

B5G sidelink – High-level design: The B5G sidelink sub-enabler can be used to enable device-to-device communication, e.g. direct communication between vehicles, as well as allow for extended coverage via its relay functionality. The direct communication between devices is performed over the sidelink (or the Proximity Communication 5 (PC5) interface). Direct communication, instead of routing traffic through the RAN, can offer higher quality of service and reduced latency if the two devices are in close proximity. When designing B5G sidelink, different aspects should be taken into account, for example, the mode used for resource allocation for the sidelink, interference mitigation between the sidelink and the channels between the base station and other UEs, and communication range over the sidelink as UEs have lower transmission power and simpler antennas than the base stations.

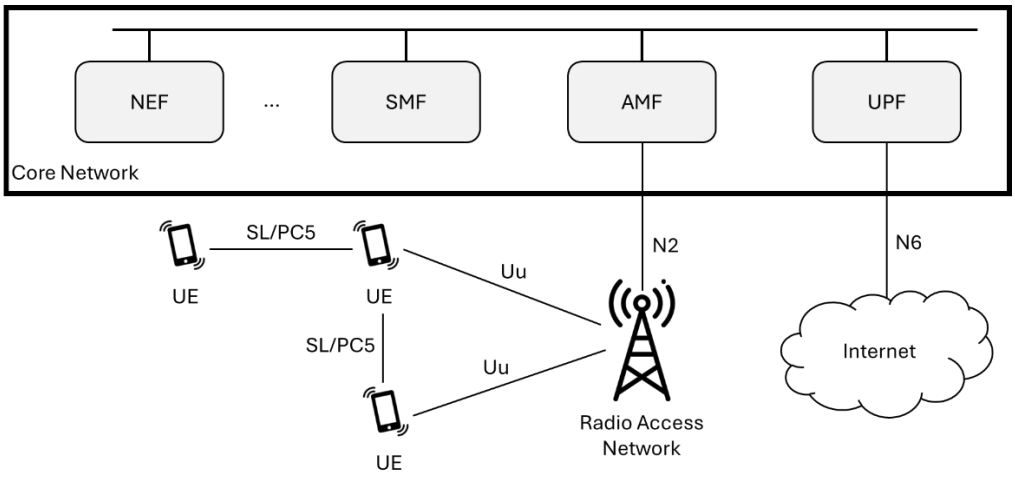

Figure 2-30 High-level design of B5G sidelink

Wi-Fi integration – High-level design: The Wi-Fi integration sub-enabler provides a flexible framework for incorporating Wi-Fi access into the 5G/6G ecosystem, leveraging standardized non-3GPP access integration mechanisms. Its primary objectives are to enable seamless connectivity, optimize multi-access performance, and facilitate experimentation with heterogeneous access scenarios in use cases during trials. Key design aspects are shown in Figure 2-31 and they are 1) Interworking architecture supporting 3GPP-defined non-3GPP access integration via the 5G Core (5GC), using N3IWF for secure connectivity and policy control. 2) Capability to implement Access Traffic Steering, Switching, and Splitting (ATSSS) for dynamic traffic management across Wi-Fi and 5G. 3) EAP-based authentication to ensure consistent security across access technologies. 4) Modular design allowing selective activation of features to match use case requirements.

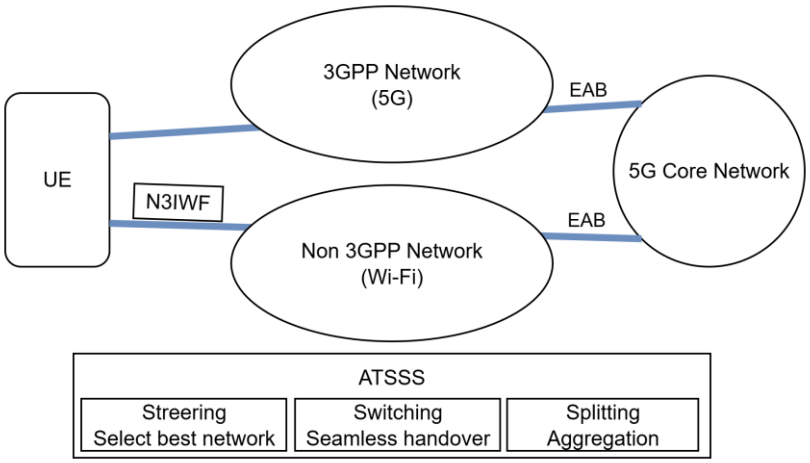

Figure 2-31 High-level design of Wi-Fi integration with a 3GPP network

NTN integration – High-level design: The Figure 2-32 presents two architectural models for integrating Non-Terrestrial Networks (NTN) into a 5G system to support connectivity for User Equipment (UE) in remote or offshore environments. In areas where terrestrial coverage is unavailable, satellites serve as access points to maintain reliable communication. On one side, the architecture shows a transparent satellite relaying communication between a ground-based gNB and the 5G Core. This model assumes that the gNB remains on land, and the satellite acts as a passive link, extending coverage to isolated regions. On the other side, a regenerative satellite hosts a gNB on board, allowing UEs to connect directly to the satellite as if it were a terrestrial base station. This configuration enables full access to the 5G Core without relying on ground infrastructure, making it particularly suitable for fully remote operations. Both models demonstrate how NTN can ensure service continuity, mobility support, and reliable data transmission for UEs operating in hard-to-reach locations, with the satellite acting either as a relay or as an active network node.

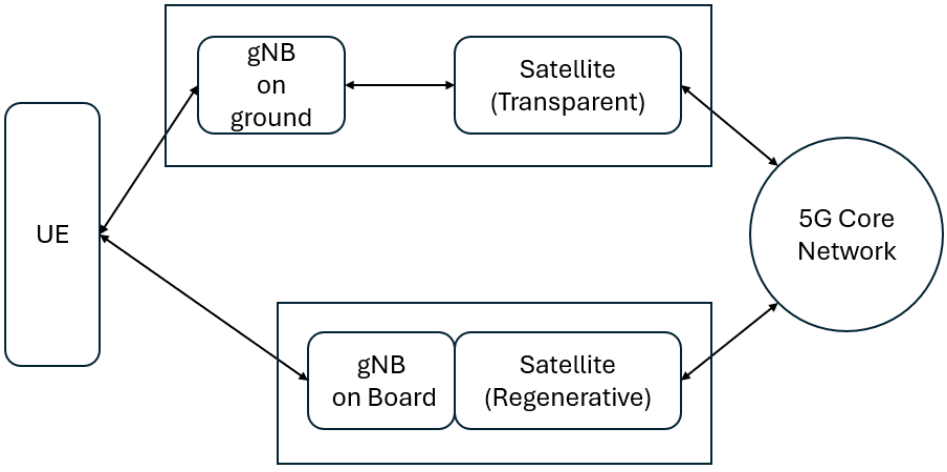

Figure 2-32 High-level design of Wi-Fi integration with a 3GPP network

In the rest of the subsection, we describe use case specific design, development and implementation for the enabler.

Design, development and implementation for the H1 and H2 use cases: With the relay functionality provided by B5G sidelink, the medical device can connect via e.g. a smartphone to the B5G/6G network, when there is not sufficient network coverage at the location of the medical device, or when the transmission power of the low-powered medical device is not sufficient to maintain a connection to the B5/6G network. For the evaluation of this scenario, both the medical device and the smartphone need to support B5G sidelink. However, the 5G sidelink devices that are commercially available (e.g. 5G-V2X

Sidelink Platform SIRIUS by Ettifos) are targeting vehicular communication applications and thus are specifically configured for the said applications, for example they are operating at the 5.9 GHz band and adhere to the ITS specific protocol stack. Therefore, due to hardware limitations and unavailability, the B5G sidelink enabler cannot be currently evaluated for the H1 and H2 use cases but it is still identified as an important enabler for the H1 and H2 use cases for future development. Monitoring on availability of hardware will be performed during the course of the project to identify whether the enabler can be evaluated at a later stage of the project.

The Wi-Fi Integration sub-enabler ensures continuous and reliable connectivity for medical devices by allowing them to switch to Wi-Fi when B5G coverage is weak or unavailable. In this scenario, Wi-Fi access points are securely anchored to the B5G core network, enabling seamless handover and unified mobility management. Figure 2-33 shows a home-based setup where a medical device, acting as the UE, connects via the Wi-Fi access point, which routes traffic to the 5G core, ensuring continuity of service.

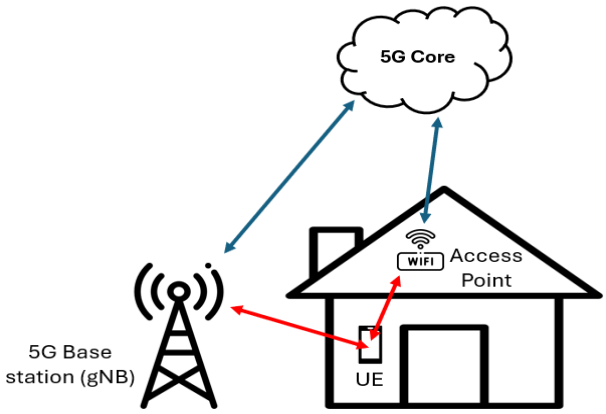

Figure 2-33 Home-based setup for Wi-Fi integration

For the broader view of 3GPP and non-3GPP access integration, Wi-Fi connectivity is enabled through the N3IWF interface, allowing the medical devices to maintain access to the 5G core via either 5G NR or Wi-Fi. This design supports uninterrupted health monitoring, secure data transmission, and consistent QoS, even in indoor or coverage-challenged environments.

The implementation begins with defining access selection policies and security requirements for switching between 5G and Wi-Fi. The system is configured to support non-3GPP access via N3IWF, ensuring seamless integration with the 5G core. Prototypes are developed to test handover behavior and QoS preservation. Integration focuses on validating connectivity and data integrity under varying network conditions. Deployment involves testing in real environments, with monitoring to ensure reliability and compliance with healthcare standards.

No further results will be reported on the evaluation of 5G sidelink for the H1 and H2 use cases due to software and hardware limitations. The results evaluating the performance of the Wi-Fi integration enabler will be documented and presented in Deliverable D3.2.

Design, development and implementation for the P2 and P4 use cases: a connected vehicle is equipped with 5G NR, Wi-Fi, and possible NTN satellite communication modules. Wi-Fi integration in multi-RAT connectivity enables to use Wi-Fi together with 5G or B5G when it is available and guarantees better data transmission and QoS to the connected vehicle. NTN integration in multi-RAT provides seamless and resilient communication. NTNs can be utilized in areas where connectivity via terrestrial networks is limited, especially in rural areas.

Currently, 5G Core (5GC) in the 5G Test Network (5GTN) does not provide Wi-Fi integration. If 5GC does not provide this support during the project, Wi-Fi integration is implemented at the device level either using multipath TCP (MPTCP) or network bonding. In addition, a network controller application is implemented to use network QoS information to select either Wi-Fi or 5G, or to distribute traffic across

both networks. Figure 2-34 shows the multi-connectivity setup for use case P2 and P4. The network controller may use Network and Energy management APIs in access network selection.

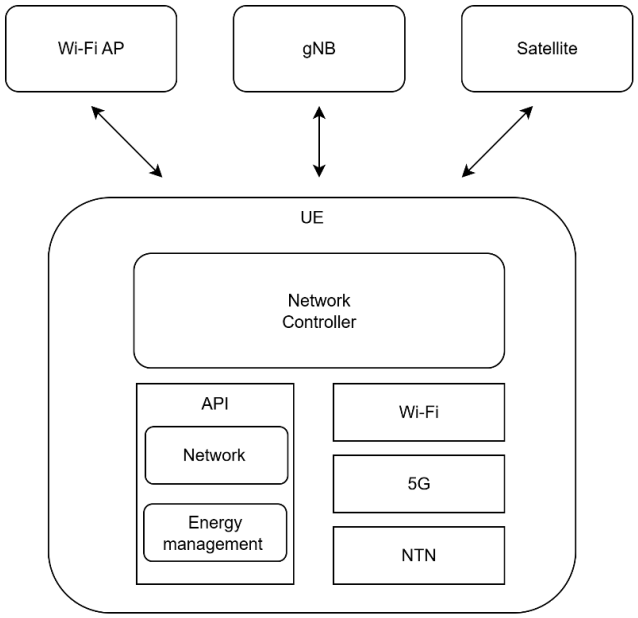

Figure 2-34 Multi-connectivity setup for the P2 and P4 use cases

NTN integration can be implemented using either transparent NTN where satellites are used as relays that forward user signals to ground-based gNBs or non-transparent NTN which integrates gNB functions on the satellite, enabling on-board processing, inter-satellite links, and lower latency. Transparent or non-transparent NTN integration will probably not be able to be implemented and tested because the necessary hardware and software components will not be available for 5GTN. Instead, NTN-integration is implemented using satellite technology and 5G terrestrial network in the vehicle. The vehicle can connect to satellite and 5G network using different multi-connectivity techniques. For instance, redundant (full duplication) where same data is transmitted over interface, dynamic where single interface is used at a time, or round-robin based technique where data is transmitted over both interfaces. Redundant mode offers high reliability and low latency for mission-critical applications but uses more energy and bandwidth. Dynamic mode saves energy and bandwidth, though with less reliability, and depends on real-time network QoS information to switch networks. Multi-network controller uses certain technique, and it can select optimal access technology based on QoS values from API. In addition, satellite connectivity is implemented between gNB and 5GC. The objective is to measure how NTN's latency and jitter affects network and application performance. Figure 2-35 shows the NTN backhaul setup.

The results evaluating the Wi-Fi integration and NTN integration for use cases P2 and P4, considering hardware and software limitations of the test network will be reported in D3.2.

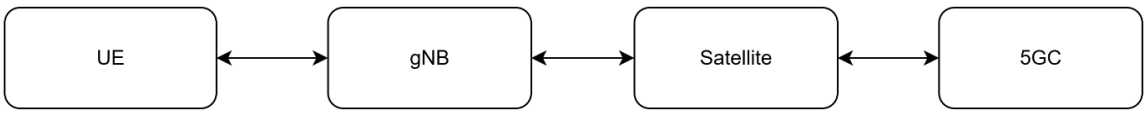

Figure 2-35 NTN backhaul setup for the P2 and P4 use cases

Design, development and implementation for the E2 use case: In the E2 use case, drones perform wind blade inspections in offshore environments where terrestrial 5G coverage could be unavailable. To ensure connectivity, the design leverages NTNs, where satellites act as base stations, providing direct access to the 5G core. Moreover, handover between terrestrial and non-terrestrial networks can be performed to ensure that the drone maintains a reliable connection to the network. For the evaluation of these scenarios, access to NTN is required, where the satellite can be configured to act as a base station.

However, with the current hardware and software available, it is only possible to emulate the Earth-space wireless link and evaluate the performance of the setup through simulations. Additionally, for the evaluation of the handover scenario, a device that supports connections to both terrestrial and non-terrestrial networks is required. Therefore, due to these two reasons, the NTN enabler will not be evaluated for the E2 use case but it is still identified as an important enabler for the E2 use case for future development. Monitoring on availability of hardware will be performed during the course of the project to identify whether the enabler can be evaluated at a later stage of the project.

Design, development and implementation for the T3 use case: In the T3 use case, sidelink communication enables direct train-to-train communication for safety-critical information exchange. Each train is equipped with a 5G modem supporting sidelink interfaces, allowing messages to be transmitted directly between nearby trains without the need to traverse the core network. When potential obstacles or emergency conditions are identified through ISAC, the results are disseminated through sidelink to following trains in real time. The sidelink channel operates in parallel with the traditional uplink–downlink communication, ensuring redundancy and ultra-low latency for critical notifications. To prioritize safety messages, alerts are transmitted with the highest reliability and minimal delay. The results evaluating the sidelink in T3 will be documented and presented in Deliverable D3.2.

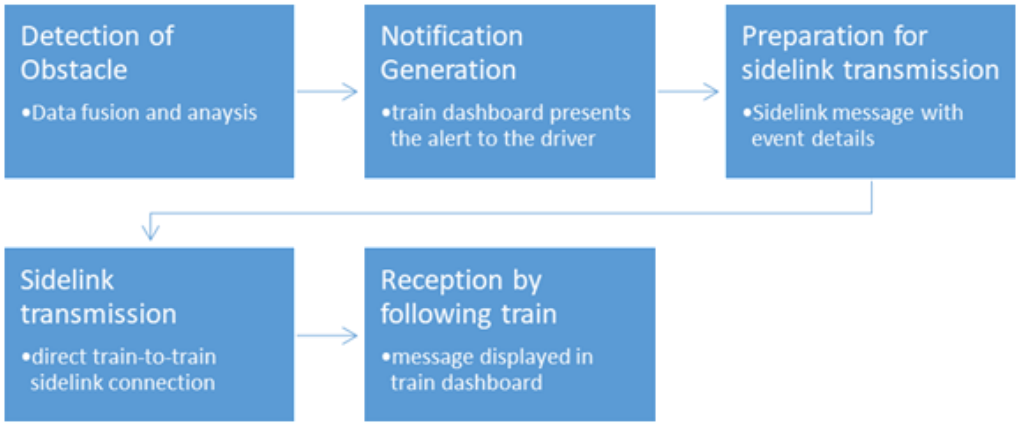

Figure 2-36 Sidelink-based information delivery in the T3 use case

Design, development and implementation for the T5 use case: The design of the Wi-Fi Integration sub-enabler focuses on creating a seamless, efficient, and robust multi-radio access technology (Multi-RAT) framework for the private 5G network. The goal is to intelligently leverage both licensed 5G and unlicensed Wi-Fi spectrum to support a wider range of devices and ensure uninterrupted connectivity for critical port operations. Also, Acromove's **"5G-In-box"** is the central component for Multi-RAT Connectivity. This portable edge data center includes a built-in Wi-Fi AP and a 4G/5G Router, which allows for a heterogeneous network. Overall, the solution provides:

- Standardized Interfaces: The design will leverage standardized 3GPP and IEEE protocols to ensure smooth interoperability. This includes using common authentication protocols (e.g., EAP-AKA) and signaling messages for handovers to ensure that devices can connect and roam freely between the two networks.
- Flexible Deployment and Cost Optimization: This component leverages the inherent strengths of each technology to optimize network performance and cost.
- Leveraging Wi-Fi for General Purpose Connectivity: Wi-Fi's ubiquity and lower cost per device make it ideal for non-critical, general-purpose connectivity. Devices like standard laptops, general-purpose IoT sensors, or visitor's mobile phones can be offloaded to the Wi-Fi network.
- Dedicated 5G for Mission-Critical Operations: The private 5G network provides the reliability, low latency, and security needed for mission-critical applications like the teleoperation of an STS crane. By using it only for these high-value applications, network resources are used efficiently.

- Hybrid Architecture: The "5G-in-a-box" approach creates a versatile, heterogeneous network that can be flexibly deployed. This allows the port to use the high-performance 5G network in specific operational areas (e.g., where cranes operate) and Wi-Fi in others (e.g., offices or less-critical zones), or even both simultaneously. This hybrid model ensures all connectivity needs are met in a cost-effective manner without compromising on performance for critical tasks.

A first version of the sub-enabler is under development. Initial testing will take place in lab environment in the 1st year. After this period, deployments in the field will follow in the 2nd year for conducting some preliminary experiments. In the 3rd year, final trials will take place.

2.6 Summary

This chapter has addressed the identified communication enablers: (1) Communication resource management, with five (5) sub-enablers: Zero-touch network and service management; Network programmability; Automated gNB configuration; Network performance monitoring and control; Identification and selection of backhauling. (2) Network slicing, with both possibilities of slicing implementation in the radio and core networks. (3) Integrated sensing and communication. (4) Public-private network integration, based on either SNPNs or PNI-NPNs. (5) Multiple-RAT connectivity, with three (3) sub-enablers: sidelink, Wi-Fi integration and NTN integration.

For each of the enablers (and its sub-enablers, if applicable) a general description was first given, followed by the analysis of its association with the AMAZING-6G use cases (i.e. how the enabler may empower the AMAZING-6G use cases). Depending on the used testbeds, different design and implementation plans were given for different use cases. Nevertheless, a high-level use-case/testbed independent design was given for each of the (sub-)enablers, for the sake of its potential applicability to verticals and use cases beyond those of AMAZING-6G.

3 Compute as a Service enablers

This section describes the Compute-as-a-Service (CaaS) enablers listed in Table 1-2. The CaaS enablers are defined to optimize the use of compute resources in the user-edge-cloud compute communication continuum as part of the overall 6G network. We start with the overarching enabler, i.e., compute resource management, which embeds principles of intelligent service and resource orchestration, intent-based network management for simplified processing of service requirements, and real-time monitoring that captures a wide variety of performance metrics as an essential input for intelligent orchestration. The overview further continues with the compute continuum, deep diving into specifics of the distributed compute resources within diverse Network Function Virtualization Infrastructure (NFVI) environments, and with the cloud-native service design, which shows the main principles behind virtualized services and applications in Beyond 5G and 6G, including the overview of application packages and their deployment options. Finally, we close the CaaS enablers with a specific enabler that focuses on creation of Kubernetes clusters on demand, as a process of dynamic network and compute management that stretches over distributed NFVIs.

3.1 Compute resource management

This section covers a broad technological enabler that embeds principles of closed-loop resource management, covering three essential phases: (i) intelligent service and resource orchestration in compute continuum (decision-making), (ii) intent-based processing (service requirement analysis and validation of decisions made by orchestrators), and (iii) real-time monitoring. This enabler is in charge of monitoring the NFV infrastructure within compute continuum (edge to cloud), making proactive decisions on compute and service scaling, instantiation, relocation/migration, and termination, based on the decisions made by the orchestration layer, to fulfil the service requirements stated in the intents. In the following sections, we describe the enabler providing a general overview, and then define the specifics of how this enabler is used and upgraded in the context of particular AMAZING-6G use cases.

3.1.1 Description of the enabler

Intelligent service and resource orchestration in extreme edge/edge/cloud compute continuum

The service and resource orchestration solutions are stemming from European Telecommunications Standards Institute (ETSI) Management and Orchestration (MANO) standardized framework, whereas now they are considered as part of the Zero-touch Network and Service Management paradigm. **It is important to note that the ZSM solution presented in Section 2.1 is entirely focused on automated and intelligent orchestration of communication resources, while this section dives deeper into the compute resource allocation, scaling, migration, and termination operations.**

The goal of the orchestration solutions is to perform resource and service allocation/instantiation, scaling, migration, and termination, in alignment with service requirements and real-time monitoring data. The intelligent orchestration solution embeds advanced algorithms for optimizing compute resource consumption in the 6G edge-cloud compute and communication continuum. To optimally orchestrate compute resources in an almost completely virtualized network infrastructure, it is essential to adopt intelligent functions/agents, such as those performing active learning about user mobility and network performance by using AI/ML techniques. This way, knowing the overall network performance and possibly user mobility, compute resources can be reallocated and deployments can be adjusted/migrated to improve overall network performance. The output from those functions/agents is deeply rooted into decision-making layers, which are considered as distributed service management logic running on the edge (edge-to-edge) and cloud infrastructure. Although recent research has demonstrated the potential of Artificial Intelligence (AI)/Machine Learning (ML) for orchestrating end-to-end networks [13] [14] [15] and services there are still various important challenges that need to be addressed. For instance, the complexity of orchestrating virtualized network functions across multiple

heterogeneous network slices and technological domains requires advanced AI/ML solutions that effectively manage diverse devices, services, and operational requirements. Additionally, the state-of-the-art management systems are usually centralized and as such they struggle with scalability, requiring decentralized AI/ML models to cope with heterogeneity and distributed nature of services over compute infrastructure without bottlenecks. Also, a general issue is the excessive energy consumption when deploying AI solutions, while aiming to create energy efficient and sustainable AI/ML-driven networks. The solution used in AMAZING-6G (Figure 3-1) embodies the distributed service management logic running in the UE-edge-cloud computing communication continuum, and AI/ML solutions which are carefully selected to be able to adapt to this distributed deployment fashion. The overall framework will be further enhanced by employing more advanced explainable AI/ML models for decision-making (self-healing characteristics of network and service management in 6G), and using Large Learning Models (LLMs) for intent translation and management (autonomous network and service management).

Focusing specifically on the RAN part of the end-to-end network, intelligent service and resource orchestration is achieved by dynamic placement of RAN functions based on both compute resource availability and real-time network demand. In this approach, xApps/rApps act as intelligent decision engines: they consume Key Performance Indicators (KPIs) related to user traffic throughput, and correlate these with compute telemetry from extreme edge, edge, and cloud infrastructure. When a spike in traffic is detected, the orchestration logic can trigger the deployment of Centralized Unit (CU)-User Plane (UP) instances on the most suitable server depending on the internal policy (e.g., satisfy Service Level Agreements (SLAs) while maximizing compute resource utilization). This approach ensures that the network dynamically adapts to demand by scaling user-plane capacity at the optimal location, rather than relying on static provisioning. The result is a programmable, resource-aware continuum where functions are elastically distributed across heterogeneous environments, improving both user experience and infrastructure efficiency.

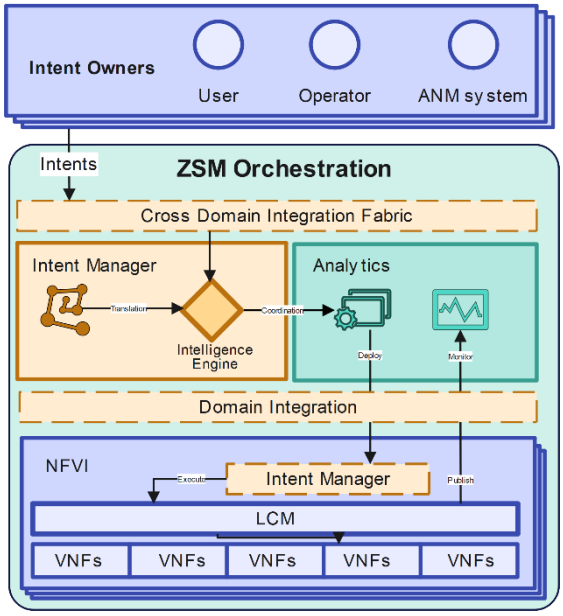

Figure 3-1 Compute and Service Management Framework embedding intent-based management principles and real-time monitoring analytics

Intent-based service and resource management

The intent-based service and resource management enabler provides a foundational component for autonomous network operation in a telecom environment. By integrating user-centric design, orchestration interfaces and intelligent feedback mechanisms, it supports scalable, agile and SLA-driven network management. It acts as an intermediate layer between user-defined objectives and

underlying orchestration and infrastructure layers. It decouples what the operator wants from how the system achieves it, by using AI, policy engines, QoS management APIs, and orchestration interfaces.

From the architectural point of view, the main components are:

- An Intent Interface Layer that sits on top of the architecture, which serves as the gateway for users or applications to express their service goals. Their intents can be submitted in various forms, including data via APIs, domain-specific languages or even natural languages.
- Intent understanding and parsing: Once the intent is received it is processed and translated into a structured internal representation. This classifies the intent, extracts parameters (e.g., slice type, target latency, 5G Quality of Service Identifier (5QI), and resolves ambiguities). These parameters are used as input for the logic of the Service and Resource Orchestration components, responsible to handle the lifecycle of end-to-end services and managing the placement of their functions in the continuum, to guarantee the fulfilment of the intent.
- Policy validation: Feasibility check and policy validation to verify available resources and SLAs. Ensure the requested action is allowed and achievable. Intent translation and decomposition: Once validated, the intents are broken down into executable subcomponents. This can result in new configuration templates, in deployment descriptors, Yet Another Markup Language (YAML) templates for slice deployment, Application Programming Interface (API) calls to orchestrators, Network managers, Quality of Service (QoS) subscriber managers, and Software Defined Networking (SDN) controllers. Execution layer responsible for actual interfaces with orchestration and management systems Actions such as creating network slices, configuring QoS profiles or provisioning edge resources are triggered here.

Assurance and monitoring collect telemetry from the RAN and core and evaluate whether the system behavior aligns with the intent. Figure 3-2 shows the high level architecture of the intent based system, where it shows the main blocks of the system, from the intent acquisition that handles the processing of the intent, by prompt, or Natural Language Processing (NLP), intent understanding which processes the intent, validates if it's a viable intent, the policy and goal orchestration module that verifies is the 5G system supports the intent, if there are available resources, and the execution and enforcement module that converts the goals in specific commands and handles the execution of those interfacing with orchestrators, controllers, management APIs. Finally, to guarantee the SLAs of the executed intents, it monitors and ensures that those intents were successfully applied and the network behavior meets the goal.

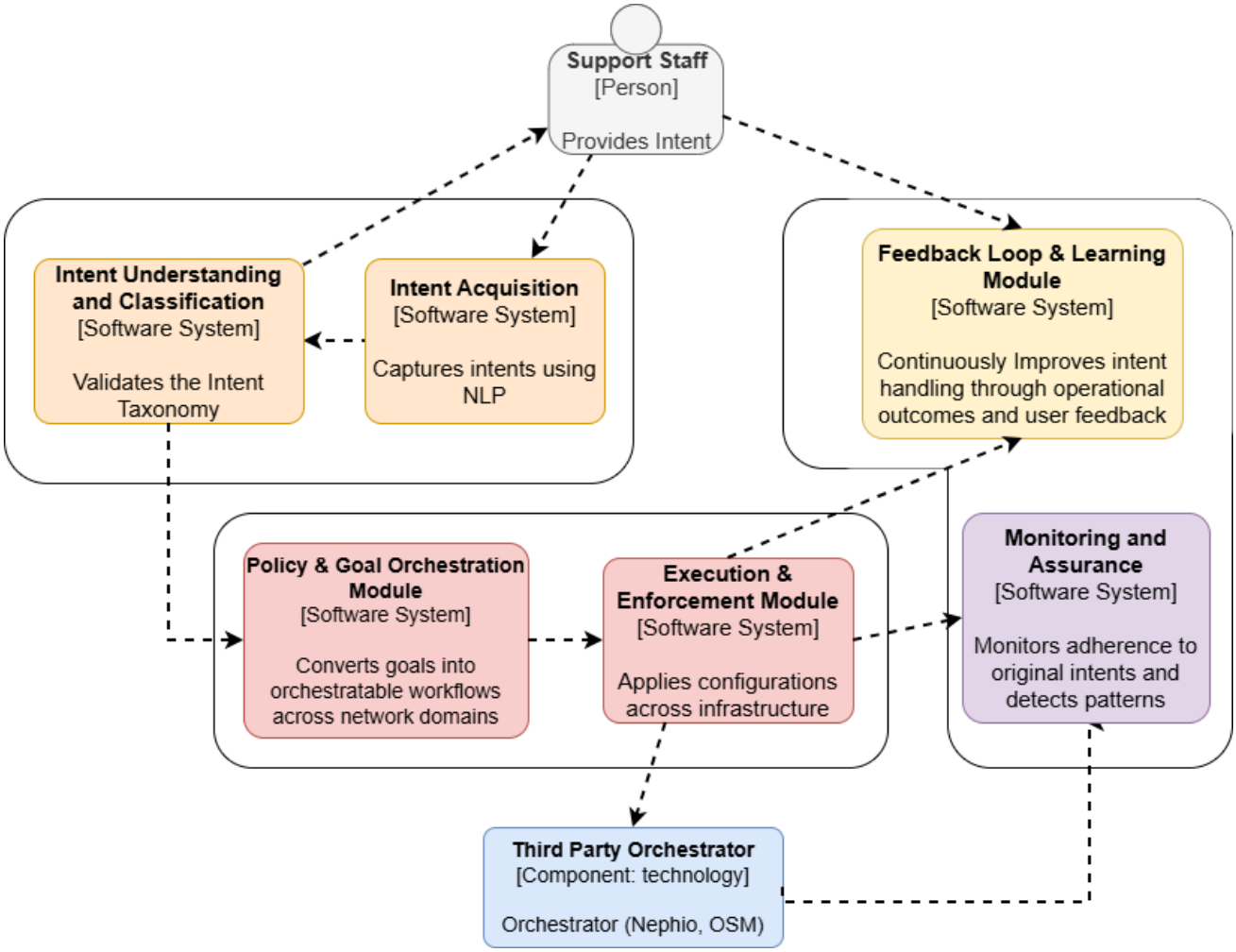

Figure 3-2 High Level Architecture of the Intent Based Service Resource Management

Within the ZSM framework, Intent-Based Networking (IBN) is a paradigm that takes automation a step further by moving it toward true autonomy [9]. Instead of prescribing how tasks should be executed, intents describe the required outcome in a high-level, machine-readable form. These intents are then enforced and monitored through closed-loop control, ensuring not only that the service is set up correctly at the outset but also that it adapts in real-time to maintain SLAs intact as network conditions change. Intents can be decomposed into sub-intents across RAN, core, edge, and cloud domains, facilitating the efficient orchestration of services and network resources across different domains.
As part of the ZSM architecture, the Intent Management Entity (IME) is a central component that bridges high-level business or application goals with the operational capabilities of the underlying network and service domains as shown in Figure 3-2. It captures, interprets, and refines intents into actionable policies and configurations. At the same time, it ensures that these intents remain consistent with domain-specific constraints and available resources. The IME coordinates with intent handlers across domains and manages conflicts when multiple intents interact. It also relies on closed-loop assurance to continuously validate outcomes against the declared objectives. In practice, this allows operators or applications to specify outcomes such as “maintain latency below 100ms”. The IME then orchestrates the necessary processes, optimizations, and adaptations so that the system autonomously delivers and sustains the required performance under changing conditions.

Finally, the integration of AI/ML techniques enhances intent management by supporting natural language interpretation, predictive analytics, and proactive assurance. For example, LLMs can help translate ambiguous human requests into formalized intents [10]. These features position IBN as a

fundamental component of ZSM for enabling networks to evolve from automated systems to fully self-managing infrastructures.

Real-time monitoring of compute resources and energy consumption in compute continuum

The Real-time monitoring of compute resources and energy consumption in the compute continuum is a key enabler that delivers a continuous, granular view of how compute resources and energy consumption are used from the edge to the cloud. Its main function is to collect accurate, real-time data in order to understand and optimize performance, reduce costs and minimize the environmental impact of computational activities, contributing to sustainability goals. Accurate measurement of energy consumption is essential, as it is impossible to improve what cannot be measured. The data collection process begins with the identification of the data sources, and proceeds with the definition of the parameters to be monitored, the methodology for their collection, and the data model to use in order to facilitate analysis. Data sources can include the application itself, external applications, and the underlying infrastructure.

In the context of real-time monitoring of the computing continuum, a truly valuable range of measurements is obtained, going from energy consumption to system performance and resource efficiency. On the energy perspective, the power in watts and energy in joules of edge and cloud devices, including CPUs, GPUs, network modules, monitors and sensors, are rigorously monitored, as well as overall cooling and lighting system. At the same time, CO_2 emissions as well as battery life on Internet of Things (IoT) and edge devices can be estimated. On the computational resources side, Central Processing Unit (CPU), Graphics Processing Unit (GPU), Random Access Memory (RAM) (used, free, buffer, cache and swap) and disk space usage percentages are analyzed for each physical or virtual machine. Finally, there are network traffic metrics (bytes and packets per second) and log collection, like anomalies and security events which add real value.

3.1.2 Use case association and contributing partners

Table 3-1 shows the association of the three sub-enablers and use cases with which these sub-enablers will be integrated, tested, and validated. In this section, therefore, we provide more information about the aforementioned integration and use-case specific context around these enablers.

Table 3-1 Use case association for Compute management enabler and sub-enablers.

	Intelligent service and resource orchestration in extreme edge/edge/cloud compute continuum	Intent-based service and resource management	Real-time monitoring of compute resources and energy consumption in compute continuum
P1	UPAT/PNET		
E1	NXW	CAPG	
T1	NXW, LINKS		NXW
T2	NXW, LINKS		NXW
T4	IMEC	IMEC	IMEC
T5	CERTH/ThPA		

The enabler is extremely important for the **P1 use case** as it enables extreme edge computing (on-site MEC servers, drones, or even private networks with portable base stations) to process possible Augmented Reality (AR)/Virtual Reality (VR) rendering and sensor fusion locally, while offloading only

heavy AI/ML tasks to central cloud when transport network is available, in order to ensure continuous Situational Awareness even in disrupted networks and seamless mobility of operatives across operators without service interruption. Especially for compute resources, when a disaster affects network and compute resources in a PPDR situation, operations may become unpredictable: new users, sensors, and drones may join or move in the private network with various applications but also between operators. Orchestration must dynamically allocate and migrate workloads (e.g., AR rendering engines, video analytics) between extreme Edge (on drones, or local Multi Access Edge Computing (MEC) servers), Edge (operator MECs at gNBs) and in transport network allows to cloud (central PPDR control centers or cross-operator slices). Especially in the case of AR/VR applications (e.g., real-time 3D situational awareness) require extreme edge computing (on-site MEC servers) must process most AR/VR rendering and sensor fusion locally, while offloading only heavy AI/ML tasks to central cloud.

The **E1 use case** on Renewable Energy Communities (REC) adopts a Service and a Resource Orchestrator to deploy the cloud-native application components of Energy Management Systems (EMS) for Smart Buildings and RECs in a computing continuum that involves cloud resources as well as devices and edge nodes available in the private network infrastructure of the buildings.

The **T1 use case** on "Protection of vulnerable road users" integrates a Resource Orchestrator to coordinate the deployment of containerized application components for roads monitoring, collection of data from vehicles and real-time assessment of risks for pedestrians and vulnerable road users. The orchestration logic distributes dynamically the tasks among Road Side Units (RSU) equipped with solar panels and edge nodes. The placement and task offloading decisions take input from the real-time monitoring, in order to jointly consider availability of computing resources, power consumption and RSU battery level. The procedures for application components and tasks migration guarantee service continuity.

The **T2 use case** on "Improving urban safety with Unmanned Ground Vehicle monitoring" implements the same enablers as in T1, but handling workloads that can be deployed in a computing continuum extended to UGV. The real-time monitoring collects data related to usage of computing resources, power consumption and battery level of the UGVs to feed placement and offloading decisions.

For the **T4 use case:** Intent-based networking will allow teleoperation services to express high-level requirements (e.g., maintaining ultra-low latency and reliable connectivity) as intents rather than low-level network configurations. These intents are processed by intent management entities, which decompose them into sub-intents for the RAN, edge, and core domains, ensuring resources are allocated and optimized automatically across heterogeneous infrastructures. By doing so, IBN enables zero-touch orchestration that continuously adapts to mobility and network dynamics, guaranteeing service continuity and QoS for mission-critical teleoperated vehicles. The intelligent compute orchestration will be performed to ensure optimal placement of T4 anomaly detection service across edge-cloud continuum (stretching over UE compute units all the way towards cloud), which is used as a trigger for switching between teleoperation and automation modes. The proactive placement and lifecycle management decisions will be generated based on the real-time and historical monitoring data, which will be used as input for ZSM decision-making engine.

The **T5 use case** targets optimization of port logistics and transport operations through efficient orchestration of compute resources across edge and cloud. The scenario requires ultra-low latency and high reliability to support tele-operation, real-time analytics, and safety-critical port processes. A containerized, cloud-native architecture enables dynamic deployment and scaling of services for predictive analytics, optimization, and IoT data processing. The Compute-as-a-Service (CaaS) enabler provides automated orchestration, resource allocation, and lifecycle management across heterogeneous infrastructure, ensuring that latency-sensitive and compute-intensive tasks such as crane scheduling, AGV coordination, and workload balancing can run efficiently and securely. Continuous integration and CI/CD pipelines facilitate agile deployment of AI and analytics components while maintaining operational resilience and energy efficiency. The orchestration capabilities described

here provide the computational foundation upon which the Digital Twin platform (detailed in Section 4.2) operates.

3.1.3 Design, development and implementation

High-level design of intelligent service and resource orchestration in extreme edge/edge/cloud compute continuum: The intelligent service and resource orchestration solution operates across an extreme edge/edge/cloud compute continuum, embedding advanced algorithms to optimize resource consumption and service performance through continuous, closed-loop adaptation. Its high-level design integrates three main functional layers: the **infrastructure layer**, encompassing distributed compute and communication resources such as RSUs, OBUs, and cloud nodes; the **orchestration and management layer**, which automates service deployment, scaling, and migration through zero-touch and intent-based mechanisms; and the **intelligence layer**, which hosts AI/ML-driven network functions that actively learn from real-time telemetry, including user mobility, CPU and memory utilization, and end-to-end latency. These intelligent functions use reinforcement learning techniques, such as Deep Q-Networks, to refine decision-making policies and dynamically select optimal resources under changing conditions. Through open interfaces and closed-loop control, the orchestration continuously aligns network behavior with service-level goals, enabling adaptive, self-optimizing management. Figure 3-12 shows the integration of three functional layers in the ZSM Framework: (i) AI/ML functions for interpretation, decision-making, and cognition; (ii) ZSM functions enabling closed-loop orchestration and intent-based control; and (iii) NFVI handlers managing deployment and virtualization infrastructure. These layers collectively support adaptive, self-optimizing management across the edge–cloud continuum.

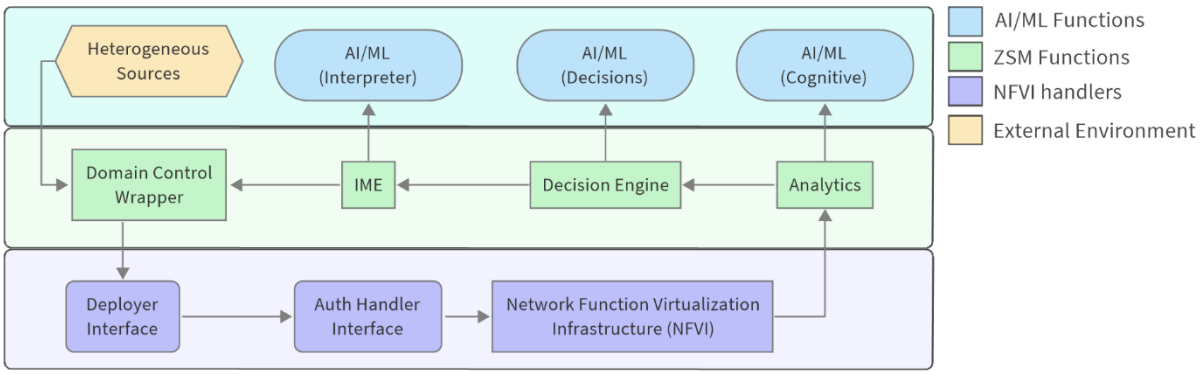

Figure 3-3 High-level architecture of the intelligent service and resource orchestration framework

High-level design of Intent-based service and resource management: The intent-based service and resource management enabler provides the automation layer that links high-level service objectives with the underlying B5G and edge infrastructure. Operators and applications will be able to express requirements as intents, and these intents are processed, validated, and translated into network templates, resource allocation policies. The enabler interacts with orchestration systems to enforce the requested configurations and continuously monitors telemetry from the RAN and UE datasets to ensure SLA compliance. This enables adaptive, optimization of connectivity and compute resources, ensuring that the smart building environment remains reliable, efficient, and aligned with evolving operational demands.

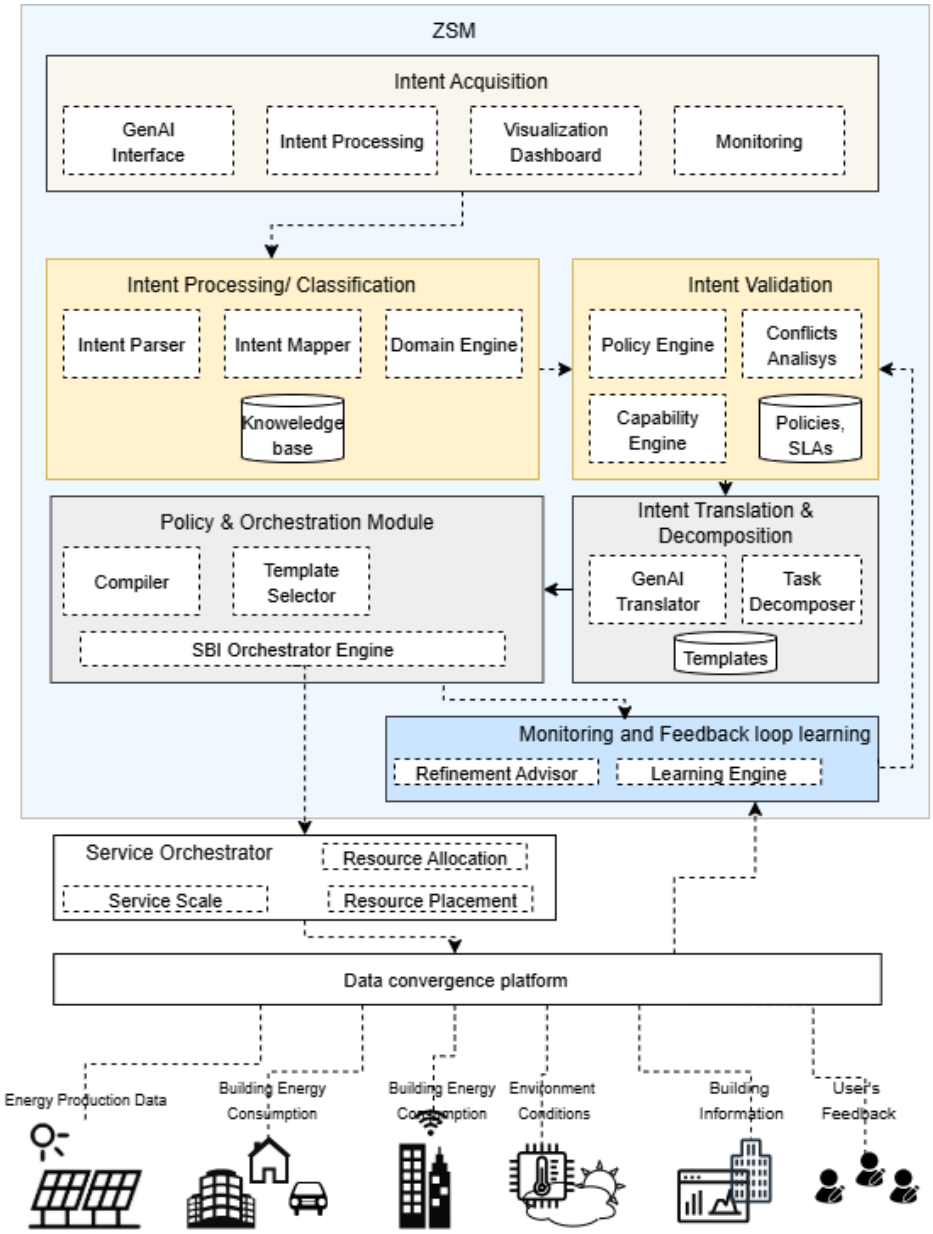

Figure 3-4 High-level design of Intent-based service and resource management

High-level design of Real-time monitoring of compute resources and energy consumption in compute continuum: The Real-Time Monitoring System (Figure 3-5) integrates programmable and configurable plugins to collect monitoring metrics on usage of computing resources and related energy consumption from different sources, both physical and virtual, along the continuum. These sources may include edge compute nodes, devices in the extreme edge, and IoT platforms. Metrics at both physical and virtual level can be collected, e.g., resource usage on a physical computing node, or related to a single Virtual Machine (VM) or container. Metrics collection can be regulated via APIs for monitoring jobs configuration, to adapt type, frequency, and granularity of monitored data to the real-time requirements of the consumers, e.g., to feed decisions on automatic orchestration, service recovery, or application level tasks. Data consumers will be able to retrieve both historical data sets and real-time data streams, as well as subscribe to receive notifications or alerts on programmable patterns and events.

Real-time monitoring of compute resources and energy consumption in compute continuum is fundamental for Compute Resource Management, providing essential data for service orchestration and management. Collected information allows for smart and proactive decisions on load allocation, scalability and cost optimization, as well as preventing operational problems. Continuous monitoring of

CPU, memory and network traffic allows, for example, AI jobs to be balanced across different GPUs, avoiding any bottlenecks and maximizing efficiency. An additional example is provided by the use of AI for the analysis of metrics such as temperature, CPU usage and error logs. This enables the prediction of potential anomalies, with a view to extending the life cycle of hardware. In short, the enabler is also crucial for assessing the performance and sustainability of the continuum, comparing usage indicators with pre-established requirements and supporting predictive maintenance and fault prevention to extend the life of the hardware. In particular, it is worth noting the possibility of selecting the deployment (edge or cloud) in a targeted manner based on latency, computational power and availability of renewable energy, as well as the contribution to security and privacy, thanks to the detection of anomalous activity and integration with zero trust and federated learning strategies within AMAZING-6G.

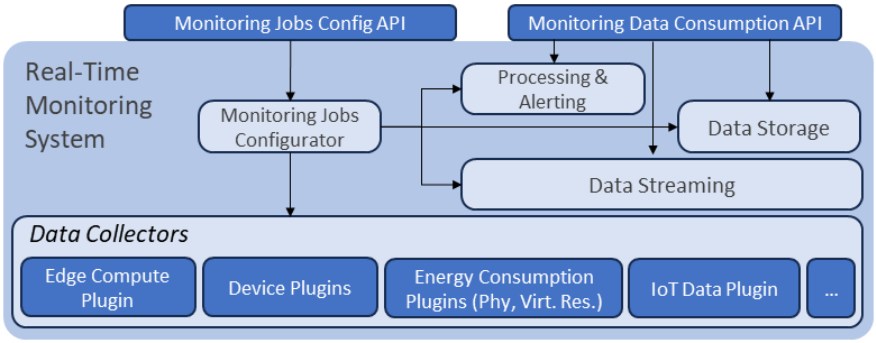

Figure 3-5 Real-time monitoring - high-level design

Design, development and implementation for the P1 use case: For the P1 use case, Patras5G testbed will be able to deploy on demand 5G networks. To this end, a K8S cluster will be created on demand on which the 5G Core be deployed and with the use of previously described enablers the deployed gNBs will be connected to, providing a fully operation 5G network. The design will be available with the P1 use case design (Q1 – 2026) and the deployed solution will be integrated by Q1 2027.

Design, development and implementation for the E1 use case: The vertical application targeted in E1 consists of a Smart Building Energy Management System (SB-EMS), designed as a cloud-native service composed of a set of containers that can be deployed over Kubernetes (K8S) environments. The containers may be instantiated in the public cloud. However, in order to better exploit the computing capabilities of the private network infrastructure and guarantee the confidentiality and security of the users data, the proposed solution is based on a distributed, per-building deployment in local K8S clusters at the smart buildings' edge nodes. As option, some application modules can also be deployed over extreme edge nodes or users devices (e.g., Customer Premise Equipment (CPE), IoT gateways) equipped with computing resources.

The SB-EMS application integrates functions with different types of computing and network requirements. For example, monitoring functions for aggregation and storage of data from the IoT platforms need mainly disk resources, while the monitoring collectors should be placed closer to IoT gateways (or if possible co-located with them) to reduce the network traffic. Training of AI/ML models needs GPUs, while nodes equipped with only CPUs can be suitable for decision making processes based on the outputs of AI inference tasks. For this reason, it is crucial that the resource allocation and placement logic jointly considers the nature of the processing tasks to be offloaded, the capabilities and the characteristics of the computing nodes in the device-extreme edge-edge continuum, and the impact of the data flows on the mobile network connectivity.

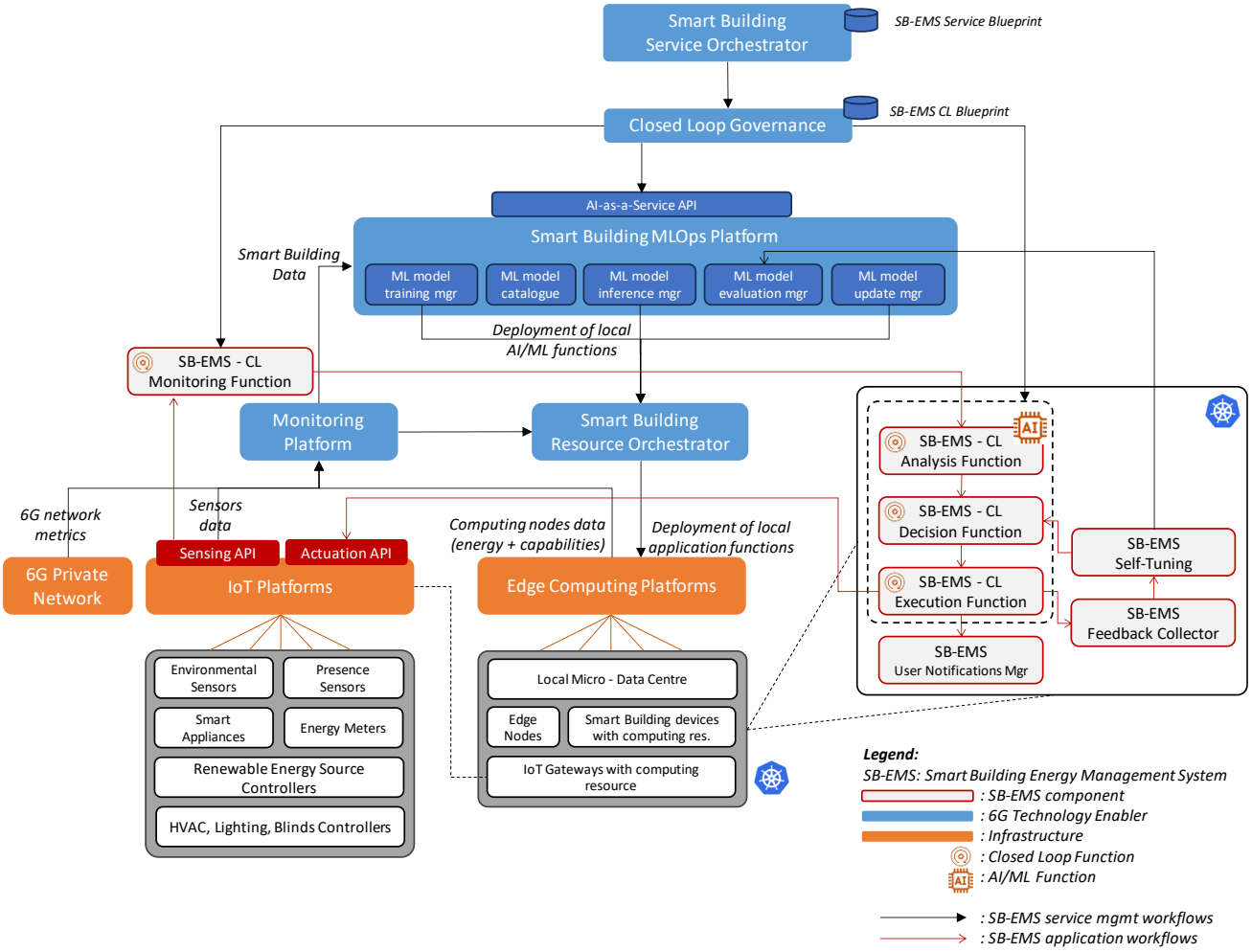

Figure 3-6 Service and Resource Orchestration solution for E1

The orchestration solution designed for the implementation of E1 integrates a **Service Orchestrator** and a **Resource Orchestrator**, as shown in Figure 3-6. They are both deployed locally and they are specialized for handling resources within the smart building environment, in combination with the IoT platforms and the 6G private network deployed there. The Service Orchestrator provides the logic for the management workflows of the SB-EMS service, defined through a service blueprint.

Since part of the SB-EMS is based on a Closed Loop (CL) working at the application layer (SB-EMS – CL), the Service Orchestrator interacts with a **Closed Loop Governance** component that is in charge of provisioning and managing the related CL functions. The interaction with the **Smart Building MLOps Platform** (see section 4.1 for further details) allows to select and deploy the most suitable ML models for the inference tasks of the CL analysis functions, e.g., for prediction of energy consumption and production, rooms occupancy or related comfort preferences. If no models are available, a training task is triggered. This may happen also at runtime in case of drifts or model tuning to adapt to new users' preferences.

Service Orchestrator, CL Governance and Machine Learning Operations (MLOps) Platform use the mediation of the **Resource Orchestrator** to deploy their managed workloads on the computing nodes. The Resource Orchestrator identifies the optimal set of target nodes in the continuum taking into account the nature and the requirements of the various workloads (as specified by its invokers), the resources available in the infrastructure, the characteristics of the nodes, as well as their mobile connectivity. The Resource Orchestrator integrates the modelling of devices and extreme edge nodes, also capturing the relationship with the IoT platforms and their sensors and actuators. This approach allows to take decisions on the placement also considering the optimization of the data flows at the

application level, e.g., for the collection of data from the smart building sensors, to their storage and processing at training and inference time.

In the deployment plan for E1, Service and Resource Orchestrator will be deployed and integrated with the edge computing platforms in ORO testbed during Q4-2025, while the deployment of Closed Loop Governance and MLOps Platform (currently still under implementation) will be performed in Q1-2026. The Monitoring Platform will be integrated with the IoT platform in Q2-2026, with an initial version of the SB-EMS application and closed loop functions deployed in Q3-2026 for tuning and testing. The second version of the integrated system is planned for the end of 2026, to feed more extensive trials in 2027.

Design, development and implementation for the T1 and T2 use cases: The Service and Resource Orchestration solution designed for T1 and T2 use cases is shown in Figure 3-7.

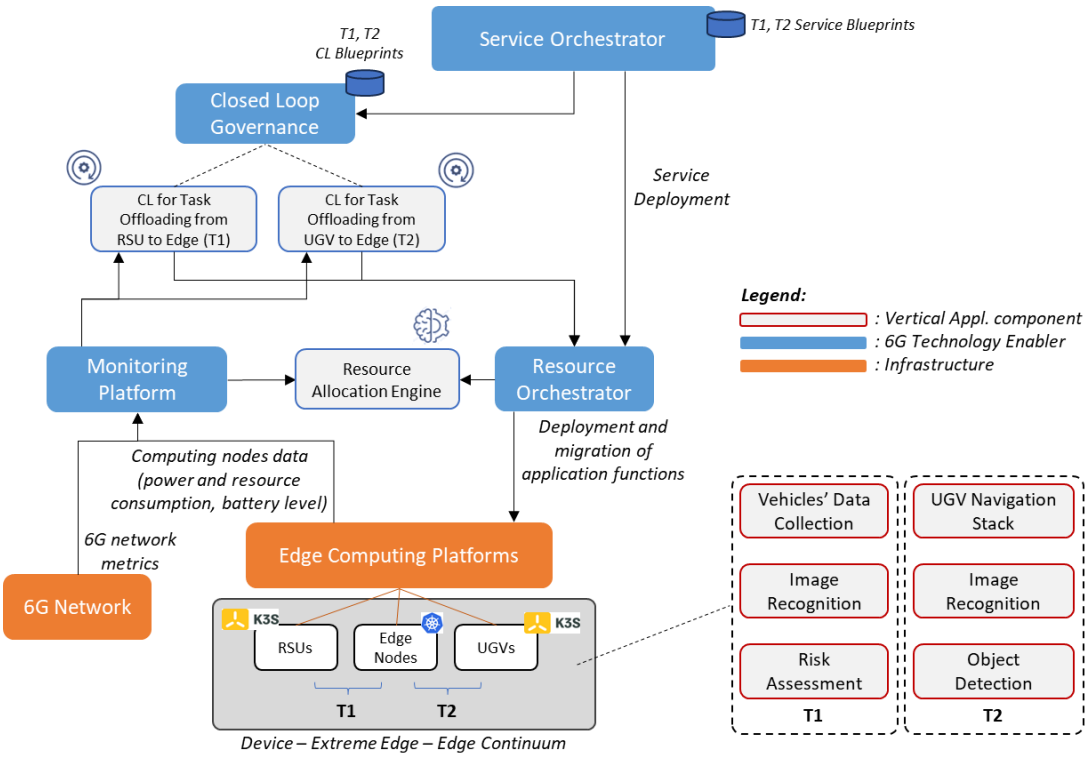

Figure 3-7 Service and Resource Orchestration solution for T1 and T2 use cases

The lifecycle management of the end-to-end services for T1 and T2 is handled through the **Service Orchestrator**. Both services are defined through service blueprints that describe their application components, related computing and networking requirements, as well as references to the management CL blueprints defining the rules for their automation at runtime, e.g., in terms of scaling, migration, or tasks offloading. As for E1, the management of the CL functions[1] is delegated to the **Closed Loop Governance** component. In T1 and T2 the CL analysis and decision logic is based on static and pre-defined rules, without involving any AI/ML algorithm. For this reason, no interaction with an MLOps platform is foreseen.

T1 and T2 services are designed as cloud-native applications with containers that can be deployed on devices, extreme edge or edge nodes. In detail, T1 includes components for the collection of data generated by the vehicles, for image recognition, and for the assessment of the risks for pedestrians and vulnerable road users. T2 also includes image recognition, but associated to object detection, and

[1] It should be highlighted that in T1 and T2 use cases the CLs are applied to the service management logic, while in E1 use case the CLs are part of the vertical application.

integrates an UGV navigation stack for the movement of the specific device. The closed loops for their management at runtime are partially generic (e.g., for scaling of resources in case of increased load) and partially specialized for their own applications and for the constraints of their devices. In case of T1, tasks can be offloaded from the RSU to the edge, depending on the RSU processing load (which in turns depends on the amount of data collected by the vehicles) and its battery level (which is also impacted by power consumption and energy production at the solar panel). For T2, task offloading is from the UGV to the edge, following a similar logic but also taking into account the power consumption due to the UGV movement and the quality of the network connectivity depending on its location.

Service deployment is actually performed by the **Resource Orchestrator**, under the trigger of the Service Orchestrator. The Resource Orchestrator handles a set of heterogeneous edge platforms controlling computing resources over edge nodes for both T1 and T2, RSUs for T1, and UGVs for T2. Kubernetes is used for edge nodes, while K3S is used for RSUs and UGVs. The resource allocation logic for service provisioning is handled through a **Resource Allocation Engine**. Its algorithms take as input static information about the capabilities of the nodes (CPUs/GPUs, memory, disk), as well as real-time monitoring data related to mobile network metrics, edge resource consumption, power consumption, battery level for RSUs and UGVs, and energy production from the related solar panels.

Real-time monitoring data are collected through a **Monitoring Platform**, with dynamic and programmable monitoring jobs to retrieve data from 6G network and edge or extreme edge nodes and devices. In T1 and T2 the monitoring platform is specialized with data collectors for energy and power consumption metrics, with a configurable granularity up to per-container and per-process scope. This approach allows to take more accurate offloading decisions, with the possibility to predict the power consumption on the basis of the active processes running in the diverse node.

The Monitoring Platform proposed for E1, T1, and T2 is developed by Nextworks and it consists of an open-source, modular framework for collecting, processing, and delivering unified time-series from various data sources (see Figure 3-8). To achieve this process, two layer interacts each other. A management layer that through REST APIs or GUI provides programmatic access to lifecycle management of data-source plugins, such as starting/stopping collection, configure frequencies and additional attributes useful for data correlation. The second layer, identified as adaptation layer focuses on the actual data collection and ingestion. Since the solution is highly modular, this layer includes many ad-hoc plugins that could be deployed to pull metrics from different target data sources (e.g., via Representational State Transfer (REST) or Message Queuing Telemetry Transport (MQTT) protocols, from 6G network monitoring systems, computing platforms, IoT platforms, proprietary management APIs of the devices, or Power Distribution Units - PDU). The collected data are then transformed into the Influx Line Protocol, a time-series tag-value based standard format.

The Monitoring Platform is configured for computational resource monitoring in the device-to-edge continuum. CPU, memory, network and disk I/O consumption are gathered from virtual and bare-metal environments using custom agents. Energy metrics are retrieved from servers' smart plugs, rack PDUs or using Scaphandre and Kepler tools for per-container/per-process power consumption. The collected values, once unified, are streamed out across a Kafka bus for real-time usage and stored in an InfluxDB timeseries database, feeding the CL analysis and decision functions.

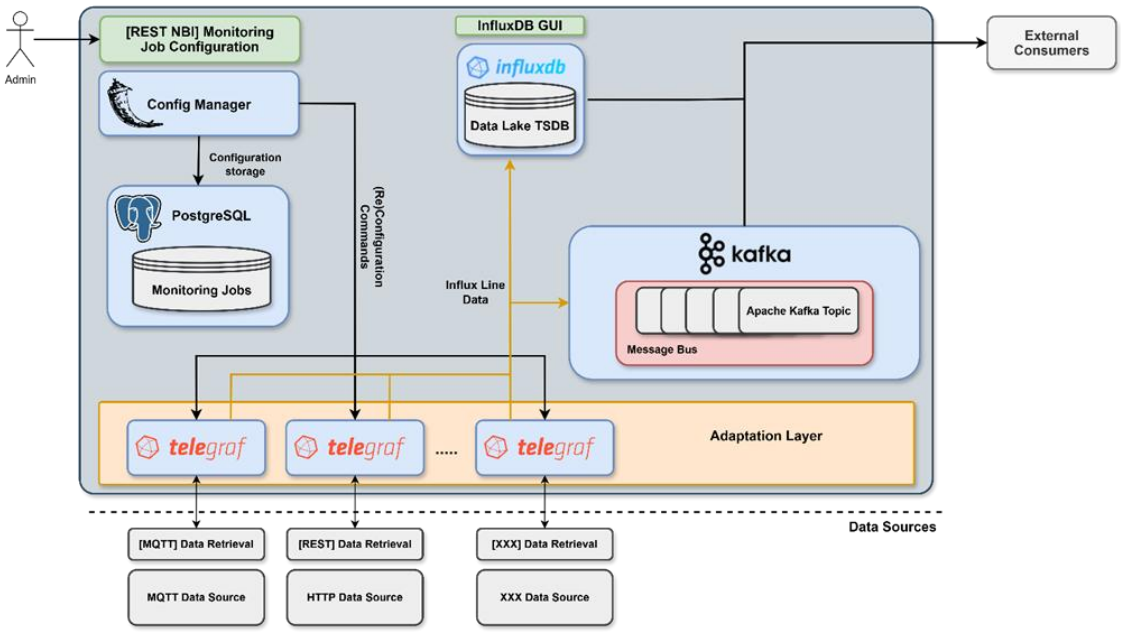

Figure 3-8 Internal design of monitoring platform for E1, T1, and T2 use cases

The deployment of Service Orchestrator, Resource Orchestrator and CL Governance is planned for the beginning of 2026, while the integration of the Monitoring Platform and the support for the collection of energy metrics is planned for Q2-2026, together with an initial version of the Resource Allocation Engine.

Design, development and implementation for the T4 use case: The proposed solution will be deployed through the ZSM framework, which continuously ingests exposed metrics from the 5G infrastructure via standardized interfaces such as CAMARA to monitor vehicular services in real time. These metrics, combined with multimodal data sources (e.g., network performance, historic usage, anomaly reports), feed predictive analytics modules that allow the orchestrator to proactively optimize resources like bandwidth and computing power before QoS degradation occurs. By translating service requirements into intents, the framework autonomously enforces energy-aware and latency-sensitive orchestration decisions across RAN, edge, and core domains, ensuring resilient and adaptive support for teleoperation and Vehicle-to-Everything (V2X)-assisted driving under dynamic conditions.

The preliminary tests are ongoing in IMEC's testbed environment, while the full integration with the T4 use case will be finalized during the second year of the project.

Design, development and implementation for the T5 use case: The Compute-as-a-Service (CaaS) enablers development for Use Case T5 follows a modular, containerized architecture to support distributed execution across the whole continuum. The architecture consists of containerized microservices hosting ML/AI models for Digital Twin analytics, containerized MQTT brokers and subscribers for IoT data ingestion and AnyLogic simulations deployed in Docker containers. These components are orchestrated using Kubernetes, which manages deployment, scaling, and lifecycle operations. The orchestration layer also integrates resource monitoring and auto-scaling mechanisms to dynamically allocate CPU and memory resources, ensuring performance and energy efficiency. Security is embedded through authenticated APIs and MQTT topics, while CI/CD pipelines (e.g., GitLab) enable automated deployments and continuous integration of new or updated models and services.

The implementation plan is structured in four phases. The design phase defines the architecture, interfaces, and security mechanisms. The development phase focuses on containerizing ML/AI models, AnyLogic simulation services and IoT components, and configuring orchestration and CI/CD pipelines. The integration phase validates orchestration across edge and cloud environments, resource management, and security features. Finally, the deployment phase involves rolling out the complete

CaaS-enabled architecture in the port logistics pilot, where it will be tested under real operational conditions and eventually validated against KPIs such as latency, reliability, and energy efficiency.

The design and integration activities will be completed in Years 1 and 2 respectively. Year 1 will focus on architecture design and containerization, while Year 2 will cover orchestration setup, CI/CD implementation, and integration testing across edge and cloud environments. An initial rough implementation and partial deployment will take place in Year 2 (M19–M23) to enable early pilot trials and gather feedback under real conditions. The full and complete deployment of the CaaS enablers is scheduled for Year 3 (M29–M33), following refinements and integration improvements based on Year 2 pilot results.

3.2 Compute continuum

This section presents insights into Compute continuum as a CaaS enabler, as well as its two associated sub-enablers (Distributed user-edge-cloud compute continuum and Multi-party cloud). The aforementioned sub-enablers are delivering edge and cloud compute resources in a distributed fashion, thereby creating a pool of compute resources that can be used for deploying and managing vertical services associated with the AMAZING-6G use cases.

3.2.1 Description of the enabler

Distributed user-edge-cloud compute continuum

The implementation of a ZSM framework within a distributed user-edge-cloud compute continuum contribute to overcome critical teleoperation bottlenecks such as latency spikes, resource allocation delays, and service disruptions that threaten the safety of VRUs. It ensures consistently low end-to-end latency for vehicular services by using closed-loop controls that continuously monitor key KPIs and trigger immediate resource redistribution when thresholds are breached. For teleoperation, this translates into resilient service delivery, where the framework dynamically provisions resources closer to the vehicle, migrates functions ahead of mobility events, and self-heals disruptions to preserve control continuity.

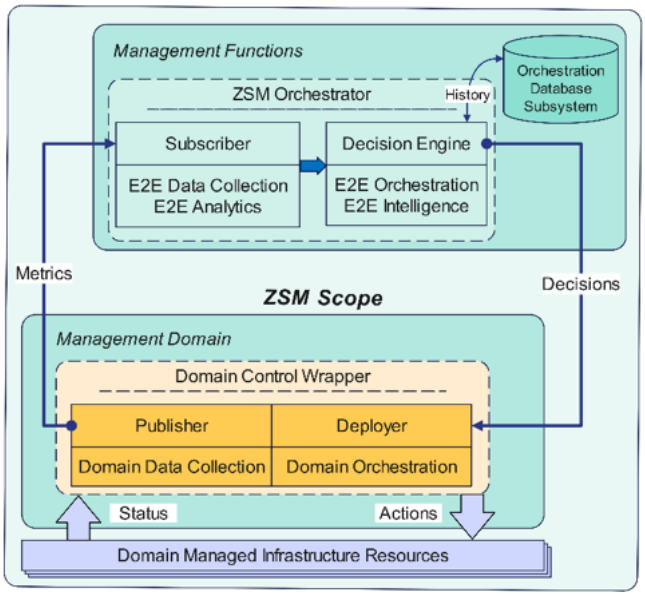

Figure 3-9 ZSM Framework architecture and sub-enablers

Figure 3-9 shows the architecture of the ZSM Framework, which is modular and designed to support heterogeneous infrastructures by automating the onboarding of new or reconfigurable NFVIs. Automated discovery, registration, and integration ensure that additional resources can join the orchestration workflow with minimal manual effort.

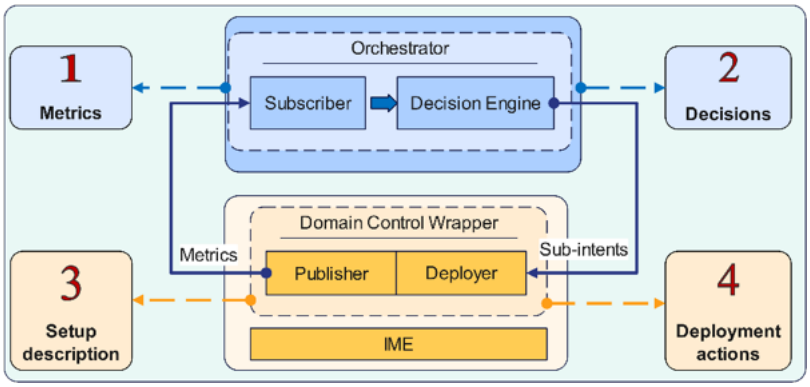

Figure 3-10 Data Management in the ZSM Framework

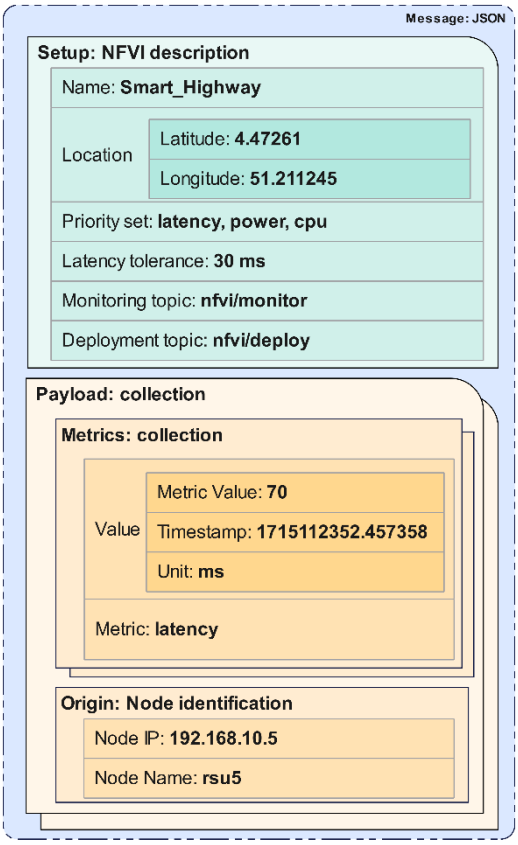

Figure 3-11 Message format for cross-domain data exchange in the ZSM Framework

The Data used during ZSM Orchestration is structured into four main categories as seen in Figure 3-10 **metrics datasets** (e.g., CPU percentages, power consumption in watts, E2E latency measurements in milliseconds) as seen in more detail in Figure 3-11, **decision datasets** that capture orchestration outputs and reasoning, **setup descriptions** that encode NFVI configurations and decision priorities, and **deployment actions** that translate orchestration decisions into executable templates such as YAML descriptors for Kubernetes or direct API calls to NFV MANO systems. Multi-criteria optimization leverages these datasets to enable the decision engine to concurrently analyse KPIs originating from different sources such as heterogeneous NFVIs. Since these KPIs often have different priorities (e.g. minimizing latency for safety-critical services while controlling power consumption for energy efficiency), the use of multi-criteria techniques allows a single decision engine to process and balance them simultaneously, ensuring coherent and efficient orchestration across domains.

Multi-party cloud

Multi-party cloud refers to a distributed cloud computing paradigm where multiple stakeholders, such as service providers, enterprises, and end users, collaborate and share resources across federated cloud infrastructures. This enabler supports the vision of a compute continuum by enabling seamless data processing and service delivery across heterogeneous and geographically dispersed environments. In a multi-party cloud setup, trust, interoperability, and dynamic resource allocation are key challenges. The architecture typically involves mechanisms for secure data exchange, joint orchestration, and policy enforcement across domains. It enables collaborative applications, supports data sovereignty requirements, and facilitates edge-cloud integration for latency-sensitive services. Multi-party cloud is particularly relevant in scenarios where data ownership, regulatory compliance, and cross-domain service composition are critical. It contributes to the overall flexibility, scalability, and resilience of the compute continuum envisioned in 6G systems.

3.2.2 Use case association and contributing partners

Table 3-2 Use case association and contributing partners

	Distributed user-edge-cloud compute continuum	Multi-party cloud
H1	TNO	TNO
H2	TNO	TNO
E1	NXW	
E2	TNO	TNO
T1	NXW, LINKS	
T2	NXW, LINKS	
T4	IMEC	

The Distributed User-Edge-Cloud Compute Continuum enabler applies to the **H1 and H2 use cases** by enabling low-latency processing of health data at the edge (e.g., near the patient), while allowing scalable analytics and storage in the cloud, ensuring timely detection and response to medical events. The Multi-party Cloud enabler supports use cases H1 and H2 by enabling secure and compliant data exchange between medical devices, healthcare providers, and cloud-based analytics platforms, while respecting data sovereignty and privacy requirements.

In the **E1 use case**, the computing continuum involves the Smart Buildings IoT platforms, devices and extreme edge nodes as well as edge platforms deployed in the smart buildings, up to cloud resources used to run applications at the REC level.

The Distributed User-Edge-Cloud Compute Continuum enabler also applies to the **E2 use case**, as drone-based wind blade inspections require real-time data processing, low-latency decision-making, and scalable analytics. By distributing compute resources across the drone (user), edge nodes near the inspection site, and centralized cloud platforms, this enabler ensures efficient task execution, minimizes latency, and supports dynamic workload allocation. It enables seamless coordination between local and remote processing, which is essential for mission-critical operations in E2.

In the **T1 use case**, the application for the protection of vulnerable road users runs over distributed edge and extreme edge nodes, in particular using Road Side Units (RSU) with computing capabilities and equipped with batteries and solar panels.

In the **T2 use case**, the application for urban video surveillance is distributed among edge nodes and 6G-connected UGV devices equipped with sensors, cameras, and computing resources. The application is split into containers for several tasks, including control of UGV navigation, image recognition and object detection. Tasks with lower latency constraints can be offloaded to the edge nodes, based on the battery level.

In the **T4 use case**, the enabler provided by the ZSM Framework is needed for this use case to ensure proactive orchestration of vehicular services across the UE–edge–cloud continuum. By leveraging predictive analytics and intent-based management, the framework ensures that computing and network resources are dynamically adjusted to maintain ultra-low latency and service continuity during the handover from autonomous to tele-operated driving. This guarantees safe and resilient operation under challenging conditions such as adverse weather, complex traffic, or construction zones.

3.2.3 Design, development and implementation

High-level design of Distributed user-edge-cloud compute continuum: The ZSM framework relies on a closed loop orchestration process across the user, edge, and cloud continuum to address the challenges of dynamic service delivery in distributed compute and communication environments. This framework operates as a closed loop orchestration system that continuously monitors network and compute conditions, derives orchestration decisions, and enforces them through control mechanisms on the underlying NFVI. **Monitoring** ingests heterogeneous telemetry and performance data such as latency, CPU load, and energy consumption. A **multi-criteria decision engine** combining analytical models, policies, and intent translation evaluates these KPIs and produces actionable strategies, while **control mechanisms** implement those strategies across target NFVI, coordinating their execution and verifying outcomes. This is done through *inner control loops* within the management domains that validate task completion and detect failures or interruptions.

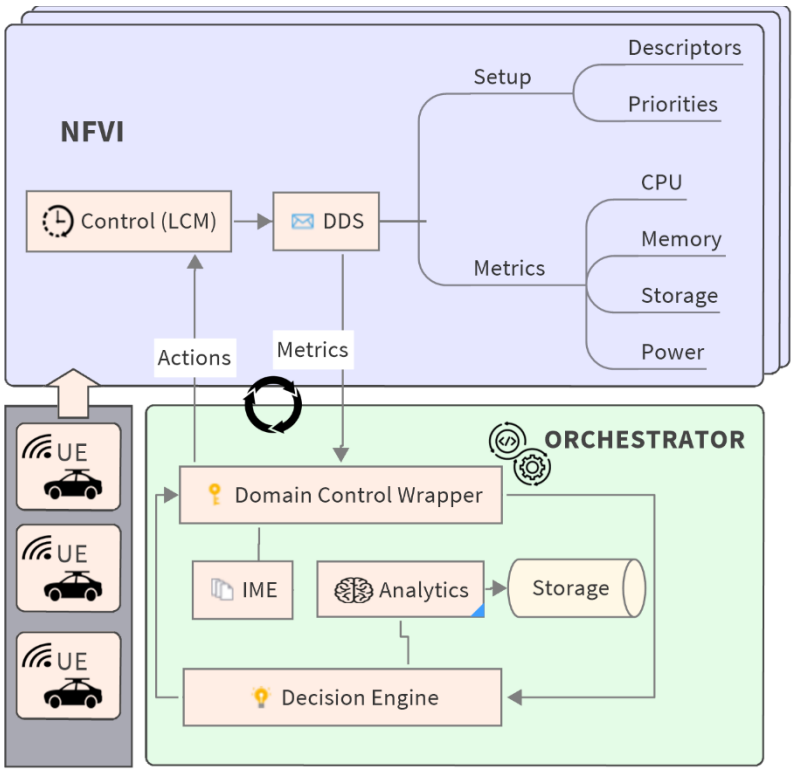

Figure 3-12 High-level architecture of the ZSM Framework working on top of the distributed user-edge-cloud compute continuum (distributed NFVI)

Through a dedicated abstraction layer (Domain Control Wrapper), the framework automates the registration of NFVI. In this way, a modular onboarding allows infrastructures from diverse providers to

be dynamically incorporated into the workflow with minimal manual intervention while maintaining interoperability. Together, these capabilities establish a technology-agnostic, intent-driven orchestration for autonomous service management across distributed administrative domains.

High-level design of Multi-party cloud: The Multi-party Cloud enabler supports a federated and collaborative cloud computing model, where multiple stakeholders, such as service providers, enterprises, and infrastructure owners, can securely share and orchestrate resources across distributed cloud domains. This design is essential for enabling a compute continuum that spans across administrative boundaries while preserving data sovereignty, trust, and interoperability.

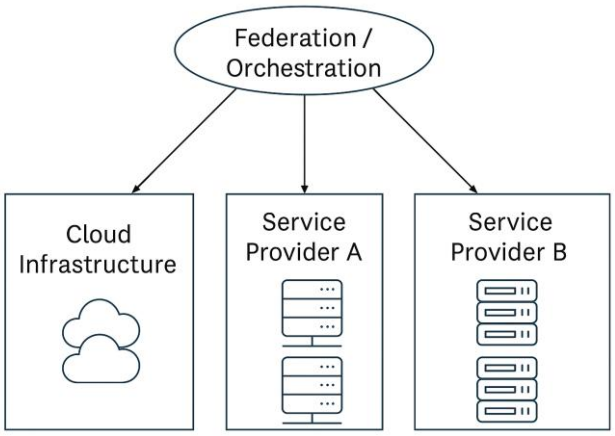

Figure 3-13 Multi-party cloud design

The architecture consists of multiple cloud domains interconnected through secure interfaces and governed by shared policies. Each domain retains control over its own resources while participating in a broader federation. A common orchestration layer enables dynamic workload placement, service discovery, and policy enforcement across domains. Trust management, identity federation, and secure data exchange mechanisms are integral to the design, ensuring that collaboration does not compromise privacy or compliance.

This enabler is foundational for scenarios requiring cross-domain service composition, distributed AI/ML workflows, and collaborative data processing, especially in regulated or multi-tenant environments.

Design, development and implementation for the H1 and H2 use cases: In use cases H1 and H2, medical devices continuously monitor patient health and transmit data for analysis and intervention. The Distributed User-Edge-Cloud Compute Continuum enables efficient processing of this data by distributing computational tasks across three layers:

- User Layer: The medical device (e.g., patch or pacemaker) performs initial sensing and basic signal processing.
- Edge Layer: A nearby gateway or hospital edge node processes data locally for real-time anomaly detection and alerts.
- Cloud Layer: Centralized hospital systems or cloud platforms perform long-term analytics, historical data correlation, and predictive modeling.

The implementation begins with identifying the performance and privacy requirements of medical devices. Next, the architecture is designed to distribute processing tasks across the device, edge, and cloud layers. Prototypes are developed to test edge analytics and cloud coordination. Integration focuses on ensuring secure data flow and low-latency response. Finally, the system is deployed in clinical settings and monitored to optimize compute distribution and reliability. The first phase of the enabler implementation is planned till Q1 2026, with the aim of first use case integration till Q2 2026. Based on the outcome of the integration, some finetuning of the implementation may take place after Q2 2026.

Design, development and implementation for the E1 use case: In the E1 use case, IoT platforms for smart buildings includes sensors like environmental and presence sensors, actuators for Heating, Ventilation, and Air Conditioning (HVAC), lighting and blinds control, and smart appliances. Moreover, the buildings are equipped with other elements for energy monitoring and management, like energy meters to measure the power consumption of specific devices or the controllers of renewable energy sources, like solar panels. Extreme edge nodes may include devices with controllable computing capabilities, like IoT gateways, smart cameras, video clients, etc. Moreover edge nodes like NUCs or local micro datacentres can be available. In E1 scenario, these elements are managed through Kubernetes or, for smaller devices, K3S platforms. On top of these edge platforms, the Smart Building Resource Orchestrator is in charge of deploying the applications on the various nodes, as previously described in Section 3.1.3. Edge platforms and Resource Orchestrator will be deployed in Q4-2025, while the integration with Monitoring Platform and IoT platform is planned for Q2-2026.

Design, development and implementation for the E2 use case: In the E2 use case, drones perform high-resolution inspections of wind turbine blades, generating large volumes of data that require timely processing. The compute continuum enables efficient distribution of tasks across three layers:

- User Layer: The drone performs initial data capture and basic preprocessing (e.g., compression, filtering).
- Edge Layer: A nearby edge node (e.g., at the wind farm or local control center) processes video streams and performs real-time anomaly detection.
- Cloud Layer: Centralized platforms handle long-term storage, advanced analytics, and coordination across inspection missions.

The implementation begins by identifying the latency and compute requirements for drone operations and inspection analytics. The architecture is then designed to distribute tasks across the drone, edge nodes, and cloud platforms. Prototypes are developed to test edge-based video analytics and cloud coordination. Integration ensures reliable data flow and responsiveness during inspection missions. Deployment focuses on real-world testing. The first phase of the enabler implementation is planned till Q1 2026, with the aim of first use case integration till Q2 2026. Based on the outcome of the integration, some finetuning of the implementation may take place after Q2 2026.

Design, development and implementation for the T1 and T2 use cases: In the T1 and T2 use cases, the computing continuum involves edge nodes, managed via Kubernetes, and extreme edge nodes or devices like RSUs (T1) or UGVs (T2) managed via K3S. On top of them, the Resource Orchestrator through a Resource Allocation Engine is responsible to handle all the resources in the different clusters, distributing the containers among the nodes taking into account the constraints on power consumption and battery level. Automatic task offloading from RSUs or UGVs towards the edge are managed through closed loops, with the objective of guaranteeing the service continuity while optimizing the use of batteries and renewable energy sources on the devices. The deployment of edge platforms on RSUs and UGVs, with the related Resource Orchestrator, is planned for H1-2026. The first version of the Resource Allocation Engine will be deployed in Q2-2026, while the closed loop mechanisms for automatic task offloading will be integrated starting from Q3-2026.

Design, development and implementation for the T4 use case: The implementation of the ZSM framework for the tele-operation use case is designed to leverage its core capabilities in AI-driven orchestration, cross-domain management, and intent-based automation. This enables seamless switching between autonomous and tele-operated driving modes while ensuring dynamic resource allocation, low-latency communication, and quality-of-service (QoS) awareness across edge and cloud infrastructures.

The first phase (Definition and system integration) involves connecting the ZSM framework to 5G/B5G network APIs, such as CAMARA QoD, to enable continuous ingestion of network metrics including latency, bandwidth, and jitter. Additionally, the framework subscribes to multimodal data streams such

as geolocation or V2X notifications to support context-aware decision-making. Concurrently, QoS intents and profiles are defined to model operational requirements for both autonomous and tele-operation modes such as standard uplink bandwidth and enhanced QoS with uplink speeds ≥25 Mbps and latency ≤20 ms. These intents are mapped to CAMARA QoD API parameters to facilitate automated policy-driven enforcement.

The second phase (Deployment and Validation), consist on testing the orchestration framework under real-world conditions. ZSM orchestration functions such as the Orchestrator, Deployer, and Publisher/Subscriber modules, along with vehicular service components like video processing and sensor fusion, are deployed on Kubernetes-based Network Function Virtualization Infrastructure (NFVI). Validation is conducted through controlled scenarios that simulate transitions between autonomous and tele-operated modes. Key performance indicators, including end-to-end latency, jitter, throughput, and network slice adaptation, are monitored in real time to ensure resilience under high load and adverse conditions, such as network congestion or sensor failures.

The final phase (Optimization), aims to enhance the intelligence and efficiency of the orchestration framework. This includes extending the ZSM decision engine with machine learning-based anomaly detection and traffic forecasting to proactively trigger QoS adaptations. Fallback mechanisms are implemented to maintain service continuity in cases where QoS upgrade requests are delayed or denied. Additionally, energy awareness is incorporated into orchestration decisions by introducing power consumption as a key performance indicator. This allows the system to optimize resource allocation, balancing performance with energy efficiency during prolonged tele-operation sessions for sustainability and operational cost reduction.

The first and the second phase will start during the first year and be finalized in the second year of the project, while the final phase will be started in the second and finalized by the middle of the third year.

3.3 Cloud-native service design

This section dives deeper into the cloud-native design principles of vertical services, which are part of the AMAZING-6G use cases. This design style is enabling vertical services to be uniformly and flexibly deployed on compute resources within the distributed compute continuum, regardless of the service type.

3.3.1 Description of the enabler

This enabler plays a critical role in translating high-level goals into executable, adaptable, and observable actions across network infrastructures. By following cloud-native principles like declarative configuration, immutability, microservice granularity, and infrastructure abstraction, it provides the agility needed to adapt quickly to dynamic environments. This is particularly important in areas such as edge computing and 5G network slicing, where services must be provided rapidly, resources allocated efficiently, and recovery automated to maintain performance and reliability.

The architecture relies on Kubernetes as the primary orchestration substrate, with network functions and service definitions represented as Custom Resource Definitions (CRDs). This approach enables declarative network service definition while supporting Containerized Network Functions (CNFs) within ETSI-compliant NFV orchestration frameworks. Through this integration, predefined templates can be applied, generated, or adapted in response to high-level user intents. For example, an intent such as “Prioritize EV telemetry with low-latency slices” can be translated into actionable network slice templates with specific QoS parameters, placement rules, and policies. These templates encapsulate services, functions, QoS characteristics, and dependencies, and can be instantiated through GitOps pipelines or direct API-driven orchestration frameworks.

The cloud-native paradigm extends beyond network functions to encompass microservices for IoT applications, data analytics, and edge services. Its microservice architecture enables independent scaling and flexible orchestration, while dynamic resource management provisions compute, storage, and I/O resources in real-time based on service demand.

In parallel, mobile networks face unprecedented challenges from traffic growth, diverse 5G use cases, and stricter service-level requirements. Traditional appliance-based RAN architecture cannot meet the agility and cost-efficiency needs of operators. Consequently, the industry is transitioning toward cloud-native RAN (CN-RAN), where RAN functions are virtualized, containerized, and orchestrated using Kubernetes. Functional splits of the gNB, such as CU, Distributed Unit (DU), and CU-UP, are packaged as microservices running on commodity servers. This design enables automated deployment, scaling, healing, and lifecycle management, treating RAN workloads like any other cloud application: deployed declaratively, updated continuously through CI/CD pipelines, and dynamically scaled according to demand. As a result, the RAN inherits the elasticity and resilience of cloud-native systems while still meeting stringent performance and latency requirements.

At the edge, Edge Network Applications (EdgeApps) further operationalize cloud-native principles. Defined and validated within the VITAL-5G framework, EdgeApps are modular, cloud-native software components packaged according to ETSI NFV standards, ensuring portability across NFV-compliant and Kubernetes-based infrastructures. Each EdgeApp includes a standardized VNF package, a blueprint encoding service/network/infrastructure awareness, and artifacts such as test cases, documentation, and licensing. The blueprint enables EdgeApps to interpret real-time network states and interact with exposure APIs, e.g., CAMARA, Common API Framework (CAPIF), Service Enabler Architecture Layer (SEAL), EDGEAPP. In doing so, they autonomously adapt their behavior, triggering QoS upgrades, shifting placement, or rerouting processing, in response to performance degradation, user intent, or contextual changes.

By combining intent-driven orchestration, cloud-native RAN, and EdgeApps, this enabler provides the foundation for responsive, resilient, and intelligent services across the UE–edge–cloud continuum. It transforms high-level intents into automated, adaptive network actions, ensuring that next-generation infrastructures can meet the flexibility, scalability, and performance demands of 5G and beyond.

3.3.2 Use case association and contributing partners

In the **E1 use case**, the enabler will be applied to realise intent-based orchestration across heterogeneous cloud and edge environments. A Generative AI-enabled intent engine will translate high-level service requirements into declarative templates, aligned with the cloud-native model described in Section 3.3.1. These templates are then instantiated and managed through the orchestrator, which extends Kubernetes to handle Custom Resource Definitions (CRDs) and supports the lifecycle management of network and service functions across multiple clusters.

This integration enables automated provisioning, scaling, and adaptation of both CNFs and supporting microservices, ensuring that slice configurations and QoS policies are consistently enforced across distributed infrastructures. By combining intent translation, template generation, the framework provides a flexible mechanism to align network and application resources with evolving service intents in the E1 use case.

Regarding the **T4 use case**, the EdgeApp framework has already been successfully applied in the VITAL-5G project to deploy various domain-specific applications across transport and logistics verticals, particularly within real-world maritime environments (i.e., improving port safety, reducing dwell times, and reducing fuel consumption). These EdgeApps, such as those enabling situational awareness and quality-aware remote vessel control, are designed as cloud-native, modular services that interact with both user equipment (e.g., vessels) and the 5G SA network. They dynamically respond to events (e.g., geofenced zones, obstacle detection, or degraded QoS) by programmatically triggering network adaptations via CAMARA APIs, ensuring optimal performance for mission-critical services. EdgeApps are

packaged following ETSI NFV-compliant principles, including descriptors, test cases, and blueprints, allowing seamless deployment across the compute continuum (extreme edge, edge cloud, centralized cloud) on any NFVI-compatible infrastructure. This same design paradigm will be applied to T4, where intelligent, event-driven EdgeApps will support context-aware transitions to tele-operation and optimize network resource usage through QoS on-demand mechanisms. Furthermore, middleware solutions like our in-house EdgeApp for slice management and orchestration demonstrate how verticals can access per-UE QoS control as a service, abstracting network complexity and enabling plug-and-play CAMARA API integration without embedding telco-specific logic.

3.3.3 Design, development and implementation

High-level design of Cloud-native services: Cloud-native services are a shift how modern networks and applications are designed, deployed, and managed. By leveraging microservices, containerization, orchestration platforms, and declarative APIs, they enable flexible, scalable, and resilient solutions. This approach transforms networks from static infrastructures into programmable, adaptive platforms capable of meeting the stringent requirements of verticals such as autonomous driving, teleoperation, and energy management. In this context, the enabler, represented in Figure 3-14, is being designed to support intent-based management across heterogeneous cloud and edge environments. Service intents are captured as declarative templates and processed through an orchestration layer that manages provisioning, scaling, and adaptation across multiple clusters. A key element of the architecture is the use of network exposure through CAMARA APIs, which allow edge applications to translate contextual events into programmable network actions—for example, dynamically requesting enhanced connectivity profiles with higher throughput or lower latency when required.

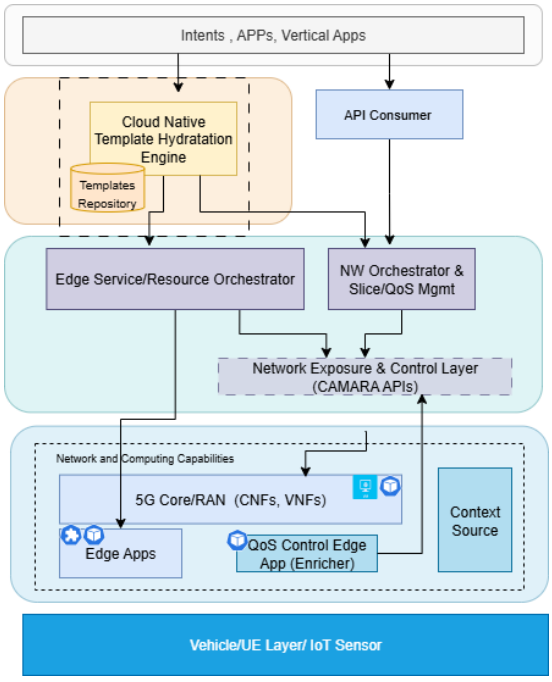

Figure 3-14 Cloud-Native Services Management High Level Architecture

At the middleware layer, the Enricher component extends this architecture by enabling event-driven QoS control. Implemented as a modular, cloud-native service, Enricher listens to triggers such as geofencing or degraded network conditions and issues Quality on Demand (QoD) requests to the 5G Core. This ensures additional capacity is provisioned during critical operations, while resources are automatically scaled down when no longer needed. Together, these components create a high-level architecture that

combines cloud native orchestration, declarative intent translation, and event-driven QoS control, delivering efficient, reliable, and adaptive network services across the edge–cloud continuum.

Design, development and implementation for the E1 use case: In the E1 use case, the enabler is being designed to support intent-based orchestration across heterogeneous cloud and edge environments. The design phase defines how high-level service requirements are expressed as declarative templates. In development, these templates are integrated with an orchestration layer that extends Kubernetes to manage network and service functions through Custom Resource Definitions (CRDs). Implementation will enable automated provisioning, scaling, and adaptation of CNFs and microservices across clusters, ensuring consistent enforcement of slice configurations and QoS policies as service intents evolve.

Planning steps for implementation:

- Requirements and interfaces – define service intents, parameters, and integration with OSS/ORO APIs.
- Template design – create declarative templates for services, QoS profiles, and placement rules.
- Workflow setup – configure orchestration workflows and map templates into CRDs.
- Prototype deployment – test closed-loop orchestration in the E1 testbed.
- Validation and optimization – refine templates and workflows for scalability, consistency, and readiness.

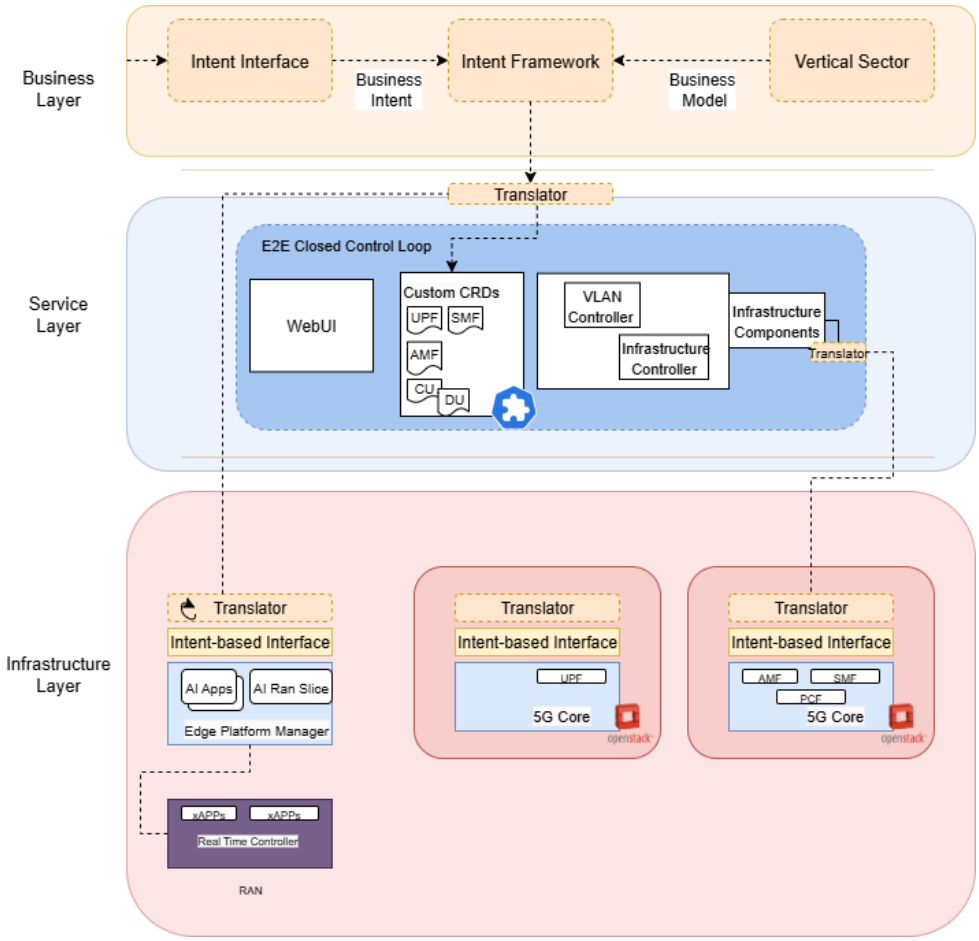

Figure 3-15 Intent Cloud Based Architecture for Network Resource Management

In the 1st year, CAPG will define service intents, interfaces, and templates for services and QoS profiles. In Year 2, integrate workflows, map templates into CRDs, and deploy a prototype in the testbed. In Year 3, refine and optimize the system and full operational readiness, in order to perform the final trial.

Design, development and implementation for the T4 use case: In the context of the T4 use case, a dedicated QoS Control EdgeApp is designed to manage per-vehicle QoS dynamically by leveraging

network exposed APIs such as CAMARA. This EdgeApp is cloud-native, event-driven, and deployable across the UE-edge-cloud continuum. Its primary function is to ensure that the tele-operation mode receives guaranteed high throughput and low-latency connectivity, while avoiding over-provisioning when the vehicle is operating autonomously.

The QoS Control EdgeApp consumes contextual triggers such as:

- The output of the ML-based anomaly detection algorithm, indicating degraded autonomous capability.
- Location-aware evens, such as geofencing into complex zones (e.g., construction areas, high-density traffic).
- Real-time network telemetry, indicating poor network quality or congestion.

Upon receiving a trigger (e.g., switch to tele-operation), the EdgeApp issues a CAMARA QoD request to the 5G/6G core to apply an upgraded slice profile with enhanced QoS (e.g., higher uplink bandwidth, reduced latency). Once the vehicle resumes autonomous driving, the EdgeApp releases the enhanced QoS, requesting a return to the default profile. This results in a context-aware, efficient use of network resources tailored to vehicle state and operational needs.

Planning steps for implementation:

- Integration with Context Sources
- Subscribe the EdgeApp to relevant event streams: ML triggers (TUC), geolocation data, and ZSM network monitoring.
- Define QoS Profiles and CAMARA Mappings
- Map tele-operation requirements (e.g., uplink ≥ 25 Mbps, latency ≤ 20 ms) to CAMARA QoD API parameters.
- Define base (autonomous) and enhanced (tele-operation) QoS profiles.
- Deploy on Edge Infrastructure
- Instantiate the EdgeApp at the network edge using the NFV orchestration layer, collocated with other latency-sensitive functions.
- Validation and Testing
- Run simulations where vehicles transition in/out of zones that require tele-operation.
- Monitor the network slice adaptation behavior and QoS performance (e.g., Round Trip Time (RTT), jitter, throughput).
- Refinement and Optimization
- Extend the EdgeApp to support predictive triggers (e.g., AI-based obstacle detection).
- Integrate fallback logic and alerts in case QoS upgrade requests are denied or delayed.

As part of prior research and implementation activities, we have already developed a modular and reusable middleware component named Enricher, which will serve as a foundation for the QoS control functionality required in T4. Enricher is implemented as a cloud-native EdgeApp capable of dynamically managing per-UE QoS through interaction with network exposure APIs such as CAMARA. It has been validated in a real-world maritime trial facility within the VITAL-5G project, where it successfully enabled dynamic QoS upgrades for remotely operated vessels based on event-driven triggers such as geofencing or network quality degradation. In these scenarios, Enricher demonstrated its ability to initiate QoD requests to the 5G Core, improving uplink capacity during remote operation, and releasing resources once remote control was no longer needed. These capabilities are directly transferable to T4, where the same logic will be applied to trigger and release high-performance network slices during transitions between autonomous and tele-operated driving. The Enricher EdgeApp is fully NFV-compatible, supports CAPIF-based event subscription, and can be orchestrated across the UE-edge-cloud continuum using standard service descriptors.

The comparative performance of Enricher was evaluated across three scenarios and is summarized in the Figure below. Scenario 1 (Baseline with static slicing) showed limited adaptability, resulting in high

SLA violations (381) and low SLA efficiency (65.5%), while wasting uplink capacity during idle periods. Scenario 2 (Unoptimized overprovisioning) improved SLA adherence (93.3%) but incurred unsustainable uplink usage (31.4 GB total, ~50 Mbps even when idle), highlighting poor scalability. In contrast, Scenario 3 (Enricher-enabled) achieved the best SLA efficiency (97.3%) and the fewest SLA violations (59), while also reducing total uplink usage to 20.1 GB (a 36% drop from Scenario 2). It also dramatically cut idle zone uplink from nearly 50 Mbps to just 2.87 Mbps, demonstrating Enricher's ability to scale resources up and down based on operational context. This makes it an ideal candidate for T4, where tele-operation requires just-in-time network resource guarantees without overburdening shared infrastructure.

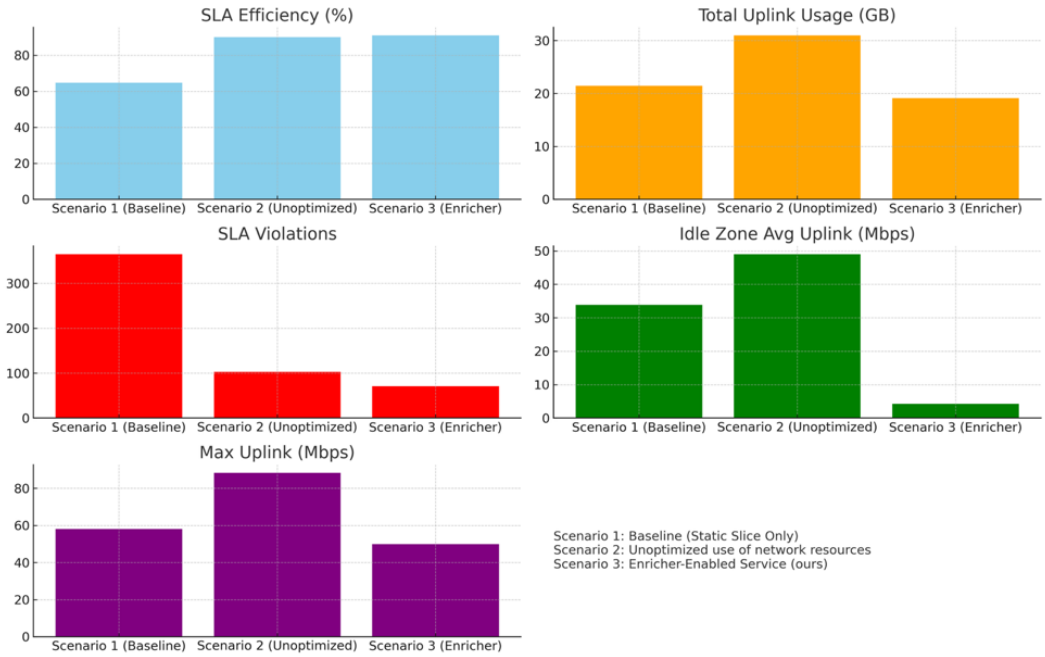

Figure 3-16 Evaluation Summary Across All Scenarios

This enabler will be fully integrated with the use case T4 in the second year of the project, while some preliminary tests are already ongoing in IMEC's testbed environment.

3.4 Creation of Kubernetes clusters on demand

This section dives deep into the final CaaS enabler, i.e., Creation of Kubernetes cluster on demand, focusing on its design, and its association with the P1 use case.

3.4.1 Description of the enabler

In the latest years the industry has turned into deploying and running application using the Kubernetes platform. As an open-source platform for automating the deployment, scaling, and management of containerized applications, such platforms enable the pooling of computing resources (like CPU and memory from multiple servers) into a single, unified resource and then takes care of efficiently scheduling and running application "containers" across those resources, ensuring they are always available and can scale seamlessly with demand. For AMAZING 6G, the ability to create these Kubernetes clusters on-demand is a fundamental enabler because it provides the essential cloud-native fabric for the network itself. As AMAZING-6G envisions a highly flexible architecture where softwarised network functions—from the core to the far edge—need to be instantiated, scaled, and terminated in real-time to meet the specific vertical needs, creating a Kubernetes cluster on-demand allows the network to instantly provision a precisely tailored, isolated compute environment exactly where it's needed. To that respect providing the agile and scalable platform required to host these critical network and vertical services, intelligent applications, and network slices, will be integrated into the Amazing-6G

concept through a Kubernetes-as-a-Service concept thereby making the vision of a truly flexible, software-defined 6G network a practical reality.

To integrate the on-demand creation of Kubernetes clusters as a native capability, the 6G network must evolve into a compute-aware, software-defined, and fully integrated fabric that seamlessly merges communication, computation, and intelligence. The network must be built upon a distributed edge cloud continuum, spanning from centralized data centers to far-edge nodes. This infrastructure must provide standardized hardware abstraction to ensure the underlying compute resources are uniformly exposed and available for provisioning, regardless of their physical location. Furthermore an AI-Native Orchestration that manages both the communication requirements (e.g., latency, bandwidth, slice isolation) and the compute requirements (e.g., memory needs etc) of a requested service is requested. This means that making real-time decisions on where to instantiate a cluster within the network fabric to meet stringent performance goals, automatically handling the lifecycle management implies a fully operational orchestration platform that interfaces the overall infrastructure through APIs.

3.4.2 Use case association and contributing partners

In disaster or emergency scenarios like the **P1 use case**, AR/VR analytics, video transcoding, and AI-based situational awareness services must be spun up quickly near the incident location even if the availability of network and compute resources is compromised. For this reason we will first deploy a private network with compute capabilities to support the first responders but will also instantiate on-demand a K8s cluster instantiation possibly at the extreme edge or edge nodes (operator MECs) to allow PPDR agencies to deploy these workloads within seconds to minutes. Furthermore as the Public Protection and Disaster Response (PPDR) slice may span multiple operators, federated K8s clusters can be created on-demand across different operator edge clouds. However a mechanisms is needed to ensure seamless workload placement and migration—for example, if Operator A's edge is overloaded, a K8s cluster can be created dynamically on Operator B's edge, maintaining service continuity.

3.4.3 Design, development and implementation

High-level design of Creation of Kubernetes clusters on demand: In order to support all the above mentioned actions, a mechanism is needed that can coordinate the compute resources available and dynamically spin up or tear down clusters across edge/cloud to handle e.g., GPU-heavy AR rendering tasks at the edge in a reliable way, noting that K8s on-demand creation at another edge or cloud site ensures automatic workload redeployment, maintaining high availability for mission-critical applications. In figure a high level design of the Creation of Kubernetes clusters on demand enabler is shown. Here when the service request is ordered, configuration for network and compute resources should take place which in turn are requested through the cloud orchestrator. Once the K8s are deployed, the applications are deployed and delivered according to the service description.

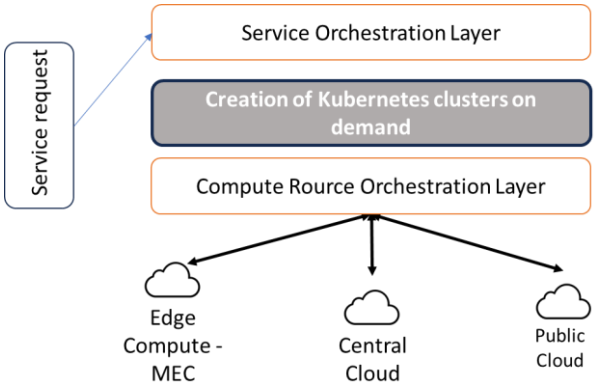

Figure 3-17 High-level design of Creation of Kubernetes cluster on demand

Design, development and implementation for the P1 use case: Patras5G testbed allows for full control of compute and network resources (including but not limited to Kubernetes) through OpenSlice

[https://osl.etsi.org/] an open-source OSS. In case of Kubernetes cluster, OpenSlice offers a service named Kubernetes-as-a-Service (K8SaaS), which provisions a Kubernetes cluster and allowing all administrative access to the ordering consumer. If needed GPU support can also be integrated in the deployed K8S cluster, while this order can be performed either by GUI or TMF compatible APIs.

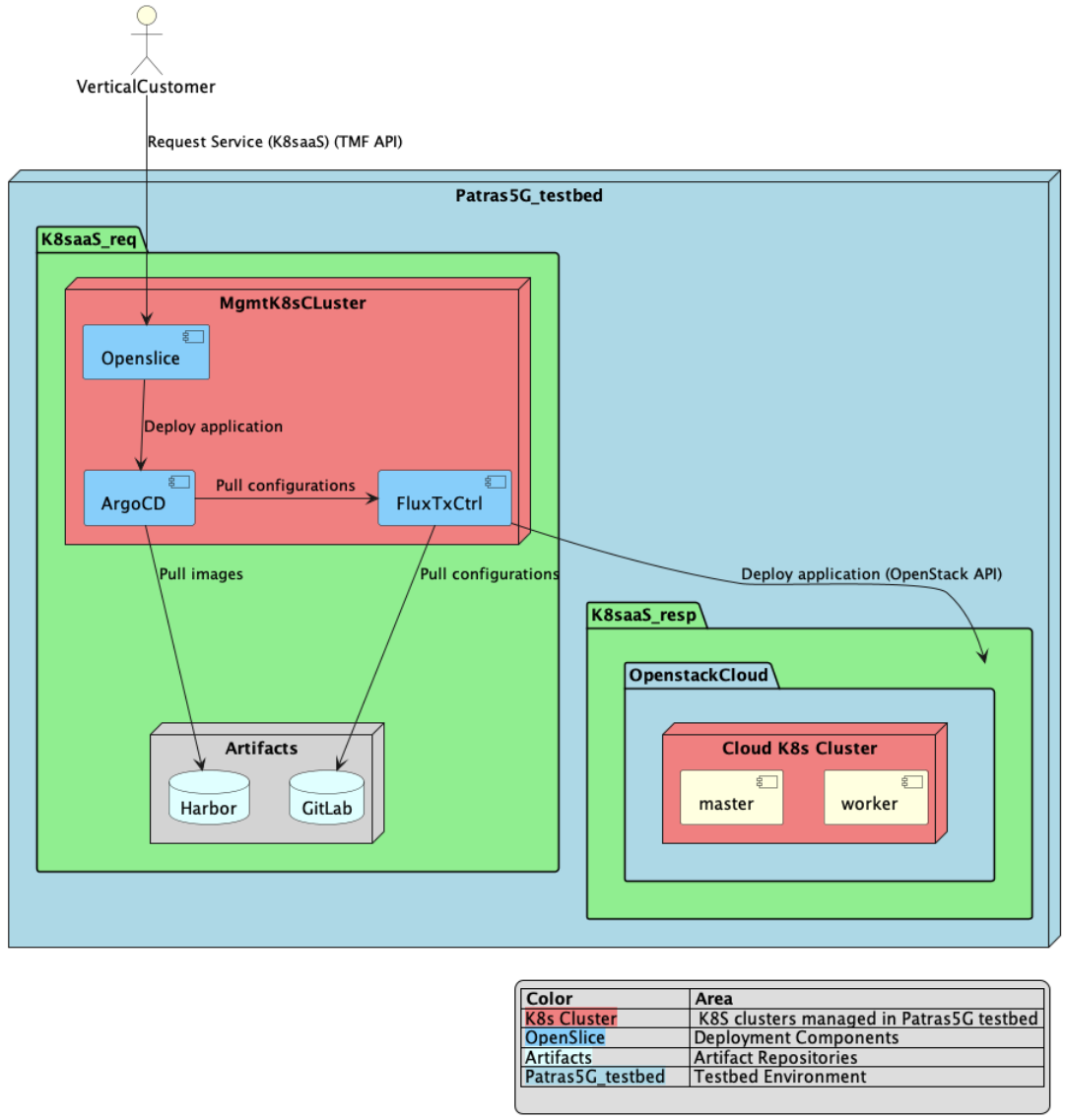

Figure 3-18 Overview of K8saaS Compute Enabler

This exists inside runs inside a K8s cluster (Management Cluster) along with ArgoCD and Flux TF controllers. These components are used to deploy Helm charts pulled from an artifacts repository, such as Harbor (ArgoCD) while the TF controller is executes Terraform code retrieved from git repositories.

A sample workflow for such a deployment can be seen in figure 318: An external customer will place an order for the creation of a K8S service. The first step will be the deployment of an ArgoCD application which will in turn use the Harbor repository to get a Helm chart that includes the required resources for the deployment. These resources are then used by the Terraform controller which also uses the OpenStack API to manage the Patras5G cloud infrastructure and finalize the deployment. Once the K8s cluster is deployed and acquired, further applications can be into it depending on the verticals customers' requirements.

When ordering the K8S, the following parameters can be defined: Attached networks; Availability zone; Flavor of worker nodes; Number of worker nodes.

The deployment of the enabler is planned for the beginning of 2026 (Q2), while the integration with thePatra5G platform is planned for Q2-2026, together with an initial tests.

3.5 Summary

This chapter has presented the CaaS enablers, covering the following categories: (1) Compute resource management with three sub-enablers: 1. Intelligent service and resource orchestration in extreme edge/edge/cloud compute continuum (focusing on decision-making logic when deploying and maintaining service deployments across compute resources), 2. Intent-based service and resource management (bringing automation to compute resource management procedures), and 3. Real-time monitoring of compute resources and energy consumption in compute continuum (providing essential input for orchestrator's decision-making logic). (2) Compute continuum covering two sub-enablers: 1. Distributed user-edge-cloud compute continuum (tackling distributed compute resources within diverse NFVI environments), and 2. Multi-party cloud (focusing on distribution of federated cloud resources used simultaneously by different stakeholders). (3) Cloud-native service design, discussing the main principles of designing Beyond 5G and 6G services/applications, including the virtualization principles, application and service descriptors, and their overall packaging. (4) Creation of Kubernetes cluster on demand, detailing on the process of dynamic assembling of virtualized compute resources based on the needs from vertical services.

Each of the enablers and sub-enablers is first described from a general perspective, and then associated with the AMAZING-6G use cases, describing the relevance and importance of those enablers for enhancing vertical use cases/services. Afterwards, the use case-agnostic high-level design is provided for each enabler/sub-enabler, which is followed by the design and implementation steps that are planned in the scope of different use cases.

4 Application enablers and AI

This section provides an overview of application enablers and artificial intelligence, as key elements which allow to use new technologies or best practices in companies for enhancing their efficiency and productivity. The main aim is to abstract technologies complexity, process automation and allow an high customization in order to be compliant to specific needs of the market. In particular, this section focuses on three enablers identified as AI-as-a-Service (AIaaS), Digital Twins, and OpenAPIs for network exposure.

AIaaS refers to the possibility to use, train and manage AI models in the edge/cloud delivered as a service through intuitive APIs. This simplifies the design and implementation of AI solutions and allows companies to focus on added value features rather than infrastructure details, reducing costs and complexity. AIaaS pones its root in the field of cloud computing. For example, several ready-to-use applications exist and are classified in SaaS (Software as a Service) category. Additionally, two other categories of cloud services can be applied: PaaS (Platform as a Service) to customize models and integrate them into production processes, and IaaS (Infrastructure as a Service) that provide raw hardware like GPUs and optimized resources for training. Essentially, the advantage for the customers is that they only pay for the services they actually use, scaling as needed, and reducing risks and development time. AIaaS field includes ML model catalogues, AI/ML functions orchestrators, and MLOps platforms designed to manage the lifecycle of AI/ML services and models.

The Digital Twin Framework is a solution that not only simply represents a model, but also offers a virtual copy of a physical asset. This twin is created processing data from IoT sensors, sensor networks and databases, updating the information of the digital twin in real time. The added value lies in the direct connection to the real object, which enables the collection of operational data, their analysis using ML algorithms, and the simulation of future behavior. AMAZING-6G, for example, aims to use this enabler for logistics, transportation, and even to orchestrate components of the network.

Finally, OpenAPIs for Network Exposure aim to make the most advanced features of mobile networks available to application developers without overcomplicating the technical side. For example, the CAMARA APIs for Quality on Demand (QoD) or for energy management allow to activate pre-established connectivity service profile and enable energy monitoring while maintaining abstraction from technical details. AMAZING-6G will contribute to the adoption and integration of a subset of these APIs as detailed in the following subsections.

4.1 AI-as-a-Service framework

This section describes AMAZING-6G AI-as-a-Service framework, which provides the foundation for delivering AI capabilities in a flexible, scalable and simplified manner via 6G infrastructures, to manage and implement the data-driven intelligence of vertical applications. The framework brings together three core technological enablers: AI-as-a-Service (AIaaS), MLOps platforms, and ML model catalogues. AIaaS exposes the access to AI functionalities delivered via edge and cloud resources, through unified and standard interfaces for on-demand interactions. MLOps platforms are used for the continuous integration, deployment, operation, and monitoring of ML pipelines. ML model catalogues provide structured repositories of trained and validated ML models, for specific knowledge domains, exposing their metadata to simplify their reusability in different contexts. The combination of these elements, which can be integrated in advanced service offers from network operators or embedded in private mobile networks, can facilitate the AI adoption in several vertical sectors, reducing development effort, promoting reusage of trained models and knowledge sharing, thus contributing to trustworthiness and sustainability of AI-based solutions.

4.1.1 Description of the enabler

AI-as-a-Service

AI as a Service (AIaaS) provides access to AI third-party tools and AI frameworks deployed on edge, cloud or hybrid platforms through a unified set of open interfaces. This simplifies the AI accessibility, reducing the need of highly specialized technical know-how and avoiding the infrastructural costs to run the highly demanding AI tasks. As anticipated in the introduction, this ecosystem offers advanced features that focus on optimizing every stage of the ML model lifecycle such as model querying, hyperparameter optimization, and the customization of the models for different environments. These features are offered to the consumer as a generic service invocation and therefore hide the internal complex implementation details. In public commercial scenarios, payment is generally made on usage basis, and scalability can be adapted to specific needs. However, the concept is also applicable to private infrastructures to efficiently manage the resources dedicated to AI/ML functions and provide a simplified interface to vertical operators, service developers or vertical applications themselves.

A key benefit of AIaaS relies on the efficient use of edge and cloud resources, since it overcomes the limitations of traditional AI approaches about the high cost of building and maintaining complex infrastructures and the high skills required in the development. Furthermore, AIaaS offers a continuous update system since the providers of such services constantly work in order to improve and keep updated their products. Consumers have access to up-to-date solutions without the need to continuously scouting innovative technologies.

AIaaS offers a wide range of functionalities such as access to pre-trained models and ready-to-use algorithms, models training and deployment, including fine-tuning of pre-trained models, inference services, tools for data pre-processing as input for ML pipelines, and ML models lifecycle management via integration with MLOps. On one hand, ML model catalogues or marketplaces give access to repositories of trained and validated models that can be reused as is or adapted to different target scenarios. On the other hand, automation services for management of AI/ML pipelines, local or on edge/cloud, allow to orchestrate custom pipelines from data ingestion and training, up to deployment and continuous monitoring stages.

AIaaS can also expose various levels of customization options, targeting customers with different expertise and requirements at development and operational stages. For example, AIaaS can offer pre-built bundles of AI services that can be directly integrated in the applications following a simple plug-and-play approach. In this case, the AI functions usually target common and generalized applications like computer vision, speech recognition, processing of natural language, etc. AI services offering deeper degrees of customization are instead designed to facilitate the development of domain-specific models, tailored to particular datasets or use cases, requiring a deeper level of expertise for the users.

Finally, it should be noted that AIaaS is not limited to deployments entirely based on public edge/cloud resources, but may also include hybrid approaches. In this latter case, the AIaaS backend logic coordinates the deployment of AI functions in local edge devices or private networks (e.g., to guarantee the processing close to the data sources) with a seamless integration of AI tasks running in the public cloud.

ML model catalogue

A ML model catalogue is a tool that supports storage, exposure, advertisement, discovery, query and selection of AI/ML algorithms and models. Searching functionalities are usually facilitated by metadata that report tags, model descriptions, versions, hyperparameters, training and deployment history, and references to the related datasets. ML model catalogues usually integrate mechanisms for security and trustworthiness, with access control, management of licenses, and audit trail. This approach allows data scientists or analysts to easily find, access and retrieve ML models, facilitating their reusability.

ML catalogues are often integrated with MLOps pipelines: the models are pushed to or pulled from the catalogue during the execution of pipeline workflows, e.g., as result of training and testing stages, or for deployment and validation in target environments. This integration allows to implement advanced functionalities to guarantee the quality of the models available in the catalogue. For example, a pipeline can apply certification procedures to validate a models before allowing its publication in the catalogue. The certification can address several aspects, from accuracy and robustness to AI security attacks, up to the compliance with different capabilities of target environments, e.g., guaranteeing the possibility to run the model over a constrained node.

ML models metadata and versioning allows to run several rounds of experiments with an algorithm, using different configuration parameters and compare the obtained performance for an effective tuning to variable operational conditions. Furthermore, rich metadata are fundamental for the efficient distribution of packaged or containerized ML models, specifying their dependencies and deployment requirements.

MLOps

MLOps (Machine Learning Operations) is a set of practices that automates the lifecycle of ML models, from their development to their deployment and monitoring. This originates from the DevOps principles, such as continuous integration and continuous delivery, in order to create a system able to automate the tasks related to ML models and pipelines. MLOps functionalities are depicted in Figure 4-1, which represents the internal functionalities of a generic MLOps platform.

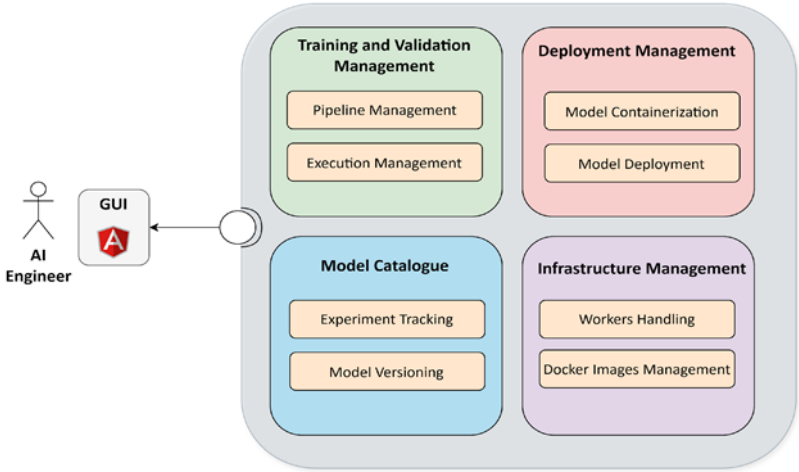

Figure 4-1 MLOps functionalities

The platform's core is its smart orchestration of AI/ML pipelines, which allows for transparent management of computational resources dedicated to AI/ML tasks, training pipelines and deployment policies. When a training request is submitted, the AI engineer simply needs to provide a few minor details. These include the requested model parameters for the given run and the target infrastructure where the experiment is to be executed. The orchestrator is responsible for setting up the necessary nodes satisfying the specified requirements and for interacting with the ML model catalogue to store training metrics and model metadata. Similarly, once the trained models have been recorded, the data scientist may request additional functionalities such as model monitoring or deployment. It is still the responsibility of the orchestrator to simplify the complexity of the service and to ensure the model serving in the infrastructure, while also exposing a simple API for inference.

4.1.2 Use case association and contributing partners

Table 4-1 shows the mapping between AI-related enablers and AMAZING-6G use cases implementing them. This enabler may also be applicable to other use cases, but this is not covered in this project.

Table 4-1 Mapping between AI-related enablers and AMAZING-6G UCs

	AI-aaS	ML models catalogue	MLOps
P1	WINGS	WINGS	
E1	NXW/CAPG	NXW/CAPG	NXW
E3	CAPG	CAPG	
T2	NXW		
T5		CERTH	CERTH

In the **P1 use case**, AI functionality is essential for environment monitoring and incident detection. Partners develop AI functionality which monitors various parameters of the environment, e.g.by using IoT sensors for temperature, humidity, microparticles, smoke etc. and provides an analysis of the data and identification of potential security incidents (e.g. fires etc.).

The **E1 use case** adopts AI/ML techniques to create profiles on rooms utilization and preferences on building comfort settings (temperature, lighting, etc.), as well as for prediction of power consumption and production by renewable energy sources. Moreover, given the dynamicity of some building environments, models often need to be tuned or re-trained, considering the feedback gathered from the system (e.g., to adapt to new users' preferences). Training, inference, continuous monitoring, tuning and re-training of ML models are managed through an MLOps platform, with models stored in a ML model catalogue. The E1 scenario with Renewable Energy Communities (REC) comprising several buildings makes use of Federated Learning (FL): models are trained by FL clients in each building and they are then merged at the REC level by the FL aggregator. Here, the system orchestrates both training and inference tasks along edge (at the buildings) and cloud continuum. This is handled via AIaaS, exposing APIs to request ML models training and deployment actions in a transparent manner. Internally, AIaaS requests are handled through a distributed MLOps platform, extended for FL support, that manages the selection of the target nodes for AI/ML workloads in the continuum and orchestrates the FL procedures between building clients and REC aggregator.

The **E3 use case** aims to provide an E2E application for the green energy vertical, featuring energy characteristics monitoring and energy production capabilities. In context of the use case, the integration of AIaaS and a comprehensive ML Catalogue plays a fundamental role. These components provide a scalable and modular environment for deploying, managing and reusing ML models in this case tailored to solar energy forecasting. The AIaaS framework enables on-demand access to these models, reducing complexity and allowing integration into the data pipelines operating at the edge or in the cloud. The ML catalogue here ensures traceability, control and feasibility of models promoting experimentation and continuous improvement. It will help in enhance the accuracy, adaptability and efficiency of predictive analytics in energy systems, ultimately supporting better resource planning and grid stability in smart energy communities.

The **T2 use case** adopts AI/ML algorithms for image recognition and object detection, with dynamic migration of related AI workloads between UGVs and edge nodes. This is managed automatically via AIaaS, allowing the users to request the provisioning of AI functions and delegating to the system backend the decisions about where these functions will be executed.

The **T5 use case** applies AI/ML predictive modeling techniques to optimize multiple aspects of operations at the Port of Thessaloniki. These models will be embedded within the port's Digital Twin environment, enabling decision makers to operate at a higher level of detail and accuracy. For example,

an advanced prediction model for vessel arrivals will provide more reliable forecasts, allowing the port to proactively allocate yard resources and anticipate the inflow of containers. This will directly support the efficient planning of both human resources and straddle carrier availability. In addition, AI-driven sequential decision-making methods will be incorporated into the simulation tool to optimize the routing and task assignment of straddle carriers within the yard, ensuring more efficient completion of container handling operations.

4.1.3 Design, development and implementation

Figure 4-2 shows the high-level design of the AIaaS enabler system. The ML Model Catalogue provides the repository for trained models, with related artifacts for model deployment and metadata for models filtering and searching. The models are onboarded and consumed by the MLOps Platform, which handles the logic for the lifecycle management of ML models, from development to training, testing, deployment, and continuous monitoring, and the orchestration of the AI/ML pipelines on top of the available infrastructure resources. The provisioning and orchestration of the AI/ML functions is optimized according to the capabilities offered by the computing resources in the edge/cloud environment, handled through the infrastructure management component. Depending on the scenario, the MLOps platform can included additional modules specialized for particular features, e.g., for the management of Federated Learning processes. Training and inference are performed over data coming from different data sources, with real-time or historical data. AI/ML-based applications can consume AI/ML services interfacing directly with MLOps platform and ML model catalogue or through the mediation of the AIaaS framework. This provides a unified interface to invoke AI/ML functionalities (e.g., the query of a trained model, the request for a new training task, the provisioning of an inference function with a given model and using a given dataset, etc.), wrapping the complexity of the underlying AI/ML system and the potential heterogeneity of the MLOps platforms, ML models, and edge/cloud platforms.

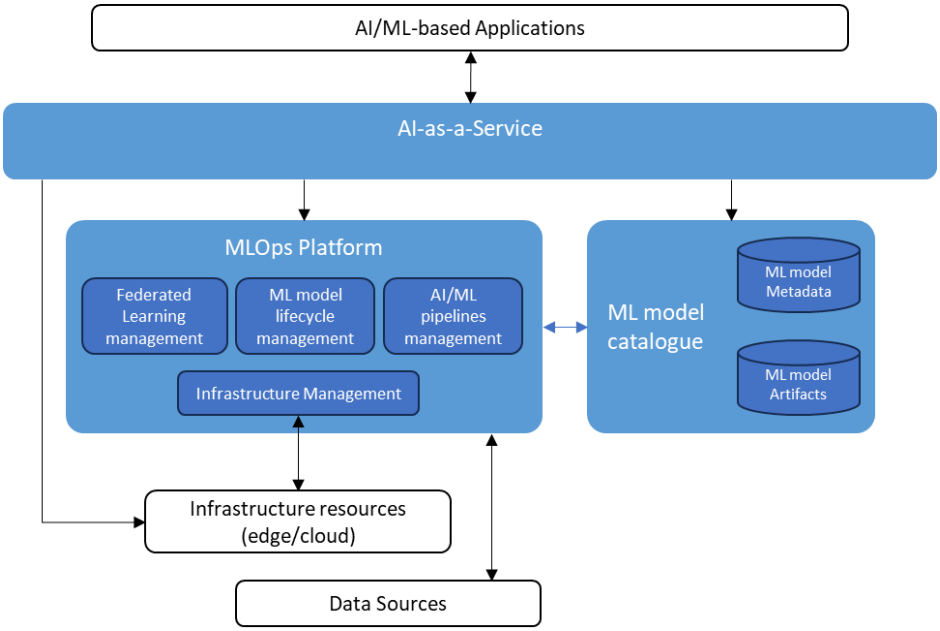

Figure 4-2 High-level design of AI-aaS and MLOps frameworks

In the following the UC-specific implementations of the AI enabler are described.

In the **P1 use case**, AI functionality plays a critical role in enabling advanced environment monitoring and timely incident detection. The system leverages AI-driven analytics to process large volumes of data collected from a diverse range of IoT sensors deployed throughout the monitored area (see Figure 4-3). These sensors continuously measure key environmental parameters such as temperature, humidity, air quality, microparticle concentration, and the presence of smoke, creating a comprehensive and real-time picture of ambient conditions. By applying ML algorithms, the AI component can not only analyze

trends and detect anomalies but also identify early indicators of potential security or safety incidents, such as the outbreak of a fire or the release of hazardous materials. This capability significantly enhances situational awareness, as it allows stakeholders to move from reactive responses to proactive and predictive management. In doing so, AI functionality ensures that incidents are identified at the earliest possible stage, reducing risks to personnel, assets, and infrastructure while improving the overall resilience of the monitored environment.

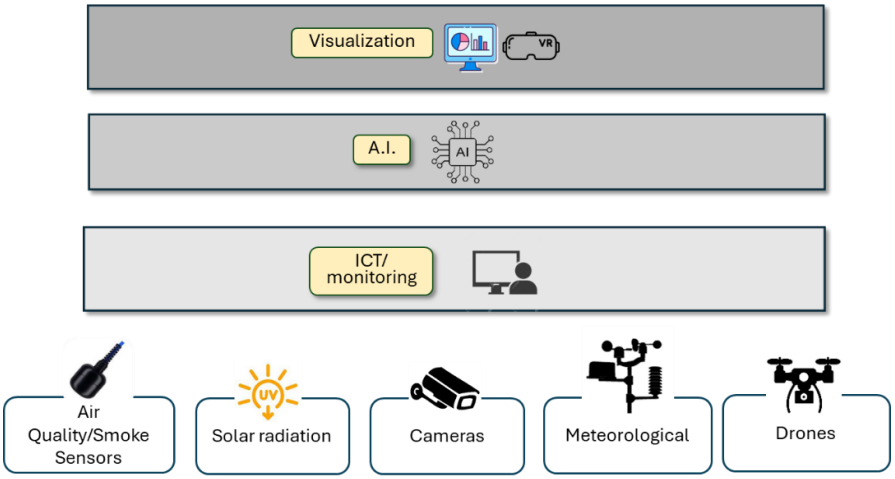

Figure 4-3 AI utilization in use case P1

The solution proposed for the management of AI/ML workflows in **E1** at the smart building level is based on an MLOps platform with an integrated ML model catalogue. This platform coordinates the pipelines for training, packaging, storage, deployment, continuous monitoring, tuning and re-training of E1 ML models dealing with single building scenarios. The high-level design of the software solution, implemented by Nextworks and largely based on open-source tools, is represented in Figure 4-4. On top of the MLOps platform, AIaaS APIs allow to access the internal ML model catalogue and provide simplified mechanisms to request AI tasks and manage AI pipelines in the smart building infrastructure.

The MLOps Platform supports the entire lifecycle of ML model development, deployment, and operation within a Kubernetes-based infrastructure, which in this case represents the edge infrastructure at the smart building. In order to interact with the platform, two access points are available for two actors identified as AI engineers and MLOps administrators. The AI Engineer interacts with the platform through the MLOps GUI for ML models uploading and training, and for running pipelines over a model as well as the operations associated to a model deployment. The MLOps Administrator manages the monitoring tools used for tracking the status for the platform and has unlimited access to the low-level tools for monitoring their execution. The core logic block is represented by the MLOps Manager block. This block integrates commonly-used ML libraries, like TensorFlow[2], Keras[3], PyTorch[4], Scikit-learn[5], for model development and training. MLFlow is used as ML Model Catalogue, for onboarding trained models and related metadata, including versions and other parameters related to the training and validation processes (see the information model in Figure 4-5 and Table 4-2). The Pipeline Manager uses Prefect[6], which serves as a workflow orchestration engine for specifying, scheduling, and handling ML pipelines. Pipelines include standard sequences of tasks such as data retrieval, model training and storage. During

[2] https://www.tensorflow.org/

[3] https://keras.io/

[4] https://pytorch.org/

[5] https://scikit-learn.org/

[6] https://www.prefect.io/

its tasks, a pipeline interacts with the MINIO[7] Artifact Storage for storing metadata and artefacts files. Similarly, the logic of the platform stores intermediate results, pipeline outputs, platform state in a relational SQL-based storage. In addition, the system includes a support registry, i.e., a Nexus repository for a versioned storage of container images, with their dependencies. Finally, the Infrastructure Management function interfaces with the Resource Orchestrator that handles resource provisioning and allocation in the Kubernetes infrastructure (see section 3.1.3).

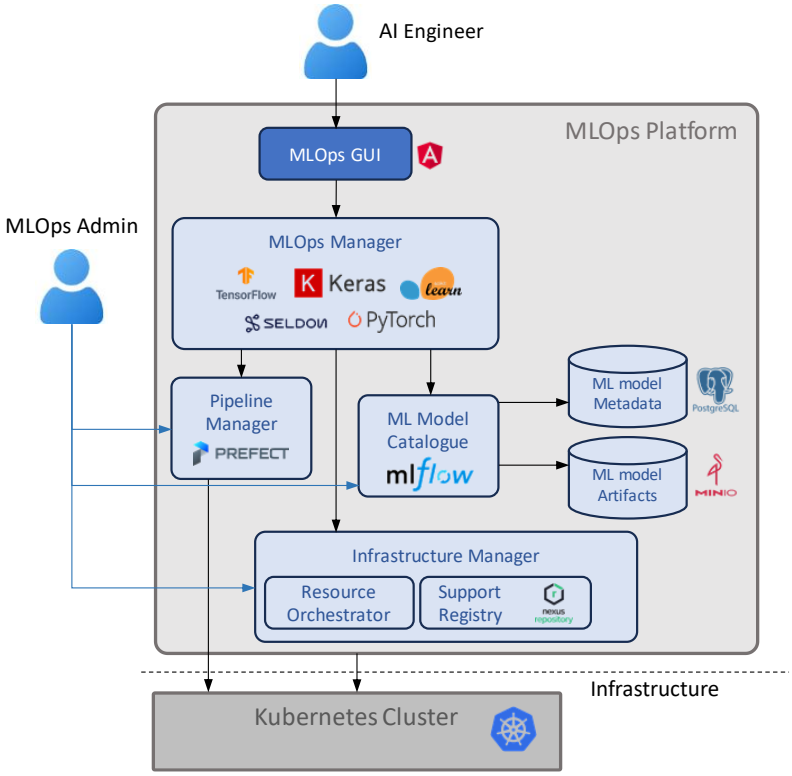

Figure 4-4 MLOps platform for E1 – Scenario for SB-EMS

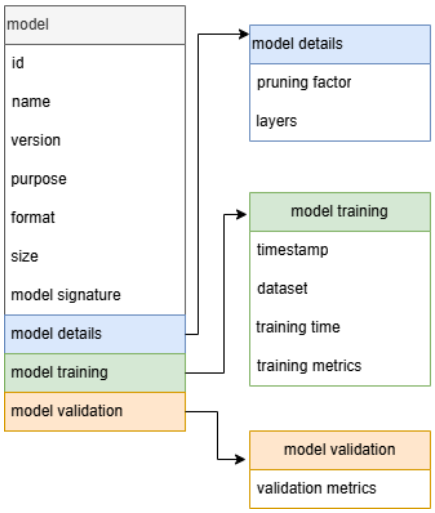

Figure 4-5 Information model for ML model metadata

7 https://www.min.io/

Table 4-2 Information model for ML model metadata

Field	Description
Id	Unique identifier for the stored model
Name	Human-readable name of the machine learning model.
Version	Specific version tag or number for the model.
Purpose	Intended task or application of the model (e.g., classification, object
Format	File format or serialization type used to store the model (e.g., .pth, .onnx).
Size	Total disk space used by the model artifact.
Model signature	Input/output schema or function signature of the model (e.g., input types/shapes).
Model details	General information and tags associated to the model like pruning factor or the layers. This information could be very useful for model query.
Model training	Information collected during training such as time for training, dataset used, computed metrics
Model validation	Performance metrics computed on validation data.

The second scenario of E1 (which will be implemented at a later stage of the project) focuses on energy management at the REC level, making use of FL techniques for the training of ML models through the collaboration of multiple buildings under the REC coordination. Smart buildings operate as clients of the federation while, at the REC level, a "parent" entity acts as an aggregator of the federation. As shown in Figure 4-6Figure 4-, the MLOps platform integrates components which are logically distributed at each building, responsible to handle the management of the FL clients and their operations, and components which are logically centralized at the REC level and they are responsible for the management of FL organization and procedures at the aggregator side.

The management communication between FL client and aggregator sides uses Federated Learning APIs, which extends the traditional AIaaS API with additional functionalities specific of Federated Learning (e.g., for clients to join a federation, for aggregators to advertise their capabilities, etc.). These APIs may be object of standardization. It should be noted that the representation in the picture is a logical one: the MLOps platform components related to the smart building may run in the smart building computing environment or on the cloud. However, placing the AI/ML workloads at the edge, i.e., in the smart building within the private network infrastructure, has several benefits, including lower latency, limitation of the traffic load towards the cloud and security of the data, which are maintained in the private environment.

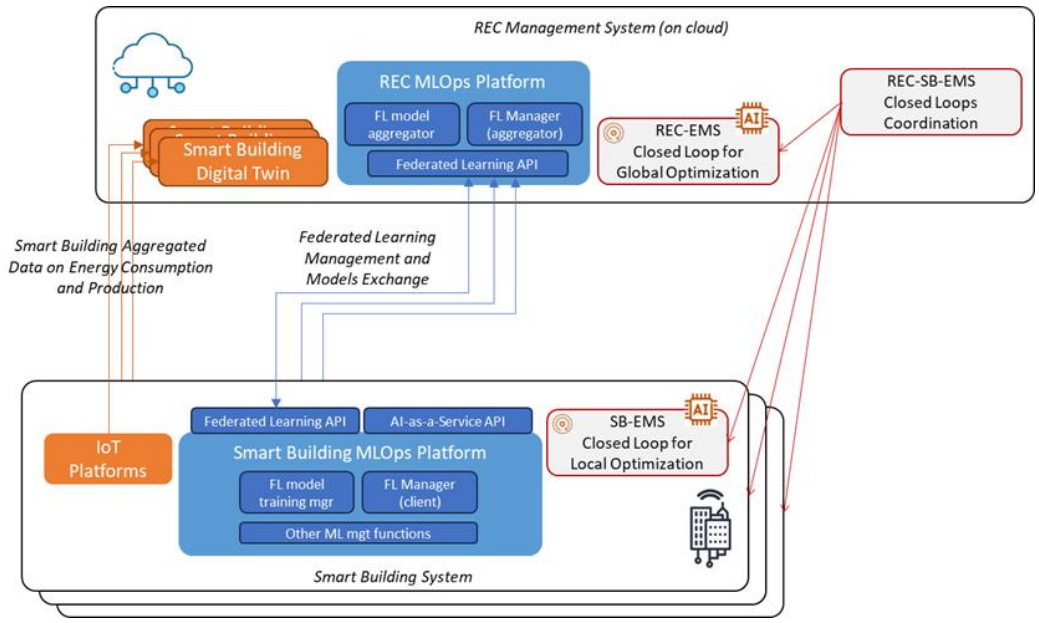

Figure 4-6 MLOps platform for E1 – Scenario for REC-EMS

In **E3**, at the initial stage, an AIaaS platform will be deployed and integrated with a centralized orchestration engine capable of collecting telemetry from all gateways (Figure 4-7). Once the setup is complete, data will be gathered from both the gateways and inverter performance metrics. In parallel, weather data APIs will be connected to provide real-time and forecasted environmental information, enriching the internal telemetry. With a reliable data flow established, forecasting algorithms, delivered through the AIaaS platform, will be developed and integrated into the orchestration engine. The resulting predictions will then feed into the control optimization logic. Following the initial deployment, the framework will transition into a continuous improvement cycle, ensuring ongoing refinement of performance and decision-making. Regarding the time planning, a preliminary development and integration of the enabler within the use case is expected by M17, while the third year foresees a full deployment of the use case scenario with the continuous improvement cycle in place.

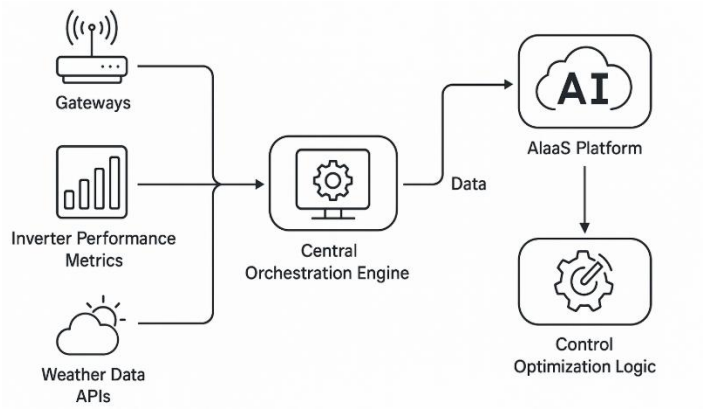

Figure 4-7 AI-aaS Platform for E3

T2 adopts the same MLOps platform already introduced for E1, in the single building scenario. In T2, the platform will be used for ML models capable of executing object detection, in order to trigger an alert to the police agents. Additionally, in this UC the MLOps platform will collaborate with the Resource Orchestrator managing computing resources on edge nodes and UGVs (as described in Section 3.1.3) to optimally distribute training and inference tasks between edge and UGVs for optimization of battery usage, service continuity and computational efficiency.

In T5, the AI process will begin with the systematic collection and cleaning of relevant data on the smart port environment, which will then be integrated into a Digital Twin to form a reliable representation of the operational environment. On this basis, predictive models and decision-making algorithms will be developed and trained to address the identified needs. Their performance will first be assessed through simulations in the Digital Twin, before being applied in real operational settings to ensure effectiveness and impact.

4.2 Digital Twin framework

This section describes the Digital Twin (DT) framework, covering both application-level and network-level Digital Twins, designed for AMAZING-6G use cases.

4.2.1 Description of the enabler

The Digital Twin Framework is designed to create a dynamic, virtual replica of real-world process flows, enabling operators to train, test, and deploy data-driven decision-making tools that reduce bottlenecks and improve operational efficiency. The development of such a framework begins with a comprehensive mapping of the process steps, including queues, arrival patterns, stochastic events, resource availability, and the behavior of key agents responsible for managing operations. This understanding is then translated into a simulation model using specialized software, providing a controlled environment where alternative strategies can be explored.

Within this environment, operators can experiment with different scenarios, such as testing infrastructure investments (e.g., adding an additional crane), evaluating process improvements (e.g., adjusting gate procedures to mitigate peak-hour congestion), or optimizing operational practices (e.g., routing straddle carriers more efficiently during high-demand periods). Historical data serves as the foundation for this development phase, allowing the Digital Twin to be calibrated, validated, and used as a training ground for predictive models and decision-making algorithms.

Once matured, the system evolves into a Live Digital Twin, where real-time data streams continuously update the virtual environment. This integration allows algorithms initially trained on the Development Digital Twin to be deployed in the field, enabling adaptive, evidence-based decision support and ensuring that operations remain efficient under dynamic and uncertain conditions.

In a network-oriented context, the Network Digital Twin (NDT) is focused on the collection of networking data including mobile Core Network and Radio Access Network. The NDT will also collect information about the computation (CPU, RAM, HD space, power consumption, etc.) to have a comprehensive view of the whole network including the IoT-edge-cloud continuum.

The (Network) Digital Twin is a centralized hub that can be accessed through dedicated APIs to retrieve data used by different types of applications but also by the orchestrator to close the loop towards the network and computation elements.

4.2.2 Use case association and contributing partners

Table 4-3 shows the mapping between Digital Twin framework and the AMAZING-6G use cases implementing it.

Table 4-3 Mapping between Digital Twin framework and AMAZING-6G UCs

	Digital Twin framework
T1	LINKS
T2	LINKS

T5	CERTH

In the **T1** use case, the DT is used to retrieve data from different devices with the scope of protecting visually impaired people crossing a junction. The DT collects data from the RSU and its sensors (camera and LiDAR), from the UEs and from the OBUs in the cars All this information can be easily retrieved through open APIs from the application that will compute possible risks. In case a risk is detected, a prompt warning is sent to the users and the other relevant actors involved (if any).

Moreover, an NDT is used for **T1** and **T2** to collect networking- and computation-related data from the network devices, including UEs. The collected information also concerns power consumption and will be used by the orchestration capabilities to make decisions about resource allocation, application migration (while keeping the context), and the maintenance of network slicing.

In the context of the Port of Thessaloniki in **T5**, the DT is being developed as a decision-support environment focused on optimizing straddle carrier operations. By replicating yard processes and container flows, the framework enables the testing of routing algorithms and operational strategies under realistic conditions. This includes evaluating alternative approaches such as infrastructure enhancements, process redesigns, or advanced decision-making algorithms. The ultimate goal is to identify the most effective solutions for minimizing delays, reducing resource inefficiencies, and improving overall terminal performance before their application in the real operational environment. The Digital Twin interacts with the underlying Compute-as-a-Service orchestration (Section 3.1.2), which dynamically provisions computing resources and manages containerized services, ensuring that the virtual and physical systems remain synchronized in real time. This integration supports operational efficiency, situational awareness, and sustainable port operations.

4.2.3 Design, development and implementation

The Digital Twin enabler provides the core capabilities to manage complex systems and streamline workflows, models and interactions across the edge–cloud continuum. Its architecture (

Figure 4-8) follows a modular pipeline, beginning with Communication Points that collect and exchange operational signals from heterogeneous sources. A Digital Twin-as-a-Service layer maintains the authoritative virtual state, exposes bidirectional update/notification flows and mediates access to twin entities. A Management Web App offers configuration, status visualization and administrative control, while a Data Storing & Management backbone persists time-series, events and reference data to support both live operations and historical analysis. Together, these components enable secure ingestion, continuous monitoring and controlled actuation through clear interfaces. On top of this core, an Intelligent Planning Suite hosts the analytics and decision services that consume twin state and feedback recommendations. A Simulation Digital Twin supports what-if experimentation and scenario execution while the Models Production tier operationalizes planning and forecasting models and a Simulation Management & Model Training capability governs model lifecycle, from calibration and validation to updates. These services interoperate through structured APIs with the DT-as-a-Service, allowing deterministic, reliable, and traceable exchanges of telemetry, metadata and control signals.

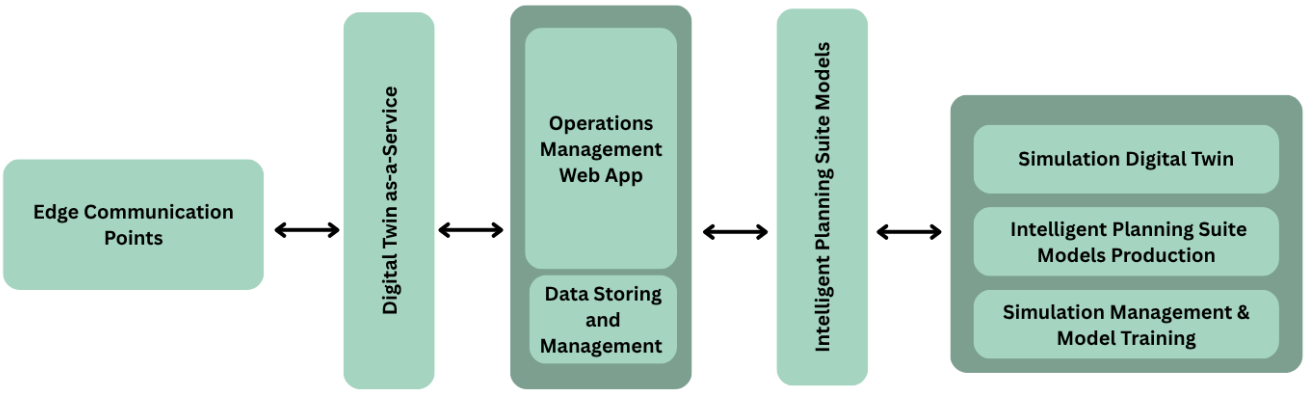

Figure 4-8 Digital Twin Framework – high-level architecture

In **T1** and **T2**, the Digital Twin will be instantiated through the implementation of the use cases and will start collecting data as soon as the UEs and the field devices are ready. Since the DT builds on previous projects, it will be available from the very beginning and will be extended according to the needs of the project. For the Network Digital Twin, an initial phase of requirements gathering and analysis will be carried out, building on the work already done for the monitoring enabler. Furthermore, the use of this information to enable closed-loop control for network slicing will be investigated.

In **T5** implementation phase follows a structured process that begins with detailed process mapping and the identification of key actors and their respective roles. Data collection and validation provide the foundation for integrating models within the Digital Twin environment. Once integrated, the system is used to conduct experiments and train predictive algorithms under realistic scenarios. The outcomes are then assessed against predefined KPIs, ensuring that the solutions deliver measurable improvements in efficiency, reliability, and resource utilization.

The diagram in Figure 4-9 illustrates the multi-layered architecture of the digital twin framework designed for port optimization. The framework streamlines the interactions between development, training, management, and operational layers. At its foundation it integrates simulation environments, optimization algorithms, and real-time data flows to provide a continuous learning and decision-support pipeline. The aim is to enhance port performance by combining historical knowledge, scenario-based experimentation, and live operational data into a coherent digital ecosystem capable of adaptive optimization under stochastic conditions.

In the development layer, three main blocks form the core inputs range from historical port operations data to optimization algorithms and scenario building. Historical data provides statistical distributions and empirical patterns that characterize the port environment, such as vessel arrivals, container flows, and equipment utilization. Optimization algorithms, such as those targeting straddle carrier routing or yard crane scheduling, are embedded within this layer to explore optimal strategies. Scenario building extends this process by enabling the creation of synthetic yet realistic operational conditions to test the robustness and scalability of proposed solutions. The outputs of this layer converge to feed the training digital twin.

The training layer hosts the simulation-driven digital twin, which acts as an environment for experimentation. Here, optimization models and scenarios are tested under controlled, data-enriched conditions. Through iterative experimentation, the system extracts agents that are able to apply in real field strategies, rules, or algorithmic configurations that demonstrate superior performance under diverse operational settings. This layer, therefore, bridges experimental findings with actionable decision analytics, ensuring that algorithmic advances are validated before live deployment.

Moving to the port management layer, the framework incorporates live operational data from ongoing port activities. This data feeds directly into the live digital twin, and through a web-based application

closely integrated with real port operations the Port Authority is able to implement pretrained optimal resource allocation plans. Unlike the training environment, the live twin is continuously updated with real-time inputs, ensuring alignment with the dynamic nature of day-to-day port processes. By embedding the agents derived from the training layer, the live twin operates not only as a monitoring tool but also as a prescriptive decision-support system capable of adjusting strategies in near real-time.

Finally, the operations layer represents the direct interface with physical port processes. Here, the live digital twin exchanges information with actual operations, enabling a two-way interaction; 1) operational data is fed back into the digital ecosystem, while 2) optimized strategies and recommendations are deployed into practice. This feedback loop ensures a continuous cycle of monitoring, adaptation, and improvement, anchoring making the digital twin a dynamic tool for both tactical management and long-term optimization of port logistics.

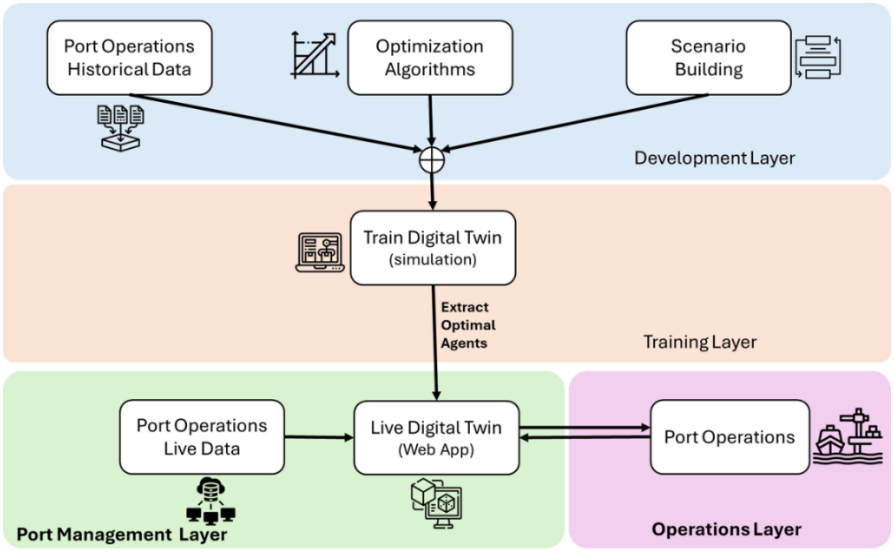

Figure 4-9 Multi-layered architecture of the digital twin framework for port optimization (T5)

4.3 OpenAPIs for Network Exposure

One of the key technological topics within WP3 is the exposure of advanced network and computing capabilities to external applications through standardized and developer-friendly APIs. This approach aligns with the broader vision of enabling intelligent, flexible, and interoperable solutions across multiple verticals, supporting the project's core objectives related to sustainability, infrastructure readiness, and trustworthiness. Network exposure refers to the standard and secure publication of network functionalities (such as quality of service (QoS) guarantees, energy management, or resource allocation) to external consumers via open interfaces. This capability enables the possibility to realize innovative 6G use cases identified in WP2, in the domains of Public Safety, Transport, Energy, and Smart Cities, where application behavior must adapt dynamically to underlying network conditions or constraints.

Network exposure builds upon a foundation of ongoing standardization efforts and industry initiatives. The **3GPP** specification for the **Network Exposure Function (NEF)** (e.g., 3GPP TS 29.522 [13]) defines how core network capabilities can be exposed securely to third-party applications, particularly within 5G system architecture. In parallel, the **CAMARA** project[8], a joint initiative between the Linux Foundation and **GSMA**, is developing developer-centric APIs to expose advanced network capabilities (such as

[8] https://camaraproject.org/

quality on demand, energy efficiency, and edge resource selection) across diverse operator networks, with alignment to GSMA Operator Platform principles [14].

Next sections focus on four main categories of OpenAPIs that are being adopted and extended within the project:

- **TM Forum APIs**: These APIs provide models and interfaces for service and resource management, order handling, and customer experience. They are especially useful in scenarios involving orchestration of services across multiple parties and administrative domains. Within the AMAZING-6G project, TM Forum APIs serve as the element for building interoperable platforms capable of managing complex service lifecycles across cloud, edge, and network environments.
- **CAMARA APIs for Quality on Demand**: This set of APIs enables applications to request specific QoS levels from the network (for example, to support real-time video, industrial control, or emergency response services). By abstracting the complexity of the underlying 5G/6G infrastructure, these APIs allow developers to define service-level requirements in a simple, declarative manner.
- **CAMARA APIs for Energy Management**: These APIs allow applications and services to get insights into energy consumption and to control their behavior accordingly, for example, by offloading tasks to more energy-efficient nodes, or reducing activity during peak power demand. These APIs also support edge computing scenarios where local energy constraints must be considered.
- **CAMARA APIs for Edge/Cloud Continuum**: As compute resources become increasingly distributed across user devices, edge nodes, and cloud platforms, there is a need for standardized APIs that can support dynamic and location-aware deployment of applications and services. These APIs facilitate interactions with the edge-cloud continuum, including service discovery, resource allocation, and deployment of application components.

The following sections will be devoted to describe the technologies and the enablers that expose the APIs, as well as their adoption in the various Use Cases.

4.3.1 Description of the enabler

TM Forum API

TM Forum APIs represent a comprehensive suite of standardized application programming interfaces designed to enable digital transformation and automation across the telecommunications industry. These APIs follow the **Open Digital Architecture (ODA)** framework and provide a consistent, interoperable approach to managing complex telecom operations, from customer management and product catalogs to service ordering and network inventory. The APIs are built around key business entities and processes, offering standardized data models and RESTful interfaces that allow telecom operators, vendors, and service providers to integrate systems more efficiently. By adopting these standards, organizations can reduce integration complexity, accelerate time-to-market for new services, and create more agile, cloud-native operational environments.

The TM Forum API portfolio covers the entire telecom value chain, including customer experience management, service and resource management, partner and supplier relationship management, and enterprise effectiveness functions. These APIs enable capabilities such as automated service fulfillment, real-time billing and charging, dynamic product catalog management, and seamless partner ecosystem integration. The standardized approach helps break down traditional operational silos and supports the creation of more flexible, API-driven architectures that can adapt quickly to changing market demands and customer expectations.

CAMARA APIs for Quality on Demand

The Quality on Demand (QoD) API from CAMARA exposes a programmable interface for vertical applications to request prioritized and quality-assured data flows, such as reduced jitter or enhanced throughput, without requiring the developer to understand the complexities of mobile networks and their interfaces. QoD API exposes a set of northbound APIs aligned with the CAMARA Quality-On-Demand specification, enabling applications to create, manage, and delete QoS sessions for specific traffic flows.

Applications define the target flow by specifying the device, application server, and optional port details. A catalog of QoS profiles is available, allowing developers to choose profiles (e.g., low latency, high throughput) that match their application's needs. Upon receiving a request, the enabler translates this intent into network-level instructions, e.g., via the 5G Network Exposure Function (NEF), interacting with core components like the Policy Control Function (PCF) and optionally the User Plane Function (UPF) to enforce the desired QoS behavior.

The QoD enabler supports event notifications, such as provisioning or session expiration, through callback URLs using the CloudEvents format. It uses OAuth 2.0 for secure access and can be integrated with CAPIF for access control and API discovery.

The core functionalities of QoD include:

- **QoS Session Management:** Create, query, and delete QoS sessions for specific App-Flows between device and application server.
- **QoS Profile Abstraction:** Abstracts network-level QoS configurations into developer-friendly profiles.
- **Session Duration and Control:** Enables control over the lifespan of the QoS session with optional early termination.
- **Event Notifications:** Supports subscription to status updates via a secure callback mechanism.
- **Secure and Open Access:** Utilizes OAuth 2.0 for authentication and may integrate with CAPIF for access control and API discovery.

CAMARA APIs for Energy management

The CAMARA Energy Footprint Notification (EFN) API provides details about the end-to-end energy consumption and carbon footprint of a service delivered by multiple application instances. This scenario envisions a service offered by numerous applications running in various locations within the telecommunications operator cloud. The EFN API supplies the API consumer with information regarding the overall end-to-end energy consumption and carbon footprint (greenhouse gas emissions) generated by the application instances hosted by the telco operator. The reported energy consumption and carbon footprint account for the energy used to operate the application instances in data centers and the energy consumed throughout the operator's network to deliver the service. With this knowledge, the operator can configure e.g. the network or services to run in energy-savvy state for defined period of time.

CAMARA APIs for Edge/Cloud

The CAMARA Edge Cloud project defines a set of APIs that enable vertical services to access telco edge computing resources that are close to user devices to meet the requirements of low-latency, high-performance applications. It supports edge discovery, application deployment, and traffic routing. The edge discovery set of APIs (i.e., simple, optimal, and end-point edge discovery) offers the possibility for verticals to discover the most suitable edge zones according to specific resource requirements (e.g., CPU/GPU, memory, volume) of a given application. The Edge Application Management API allows developers to manage application workloads on telco edge infrastructure, supporting operations such as deployment on a particular edge zone, lifecycle management (start/stop instances), and resource scaling across edge zone environments. Finally, the Traffic Influence API allows verticals to optimize the traffic routing from device towards the application deployed at a given edge zone.

TM Forum APIs and GSMA CAMARA Integration

In the context of GSMA's CAMARA initiative, TM Forum APIs serve as the essential "Operate APIs" layer that provides the operational foundation and business support capabilities required to deliver network services exposed through CAMARA's network-facing APIs. While CAMARA APIs focus on exposing specific network capabilities like quality-on-demand, device location, and network slicing directly to developers and enterprise applications, TM Forum APIs handle the critical operational processes that make these network services commercially viable and operationally sustainable. This includes service lifecycle management, customer onboarding and management, billing and revenue management, service level agreement monitoring, and partner relationship management.

The architectural relationship positions TM Forum APIs as the operational orchestration layer that sits behind CAMARA's network exposure APIs, ensuring that when a developer or enterprise customer consumes a CAMARA network service, all the necessary business processes are properly managed. For example, when a CAMARA API delivers quality-on-demand services to an application, TM Forum APIs handle the service ordering, provisioning workflows, usage tracking, billing calculations, and customer notifications. This integration creates a complete end-to-end solution where CAMARA provides the standardized network service interface while TM Forum APIs ensure operational excellence, regulatory compliance, and business process automation. Together, they enable telecom operators to transform network capabilities into monetizable, developer-friendly services while maintaining the operational rigor required for carrier-grade service delivery.

4.3.2 Use case association and contributing partners

Table 4-4 shows the mapping between Network APIs and the AMAZING-6G use cases implementing them. This enabler may also be applicable to other use cases, but this is not covered in this project.

Table 4-4 Mapping between Network APIs and AMAZING-6G UCs

	TM Forum APIs	CAMARA APIs Quality on Demand	CAMARA APIs Energy management	CAMARA APIs Edge/Cloud
H1		TNO		TNO
H2		TNO		TNO
P1	UPAT			
P2		VTT	VTT	
P3		VTT	VTT	
P4		VTT	VTT	
E2		TNO		TNO
T1		NXW, LINKS		NXW, LINKS
T2		NXW, LINKS	NXW, LINKS	NXW, LINKS
T4		IMEC. TUC		IMEC, TUC

In the **H1 use case**, AI-based image analysis component continuously collects health monitoring data from the patient and may be deployed at the edge near the measurement location for network latency processing efficiency. To achieve that, the clinical backend makes use of the CAMARA Edge Cloud APIs, in particular, the Edge Application Management API, to dynamically discover and deploy the AI-based image analysis component in the most appropriate edge location. Finally, to guarantee that the network QoS requirements for the application are met when collecting the patient's data, the CAMARA Quality-on-Demand (QoD) is used.

Similarly to H1, the AI-based image analysis and adaptation of pacing component in the **H2 use case** may be deployed at the edge to optimize network latency and processing for timely data collection and adaptations. The clinical backend uses the CAMARA Edge Cloud APIs to discover and deploy the AI-based image analysis component in the most appropriate edge location. In addition, the CAMARA QoD API is used to guarantee the application's network QoS requirements.

In the **P1 use case**, TM Forum Open APIs are required to enable standardized, cross-operator orchestration of network slices and compute resources, allowing PPDR agencies to dynamically deploy, scale, and assure AR/VR mission-critical services across the edge-cloud continuum in a vendor-neutral and automated way. In P1 specifically where PPDR slices may span multiple operators, and resources (network slices, MEC nodes, cloud compute) may belong to different administrative domains through TM Forum Open APIs (e.g., Service Ordering, etc) inter-operator communication using standardized models, is enabled. To that respect in P1 by enabling interoperability one ensures that PPDR services can request and provision resources across any operator seamlessly. This enabler will provide a single logical interface for orchestrators to request both network and compute resources in a unified way supporting catalog-driven service orders, allowing to Instantiate edge compute clusters on demand and trigger dynamic slice modification to increase bandwidth or latency guarantees in real time. Especially since P1 aspires to prove that PPDR agencies that often work with multi-vendor, multi-operator environments are assisted by this enabler to assist strict URLLC and compute resource guarantees.

In the **P2 use case,** QoD API is used to ensure QoS requirements are met when having seamless connectivity in between different connectivity solutions for the search and rescue team. This is done together with network performance and monitoring enabler. In addition, predefined QoS profiles enable tailored configurations for different latency and throughput needs. Energy management API can be used for monitoring and selecting energy-savvy configuration for the used network and services in order to reduce carbon footprint.

In the **P3 use case,** the analytics framework utilizes QoD and Energy management APIs for assessing the integrated and independent private network performance both from QoS and energy perspective. QoD is used for guaranteeing QoS i.e. by using slicing with suitable configuration resourcing.

Similarly as P2 and P3, the **P4 use case** uses QoD and Energy management APIs for optimizing both QoS and energy usage for the search and rescue team in the field. P4 integrates the efforts done in P2 and P3 for example by selecting wisely the used wireless network prioritizing the connection, or maximizing the throughput based on real-time network analytics. On the other hand as the equipment and services are relying on battery-powered devices it is essential to find and use energy-savvy parametrization both in RAN as well as in applications that try to maximize the throughput and minimize energy consumption.

In the **E2 use case,** the Digital Twin component collects and processes real-time sensor data received from the drones that perform wind turbine inspection and may be deployed at the edge for network latency processing efficiency. To achieve that, the on-shore control support invokes the CAMARA Edge Cloud APIs, in particular, the Edge Application Management API, to dynamically discover and deploy the Digital Twin component in the most appropriate edge location. Finally, to guarantee that the QoS requirements for the application are met when collecting the sensor data from the drones, the CAMARA QoD is used.

The **T1 and T2 use cases** adopt CAMARA API for QoD, energy management, edge servers discovery and edge application management. The exposure of these interfaces allows to automatically orchestrate the service at runtime depending on application-level events and conditions. For example, a service profile with high uplink data rate is requested using QoD API when the UGV needs to send HQ video streaming to the edge. Energy management API is used by the monitoring platform to collect data about battery level, power consumption and solar panel energy generation in UGVs and RSUs for decision making on task offloading. Finally, task offloading execution procedures are handled interfacing with the system using the Edge Cloud API, to select the most suitable edge node and deploy the application components there.

In the context of the **T4 use case**, the CAMARA QoD API plays a critical role in ensuring that the network can meet the stringent performance requirements of tele-operation while also supporting efficient resource usage. When the ML-based anomaly detection service identifies that autonomous driving is no longer reliable, due to factors such as unclear road markings or sensor degradation, it triggers a switch to tele-operation. At this point, the ML service, wrapped as a vertical application, uses the network-exposed CAMARA QoD API to dynamically request a network quality upgrade from the 5G/6G core. This request specifies enhanced QoS parameters such as increased capacity (uplink/downlink) to support real-time video streaming, sensor data transmission, and vehicle control signals. The network exposure layer translates this request into actionable configurations, ensuring that the end-to-end network slice is reconfigured to support the critical requirements of the tele-operation session. Once the situation is resolved and the vehicle returns to autonomous mode, a new request is sent via the same API to scale down the allocated network resources. This dynamic, context-aware adaptation ensures not only operational continuity and safety during human intervention but also optimal utilization of network and compute resources during normal autonomous driving.

4.3.3 Design, development and implementation

This section presents the technical design, development approach, and implementation plan for the OpenAPIs for Network Exposure enabler, with reference to its role within the project's use cases and associated testbeds. The enabler provides standardized interfaces for exposing network and compute capabilities through TM Forum and CAMARA APIs, enabling dynamic and secure interaction between applications and infrastructure components, as shown in Figure 4-10.

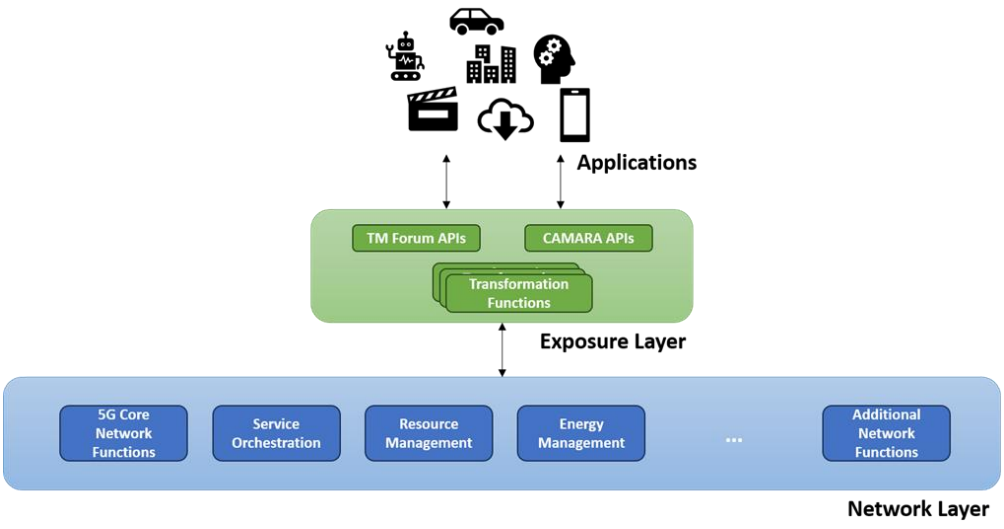

Figure 4-10 Network API Exposure architecture

This section describes the design activities, focusing on the technical architecture of the enabler, including the integration of functional components, API definitions and interactions with network functions, orchestration platforms, and resource management systems, as well as the planning activities, focusing on the necessary steps for development, integration, and deployment with the

implementation plan tailored to the specific contexts of the use cases and testbeds. Inputs have been collected based on the use cases and the corresponding testbed environments where the enabler will be deployed.

The subsequent subsections provide detailed descriptions of the design and implementation plan for each sub-enabler (e.g., CAMARA APIs for Quality on Demand, Energy Management, Edge/Cloud integration, TM Forum APIs), structured around their respective use cases and testbed environments.

In the **H1 use case,** CAMARA Edge Cloud and QoD network APIs are integrated into the 6G system as a higher 'Network APIs' layer working on top of the TNO B5G core (Figure 4-11). The clinical backend invokes CAMARA Edge Cloud APIs to discover and deploy the AI-based image analysis component at the optimal edge server. After the AI-based image analysis component is instantiated and starts collecting data from the patch, the clinical backend requests QoS guarantees on-demand when collecting health patient's data by triggering the QoD API.

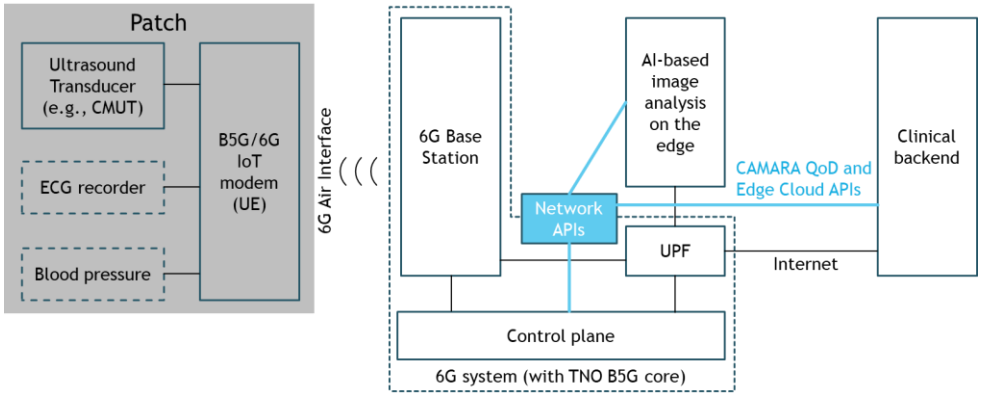

Figure 4-11 High-level architecture of H1 integrated with CAMARA Edge Cloud and QoD APIs

Similarly to H1, in the **H2 use case** the CAMARA Edge Cloud and QoD network APIs are integrated into the 6G system as a higher 'Network APIs' layer working on top of the TNO B5G core (Figure 4-12). The clinical backend invokes CAMARA Edge Cloud APIs to discover and deploy the AI-based image analysis component at the optimal edge server and later it requests the QoD API to guarantee network QoS when collecting health patient's data.

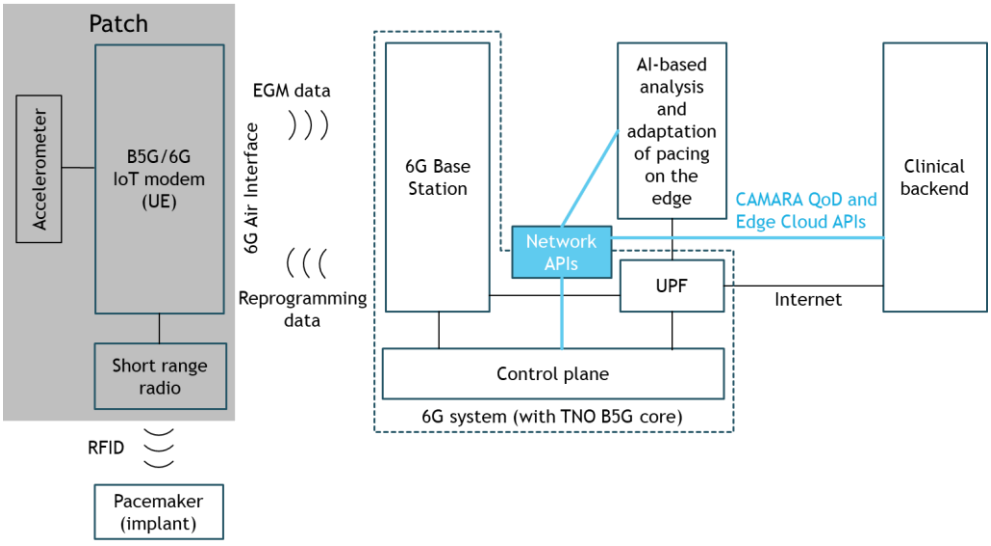

Figure 4-12 High-level architecture of H2 integrated with CAMARA Edge Cloud and QoD APIs

In Patras5G testbed, TMF APIs among others, are used in various instances when configuring the network and orchestrating services, especially in the context of the **P1 use case**. Here, in order to achieve the requested KPIs, experimentation requires automatic orchestration or resources enhanced by the

enablers that have been or are being developed for the scenarios under consideration. To be more specific however, in Patras5G testbed ETSI OpenSlice, is used for resource management and orchestration. In OpenSlice, a developer can define specifications for underlying resources. These specifications can be categorized accordingly, and they get exposed to a resource catalogue which follows the TMF634 Resource Catalog. When required, the TMF639 Resource Inventory API mechanisms are triggered to create a new resource according to its specification. To accommodate external services, resources are handled as services and are used to offer Resource as a Service (RaaS). To achieve RaaS a Resource Facing Service (RFS) is created based on the TMF model for service specifications (TMF633 Service Catalog). Other TMF APIs are also supported. The sequence diagram in Figure 4-13 shows how compute and network resources are exposed and how the developed enablers allow for timely and efficient orchestration of resources.

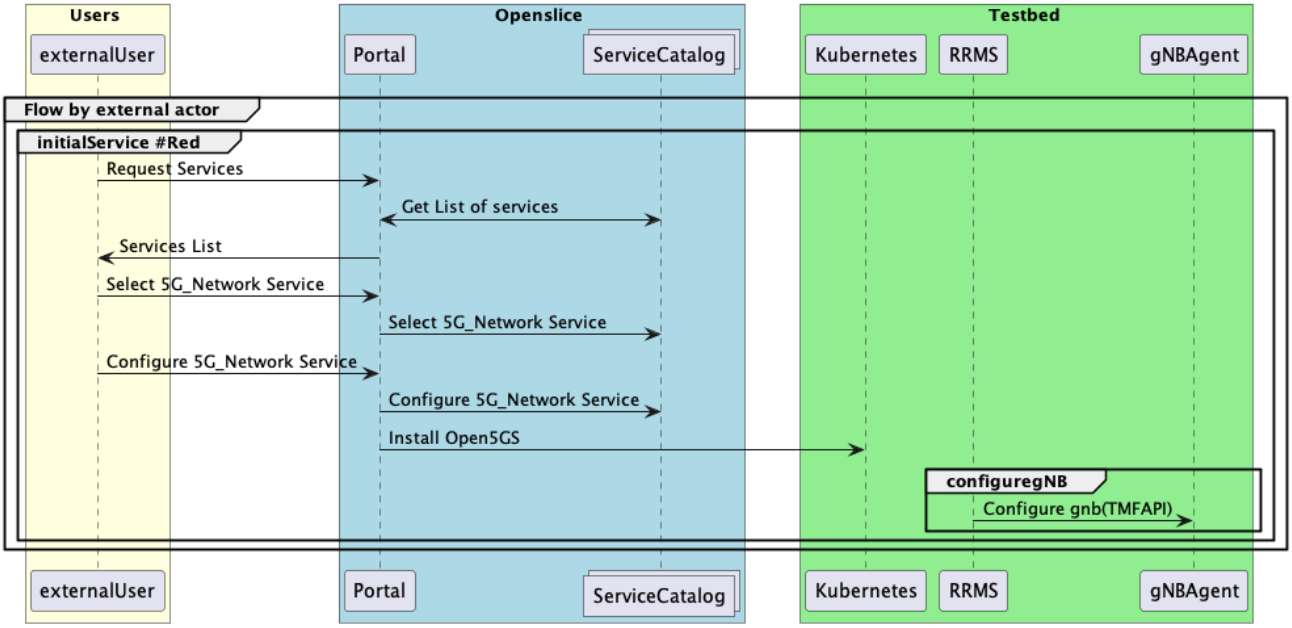

Figure 4-13 Example of network exposure and resource management through TMF APIs in P1 (Patras5G testbed)

In the **P2, P3 and P4 use cases**, the Network and Energy management APIs can be integrated into the Finnish B5G test network and work closely with the network performance, monitoring and control enabler. These APIs and high-level architecture are presented in Figure 4-14 and can be used in common for all the P2-P4 use cases. Through the controller, the Network API can set the configuration parameters of the used network based on the monitoring information of the network performance and energy usage.

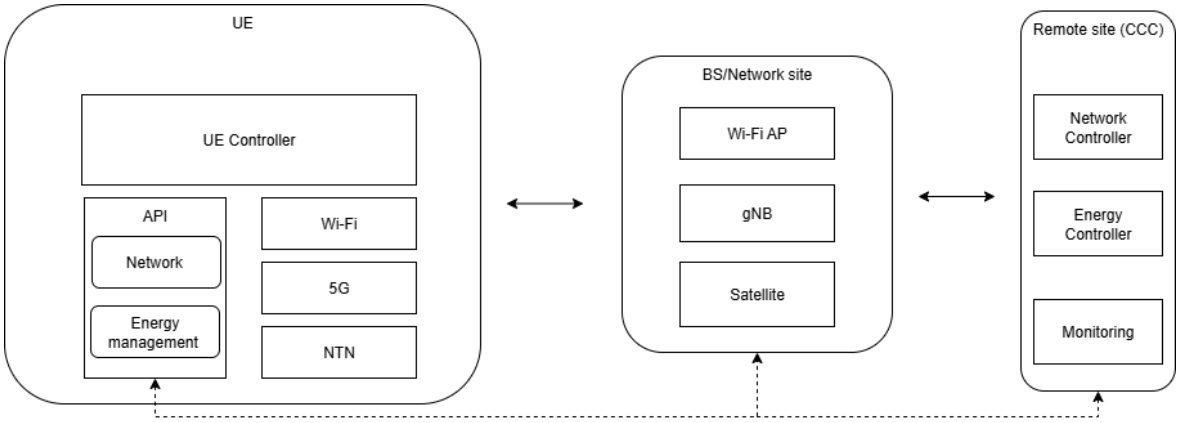

Figure 4-14 High-level architecture of APIs integration to Finnish use cases P2-P4

In the **E2 use case,** the CAMARA Edge Cloud and QoD network APIs are integrated into the 6G system as a higher 'Network APIs' layer working on top of the TNO B5G core (Figure 4-15). The on-shore control support invokes CAMARA Edge Cloud APIs to discover and deploy the Digital Twin component at the optimal edge server. At a later stage, the on-shore control support component requests QoS guarantees on-demand by triggering the QoD API for the drone's sensor data to be collected by the Digital Twin component during the inspection of the wind mill blades.

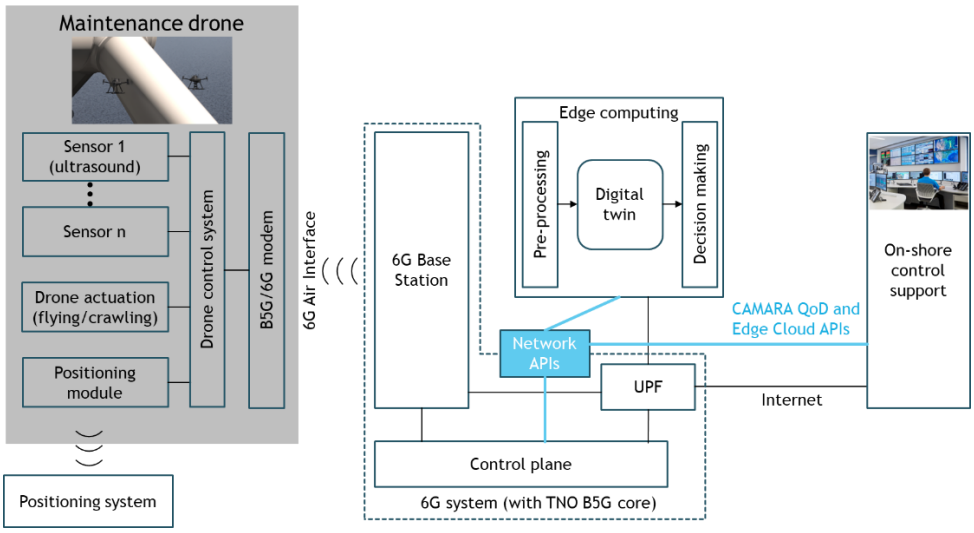

Figure 4-15 High-level architecture of E2 integrated with CAMARA Edge Cloud and QoD APIs

Use Cases **T1** (Protection of Vulnerable Road Users) and **T2** (Enhancing Urban Security with UGV Monitoring) will be deployed within the Italian Infrastructure, leveraging a CAPIF-based Exposure Framework for CAMARA APIs, whose architecture is illustrated in Figure 4-16. The aim is to enable access to network and infrastructure capabilities through standardized, secure interfaces.

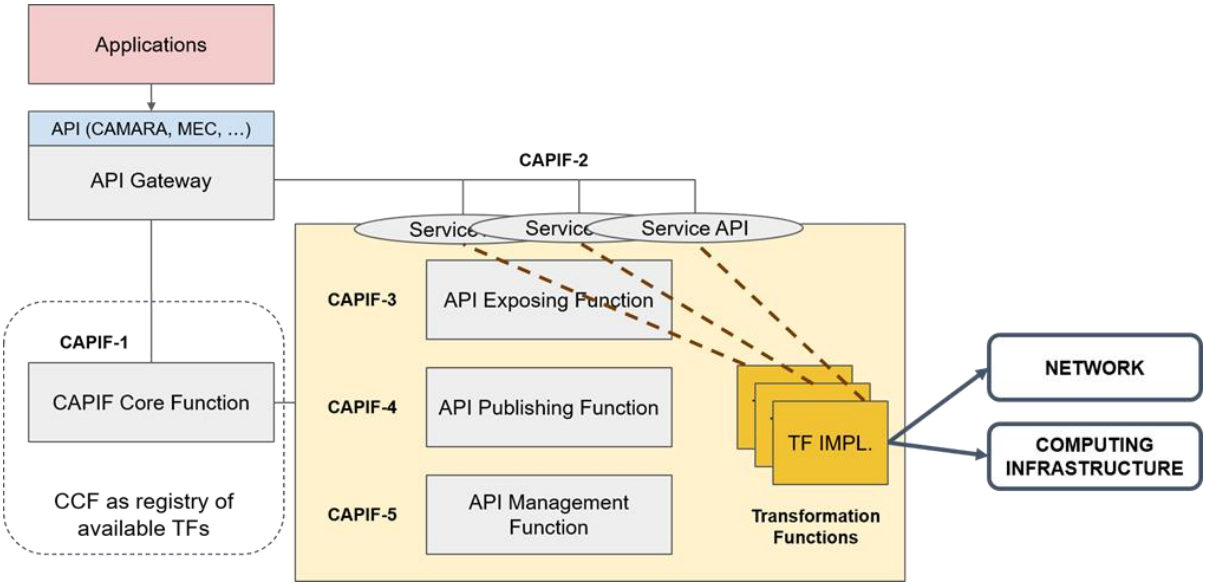

Figure 4-16 Exposure Framework architecture for T1 and T2

The Exposure Framework enables secure and standardized access to network and computational resources via open APIs. It acts as the interface between application developers and the underlying 5G/6G infrastructure, supporting application deployment, orchestration, and QoS management across different domains like smart cities and transport.

At its core, the architecture leverages the CAPIF (Common API Framework) specification to host a registry of available transformation functions for the APIs exposed by the API. The main components are:

- **Applications:** Consumer-side entities interacting with the network via open APIs (e.g., CAMARA, MEC).
- **API Gateway:** The point of entry for all application requests, which routes them to the appropriate backend service through standardized APIs.
- **CAPIF Core Function:** Provides API discovery, authentication, authorization, and usage control.
- **API Exposing Function (CAPIF-3):** Publishes service APIs offered by backend transformation functions (TFs).
- **API Publishing Function (CAPIF-4) and Management Function (CAPIF-5):** Handle service registration, cataloging, and lifecycle.
- **Transformation Functions (TFs):** Backend services that simplify access to network and computing infrastructure.

Upon receiving an API request from an application, the API Gateway is responsible for selecting the most appropriate Transformation Function (TF) to handle the request. This selection is based on multiple factors, such as the requesting user's identity or role, the context of the request (e.g., location, network conditions), the specific API being invoked, and the capabilities or constraints of the underlying infrastructure. To perform this selection, the API Gateway queries the CAPIF Core Function, which maintains a dynamic registry of all available Transformation Functions along with their associated metadata, policies, and service-level attributes. This enables the system to route each request to the TF instance that best matches the application requirements and current network context.

In T1, the exposure framework helps manage the lifecycle (onboarding, deployment, termination, migration, reconfiguration) of applications, based on real-time information like the battery level of Road Side Units (RSUs), solar power availability, and application resource needs. Using CAMARA Edge Cloud APIs, the orchestration system can offload tasks, such as object detection or risk analysis, between RSUs and edge servers to save energy and maintain service quality. In T2, the exposure framework

enables application handover between the UGV and the edge while keeping the application context. When needed, the UGV can offload scene recognition tasks to the edge to save battery.

The **T4 use case** implements the CAMARA QoD API to dynamically enhance QoS parameters for critical service flows such as vehicle tele-operation. This implementation is based on open standards and leverages both open-source core and RAN components, providing a representation of how the CAMARA QoD API should be designed and implemented in a 5G network.

The implementation adheres to 3GPP standards, including: TS 23.502 – Procedures for PDU session management; TS 29.244 – PFCP procedures for UPF; TS 29.518 – AMF interface procedures; TS 38.413 – NGAP procedures for RAN reconfiguration; TS 38.331 – RRC signalling for QoS reconfiguration.

The logic and API design follow the CAMARA open-source project developed by Deutsche Telekom (i.e., the NEF, available in the public repository), which is publicly available via the CAMARA GitHub repository. The following technologies are used:

- **Core Network:** We created a custom branch in Open5GS, an open-source 5G Core implementation. This branch includes extensions to support CAMARA QoD API integration, including NEF, PCF, SMF, and PFCP logic.
- **RAN:** We use OpenAirInterface (OAI) to emulate a realistic 5G RAN environment (gNB + UE). It supports RRC and NGAP procedures for dynamic QoS adaptation.
- **API Type:** The CAMARA QoD API is a RESTful API, operating over HTTP(S), accepting POST requests with CAMARA JSON payloads. The API endpoints conform to CAMARA OpenAPI specifications and follow standard REST design principles (resource-based, stateless interactions, HTTP status codes).

The API call flow is illustrated in the message sequence chart in Figure 4-17, and each flow is explained below:

1. Application (AS Server) initiates a POST "/sessions" to CAMARA QoD API, requesting QoS enhancement for a specific application flow, specifying source/destination IPs, ports, and QoS profile.

2. The CAMARA layer forwards this as a POST "/3gpp-as-session-with-qos" to the Network Exposure Function (NEF), which validates and authenticates the request within 5G network.

3. Upon successful validation, the NEF triggers a POST "/npcf-policyauthorization/app-sessions" request to the Policy Control Function (PCF). This initiates a new QoS policy session, generates a monitoring policy for the specified service data flow, and triggers the PDU Session Modification Procedure as per ETSI TS 123 502. Importantly, the response returned to the AS Server confirms request acceptance, but not enforcement.

4. The PCF notifies the Session Management Function (SMF) via POST "/nsmf-policy notify/update", instructing it to apply the updated policy and initiate core network changes.

5. The SMF sends a PFCP Session Modification Request to the User Plane Function (UPF), installing updated data-plane rules (PDRs, FARs, QERs) according to the new policy (ETSI TS129 244).

6. Concurrently, the SMF triggers RAN-side reconfiguration by sending an N1N2 Message Transfer to the Access and Mobility Management Function (AMF) (POST "/namf-comm/uecontexts/<id>/messages"), which delivers a message to the UE via the gNB.

7. The AMF initiates the PDU Session Resource Modify Request (NGAP) toward the gNB, requesting it to apply updated QoS configurations to the RAN. This includes QoS Flow setup/modification using the QoS Flow Add or Modify Request List IE.

8. The gNB applies the new QoS settings at the radio interface using the RRC Reconfiguration Procedure, which reconfigures the UE's RRC connection with appropriate radio bearers andflow characteristics (ETSI TS 138 331).

9. The SMF updates the PDU session context (/nsmf-pdusession/modify) with the outcome of the operation, including any new tunnel or QoS flow parameters established during RAN reconfiguration.

10. Finally, a PFCP Session Modification is sent by the SMF to the UPF to finalize and align user-plane state, including Forwarding Action Rule (FAR) and tunnel parameters for the new flow.

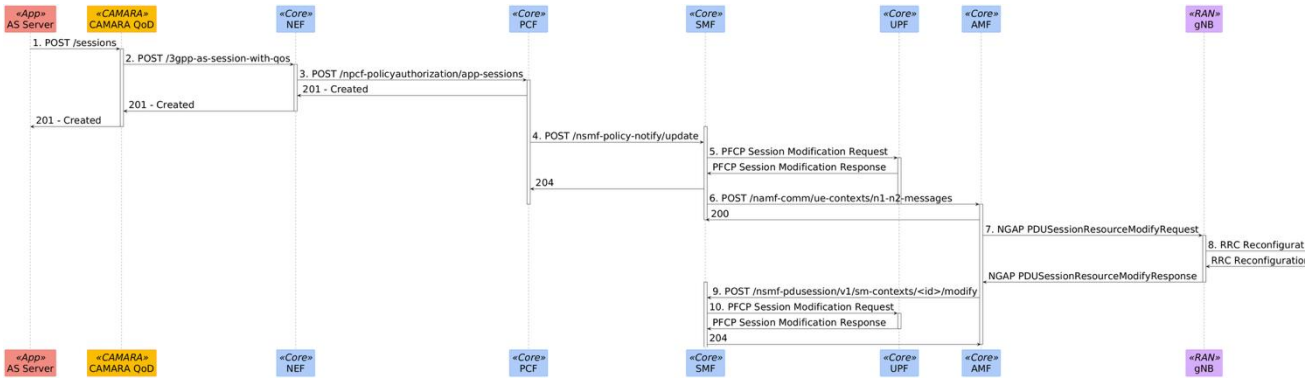

Figure 4-17 API Call Flow for CAMARA QoD API

Figure 4-18 illustrates intermediate results of the current CAMARA setup in T4 evolution of application throughput over time for two (e.g., two teleoperated trucks) in the downlink direction. Initially, both traffic flows start independently and gradually ramp up. By approximately the 20-second mark, each UE reaches its defined Service-Level Objective (SLO), operating at the minimum guaranteed throughput.

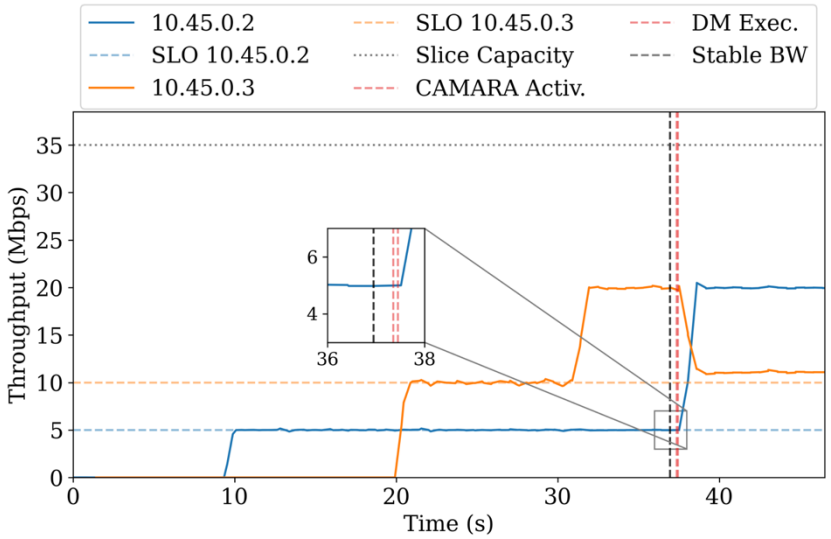

Figure 4-18 Preliminary results of QoD API activation in the context of T4 use case

At around 32 seconds, a CAMARA QoD session is triggered from the second UE with a request for a higher QoS profile (i.e., increase the bandwidth allocation), from 10 Mbps to 20 Mbps. Since the slice has sufficient available capacity at this point, the request is approved without requiring any reallocation or impact on the other flow. However, at approximately 38 seconds, the second UE attempts a further bandwidth increase to 20 Mbps while the first UE is still active. This request would cause the combined throughput to exceed the maximum capacity of the slice, resulting in a potential SLA violation. To address this, our CAMARA solution activates its balancing mechanism, which solves an underlying optimization problem to enforce service priorities while maintaining compliance. However, at approximately 38 seconds, the second UE attempts a further bandwidth increase to 20 Mbps while the first UE is still active. This request would cause the combined throughput to exceed the maximum capacity of the slice,

resulting in a potential SLA violation. To address this, our CAMARA solution activates its balancing mechanism, which solves an underlying optimization problem to enforce service priorities while maintaining compliance. As a result of the optimization, the higher-priority UE is granted its full 20 Mbps as requested. The lower-priority UE has its bandwidth reduced from 20 Mbps to approximately 11 Mbps, which still satisfies its defined SLO. This experiment demonstrates the system's ability to dynamically adapt bandwidth in response to application-level CAMARA QoD API triggers. Enforce priority-aware resource allocation using an intelligent orchestrator. Prevent slice capacity violations through runtime optimization and reconfiguration.

4.4 Summary

This chapter presented the application and AI enablers defined in AMAZING-6G and under implementation in Task 3.3: (1) AI-aaS framework, with three sub-enablers: AI-aaS, ML models catalogue, and MLOps; (2) Digital Twin framework; and (3) OpenAPIs for network exposure, with focus on TM Forum APIs and CAMARA APIs for QoD, energy management, and edge/cloud management.

Each enabler was introduced with a brief description of its functionalities and the explanation of its adoption in AMAZING-6G use cases. A generalized design was presented as guideline for further development and customization, followed by the detailed design of the per-use-case specialized solutions which are currently under implementation. Where available, preliminary results have been described.

5 IoT and Localization enablers

This section provides an overview of IoT and Localization enablers to support critical capabilities through seamless integration of sensing, monitoring and remote operation. These enablers are fundamental for managing the complex interplay between IoT devices, edge-cloud resources, contextual services and localization. The enablers were grouped into different categories, such as:

- Advanced Localization and Positioning systems: where the ability to accurately determine the position and movement of devices, robots and assets are critical. The localization enablers include: (1) hybrid positioning systems, that leverage the integration of 5G networks and GNSS to provide enhanced positioning accuracy, (2) Perception and V2X Data Fusion, where the localization is further strengthened through the fusion of LIDAR data with V2X communications, enabling vehicles and infrastructure to collaboratively understand their environment with special precision and temporal resolution, (3) GNSS and RTK positioning, critical for applications like autonomous driving and drone navigation for example.
- Connectivity and Sensing Infrastructure: reliable and scalable connectivity is essential for IoT deployment. This enabler involves the components and devices used for control and sensing. For example, 5G RedCap IoT sub-enabler offers a cost-effective and energy efficient solution to connect a huge number of sensors and devices.
- Data collection and telemetry: which involves efficient data acquisition and telemetry for monitoring, analytics and control. Involves seamless collection and routing of telemetry data from distributed IoT nodes to edge or cloud, enabling timely insights and responses. Involves mechanisms to store time series data and visualization as well.
- IoT Service Platforms and Management: this covers the IoT Sensors Orchestration capable for dynamically discovering IoT devices, managing their lifecycle, and exposing their capabilities. Also, it brings mechanisms for automated resources orchestration, policy enforcement, and service assurance across IoT -edge-Cloud. These platforms allow us to expose those devices and services to be used by applications to control and analyze it simplifying integration and interoperability.
- IoT Contextual awareness systems: enable us to include context fusion that fuses vehicles, infrastructure and environmental data to derive contextual information. Also includes intelligent systems that analyze contextual data to enhance safety such as road safety, risk assessments and predictive maintenance.
- Remote control and Operation systems: enabling real-time interaction and control over IoT systems. It covers teleoperation and actuation where platforms that support real-time command and control of remote devices, sensors and robots.

In the sections that follow, each enabler will be described in greater details, highlighting their technological foundations, implementation strategies and association and impact across use cases.

5.1 Advanced Localization and Positioning Systems

Accurate localization is a cornerstone for many 6G use cases, such as autonomous mobility, industrial IoT, and infrastructure monitoring. Since no single technology can meet all accuracy and reliability requirements, this project develops advanced hybrid solutions that combine 5G, GNSS/RTK, inertial sensors, and cooperative perception. This section presents the enablers, Hybrid Localization (5G + GNSS), Cooperative Positioning and Perception, and GNSS + RTK Positioning, describing their design, use case associations, and implementation plans.

5.1.1 Description of the enabler

Hybrid Localization (5G + GNSS)

This enabler combines 5G with GNSS (Global Navigation Satellite System) and positioning techniques to improve accuracy and reduce latency in real-time location-based applications. Enhances real-time positioning accuracy by integrating 5G and GNSS data using Kalman filters and machine learning algorithms. Figure 5- shows the Hybrid localization architecture.

The 5G Core Network, represented by light blue blocks, is on the left side. The most important function for localization in 5G is the LMF (Location Management Function), which provides location-based services by retrieving data from 5G network signals. The Positioning Framework is at the center, depicted as a green block, responsible for collecting location data from multiple sources, including the 5G network, GNSS satellites, and Inertial sensors.

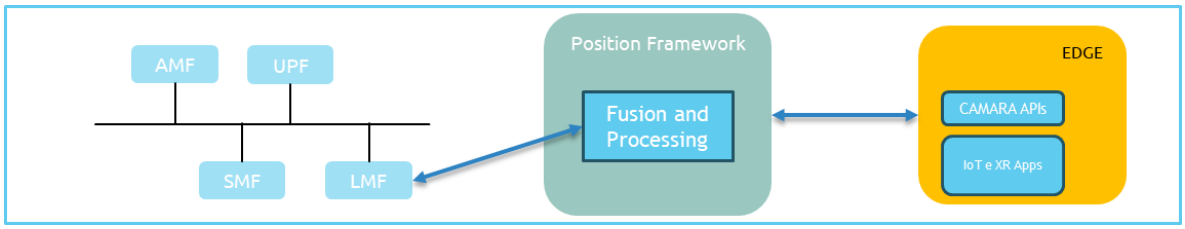

Figure 5-1 Hybrid localization architecture

The fusion and processing mechanism applies Kalman filters and machine learning algorithms to refine the data. The fusion process enhances positioning accuracy by correcting errors, compensating for signal disruptions, and predicting real-time motion paths. The edge computing layer, represented by a yellow block, is on the right side, where the CAMARA APIs expose the positioning data to applications and services. These applications utilize this data, enabling them to make the needed decisions. Examples include autonomous driving, industrial IoT, and urban security. The need for this approach is to cover the limitations of each separate technology to obtain better results. While 5G provides fast response times, its accuracy is lower. GNSS offers high accuracy but suffers from latency and signal blockages in urban canyons. Inertial sensors, such as accelerometers and gyroscopes, maintain movement continuity but experience drifts over time. By combining these three technologies, this framework can deliver real-time localization solutions. Table 5-1 shows the required resources for Hybrid 5G + GNSS Localization.

Table 5-1 Required Resources and Justifications for Hybrid 5G + GNSS Localization

Resources	Characteristics	Justification
Radio Site	Supports basic 5G FR1 bands (sub-6 GHz)	Needed for GNSS assistance via 5G connectivity.
Core Network Services	5G SA (Standalone) with LMF support, enabled for a A-GNSS assistance	GNSS via LMF reduces GNSS time-to-first fix (TTFF) and improves weak-signal performance
UEs & CPEs	GNSS-enabled devices supporting A-GNSS assistance	Ensures better satellite signal acquisition and reliability
GPS Receiver & Antenna	High-precision GNSS receiver with A-GNSS Support	Provides faster GNSS fixes and more table accuracy

Cooperative Positioning using Radio and Perception information

Perception information consists of all the data provided by visual and other perception sensors. These sensors include, but are not limited to, cameras, LiDAR, and radar. Depending on the sensor type,

perception information can be visual, depth-based, or both. For example, depth cameras (such as stereo systems) can jointly provide appearance and range information. Other sensors, like infrared cameras, extend operating conditions in specific circumstances, such as night-time scenarios.

Visual information is used to detect objects in the environment, such as vehicles, moped, pedestrians, animals, whose position needs to be determined. Knowing the installation parameters of the camera (e.g., orientation, height), it is possible to perform the georeferentiation of the identified objects based on geometrical computations. However, this information might be highly unprecise. For this reason, visual data is usually coupled with depth information retrieved from specific sensors, such as LiDAR and RADAR, to have a better accuracy of the position.

Positioning using perception information is limited to the field of view of the sensors. For this reason, a multi-sensors configuration is typically used to have perception information from different points of view achieving a cooperative positioning. For example, in an intersection sensors can be positioned at the four corners to have a complete view of the intersection and avoid also possible occlusions that may occur due to the presence of tall vehicles, such as trucks and buses. The information from all sensors is then fused together in a central processing point to enable high position accuracy.

Challenging aspects of this positioning approach are related to the sensors' calibration and to the time synchronization. It is indeed essential to have precise intrinsic calibration of each single sensor, and it is as well important to have an extrinsic calibration among the coupled sensors (e.g., a camera and a LiDAR installed at the same location with the same orientation) to correctly associate depth information to the visual data. In the scenario of a multi-sensor positioning system, it is also important to have sensors' data that are correctly timestamped following an accurate synchronization otherwise it is likely to have mismatching when data are fused.

Another challenging aspect is the time needed to estimate the positions of the objects in the scene. Having an accurate position but being delayed in time may reduce the usefulness of the information in some scenarios. To reduce the delay, edge computing solutions are typically used. An additional aspect to be considered concerns privacy. It is compulsory that the retrieval of sensors' data and the processing of this data respect the GDPR rules.

Additionally, to perception information, that is typically limited to the areas covered by the sensors, information from the analysis of radio signals can be exploited for locating users. This can complement information from sensors and achieve higher accuracy. It is possible to estimate the users' position by looking at the signal strength received or by using methods based on flight time, such as Time of Arrival, Time Difference of Arrival or Round-Trip Time approaches. 5G radio signals are considered as well as other candidate technologies such as V2X short range radio signals.

GNSS and RTK Positioning

Satellite positioning in the form of Global Navigation Satellite System (GNSS) is a proven positioning technology. In a GNSS-based system, the position is computed based on the distance between the receiver and GNSS satellites. This distance is calculated by multiplying the speed of light by the time it takes for a satellite signal to reach the receiver. While the calculation itself is straightforward, the signal propagation time is affected by several error sources. These include small biases satellite orbits and clock errors, as well as ionospheric and tropospheric effects. Collectively, all these error sources contribute to inaccurate positioning in a GNSS-based system. Standard GNSS positioning provides accuracy in the meter level, which is insufficient for applications that require more precise positioning.

Real-Time Kinematic (RTK) is a technique developed to counteract these GNSS signal errors. The basic principle behind RTK is that it leverages the carrier-phase differential technique to compensate for common errors from the satellites and atmosphere using the correction data. RTK uses a nearby reference station with known coordinates or a network of reference stations (also known as Network RTK) to provide correction data in real-time. In its simplest form, an RTK solution makes use of a single reference station near the GNSS receiver. Since the reference station's position is precisely known, it

can estimate the errors affecting each received GNSS signal. If the distance between the GNSS receiver and the reference station is reasonably short (less than 25 km) such that they experience the same atmospheric conditions, a single reference station is usually sufficient. Network RTK goes a step further by leveraging multiple reference stations instead of a single base station. This approach enables more precise error modelling and significantly improves positioning accuracy over larger areas. By combining GNSS with RTK correction services, positioning accuracy can be enhanced to the centimeter level, particularly in open or semi-open environments.

Another challenge impeding the high accuracy in a GNSS-based system is signal reflection. In urban environments, signal reflections are frequently encountered due to the presence of structures like buildings and other objects. These reflections pose challenges such as multipath and non-line-of-sight (NLOS) issues. These issues greatly deteriorate the accuracy of GNSS positioning systems, resulting in potential errors that can exceed 50 meters in challenging environments. The detrimental effects of multipath can be mitigated by using a multi-band GNSS system. A multi-band GNSS utilizes multiple frequency bands, typically combining generic L1 (such as GPS L1 and Galileo E1) and modernized L5 signals (such as GPS L5 and Galileo E5a). The rationale behind this approach lies in the distinct characteristics of different frequency bands. The L1 band is susceptible to multipath interference, whereas the L5 band exhibits superior multipath mitigation capabilities. By integrating both signals, the multi-band GNSS system can effectively discriminate between direct and reflected signals, enhancing the accuracy of positioning results.

In this enabler, the high precision positioning will be realized by utilizing multi-band and multi-constellation (such as GPS, Galileo and BeiDou) GNSS receiver in combination with RTK technology. The high-level design is shown in Figure 5-2.

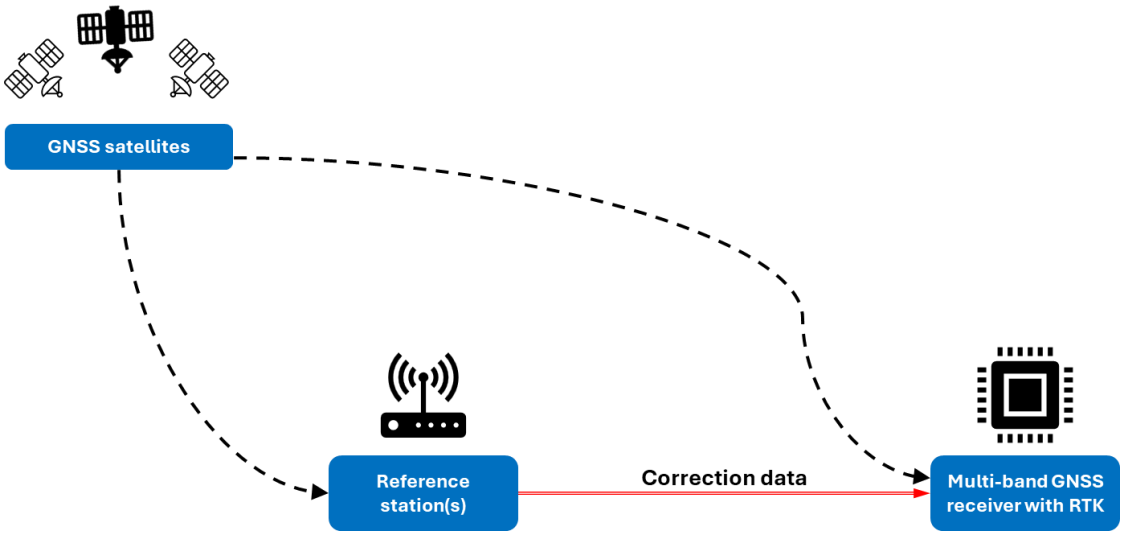

Figure 5-2 High level design of GNSS and RTK positioning

5.1.2 Use case association and contributing partners

The table that follows presents the mapping of the localization and positioning sub-enablers to the corresponding use cases in which they will be integrated, tested, and validated. In the following section, we provide further details on this integration, highlighting the specific context and requirements of each use case as they relate to these enablers.

Table 5-2 Mapping between Localization and Positioning Enablers and AMAZING-6G UCs

	Cooperative Positioning using	Hybrid Localization (5G + GNSS)	GNSS + RTK Positioning

Radio and Perception information			
E2		CAPG, TNO	TNO
T1	LINKS		LINKS
T2	LINKS		LINKS
T5		ThPa	

In the **E2 use case**, a drone equipped with B5G connectivity and an ultrasonic sensor is deployed for the inspection of wind turbine blades. The drone is designed to land on the blade surface and conduct ultrasonic measurements. Therefore, the accurate localization of the drone on the inspected blade is critical. This position information can be used to pinpoint the location of suspected defects. If further analysis of the measurement data indicates the need for additional investigation, the drone must be able to return to the exact location on the blade. This precise positioning of the drone can be achieved by using the combination of GNSS and RTK technology as described above. As the project progresses, additional 5G-based localization techniques developed by consortium partners may be incorporated to enhance positioning accuracy and operational flexibility.

In the **T1 use case**, a Road-Side Unit (RSU) is installed at an urban road intersection. The RSU is equipped with a camera and a LiDAR that monitor the crosswalk and the road segment that leads to the crosswalk. The data from the sensors on the RSU are used to estimate the positions of pedestrians and vehicles. In the case that the road actor is connected, the road shares the GNSS + RTK information to complement the positioning information retrieved from the sensors. This information is made available using standards APIs following ETSI MEC format and a CAMARA approach. Moreover, data fusion with information coming from the mmWave cell and from the radio of the RSU (V2I short-range) will be tested to understand how this approach can help in a precise localization of mobile users on the road (pedestrians and vehicles). The radio data will be collected on the edge server from the RSU and from the mmWave cell and will be fused on the edge server together with the localization information coming from the RSU sensors (when available).

In the **T2 use case**, the robot is equipped with a GNSS receiver enabled to receive RTK corrections for improving the accuracy of the precision. Furthermore, the robot is equipped with camera and LiDAR sensors to perceive the surrounding environment. Sensors' information is exploited to further improve the local positioning for navigation actions. This data will be collected on the edge of the network and made available as defined by ETSI MEC location API following a Camara approach.

In the **T5 use case**, a vehicle is equipped with a LiDAR sensor to capture a 360-degree view of its surrounding environment. The LiDAR generates a high-resolution 3D point cloud at a frequency of 15 Hz, enabling the system to continuously perceive and map everything within the area of operation. This data provides critical spatial awareness for navigation, obstacle detection, and environmental analysis. To complement the LiDAR, a Real-Time Kinematic (RTK) positioning system can be integrated to deliver precise geolocation data, achieving sub-centimeter accuracy in outdoor settings when needed. The combination of LiDAR's rich spatial data and RTK's highly accurate positioning ensures reliable situational awareness, making the system well-suited for advanced (semi-)autonomous applications.

5.1.3 Design, development and implementation

Figure 5-3 presents the design of the Hybrid Localization enabler, which integrates various data sources into a framework. At the Sensor Layer, GNSS satellites, RTK reference networks, inertial sensors (IMU/INS), LiDAR, and user equipment (UEs) deliver raw positioning and environmental data. The

Network Layer, where the network gathers UE observations and GNSS/RTK data, enabling network-assisted positioning. The aggregated information is processed within the Fusion & Processing Framework of the Edge Layer. Utilizing advanced algorithms enables accurate error correction, compensation for signal blockages, and confident prediction of motion trajectories. The positioning results are made available through the Exposure Layer via different APIs, ensuring that external services and applications can easily access the data. The Application Layer confidently uses positioning to drive a range of project use cases (E2, T1, T2, T5), including autonomous drones, vehicles utilizing LiDAR-based perception, and robust industrial IoT solutions. In addition to the primary data flow, the architecture integrates a powerful feedback loop that enables applications to communicate back to the fusion framework, ensuring precise control and configuration whenever necessary.

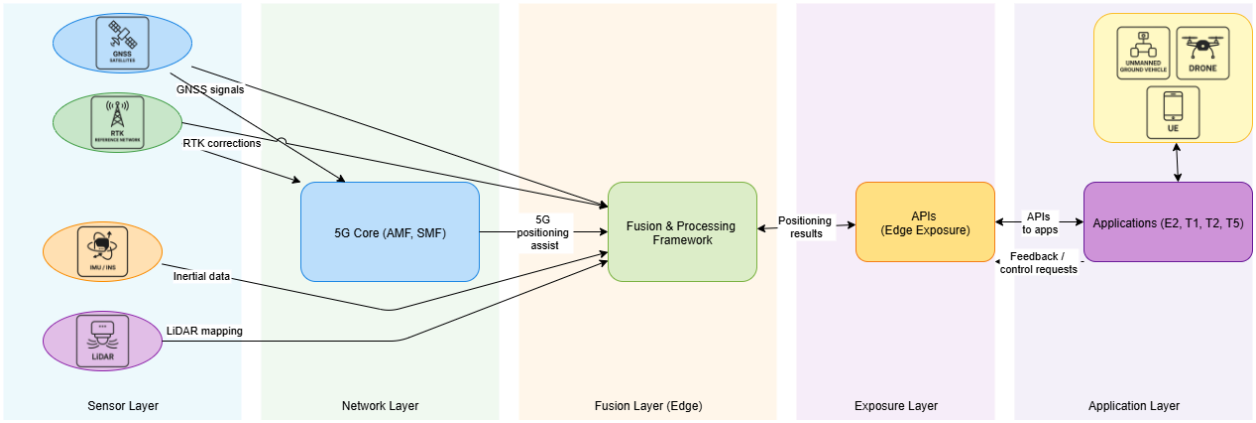

Figure 5-3 Generalized architecture for Hybrid Localization

For the **E2 use case**, precise drone positioning is realized through state-of-the-art **GNSS receivers with RTK capability**. Specifically, the implementation employs two u-blox EVK-X20P modules: one functions as a base station transmitting RTK correction data, while the other operates as a rover mounted on the drone. The EVK-X20P supports multi-band GNSS reception and integrates built-in RTK technology, enabling positioning accuracy at the centimeter level. Configuration of the modules, i.e. designating them as either base station or rover, is facilitated via a computer running the u-center application. This application also manages the transmission of correction data from the base station to the rover module. The accuracy of the GNSS and RTK-based positioning system will be evaluated through a series of field measurements tailored to the requirements of the E2 use case. These measurements will be used to assess the effectiveness and reliability of the enabler in real-world conditions. Based on the outcomes of this evaluation, the suitability of the system for the intended application will be determined. No additional development activities are planned beyond this assessment phase. The technical design for the GNSS and RTK positioning is depicted in Figure 5-4.

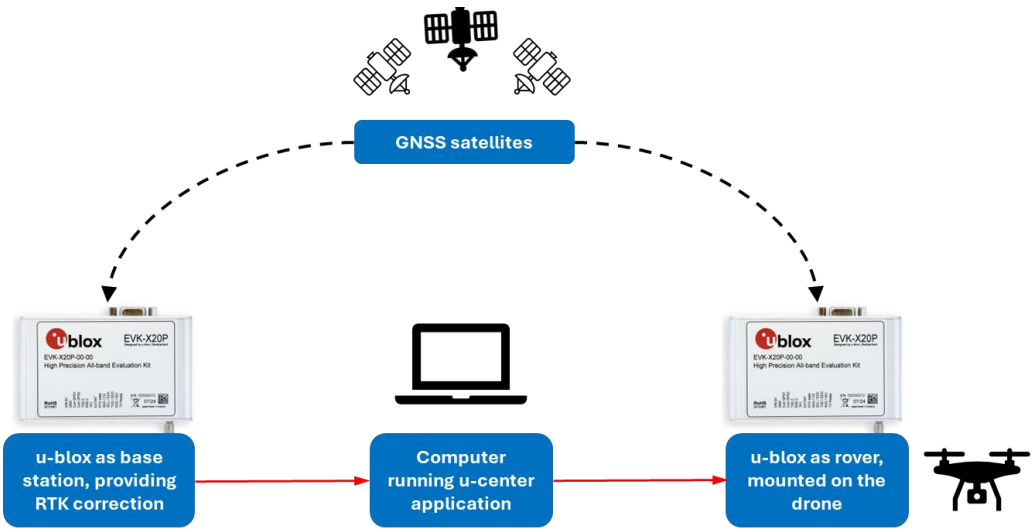

Figure 5-4 Technical design for GNSS and RTK positioning for E2 use case

For the **T1** and **T2 use cases**, **Cooperative Positioning** will use GNSS positioning with RTK, implemented using commercial GNSS receivers that present RTK capabilities. In the specific, uBlox F9R receiver will be installed on the robot. This data will be made available on the network side through standard APIs.

The approach to fuse together information coming from different radios and from visual sensors will be tested when all the cells are deployed in the testbed and when all the data collection mechanisms are implemented and tested. Figure 5-5 shows the architecture for multi-source positioning for T1 and T2.

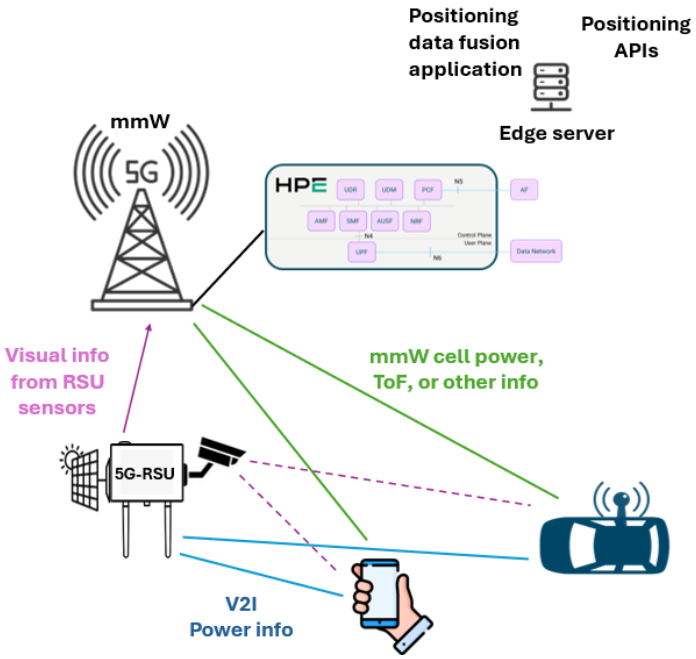

Figure 5-5 Architecture for multi-source positioning for T1 and T2

The Hybrid Localization system in the **T5 use case**, operates using an advanced algorithm that combines LiDAR-Inertial Odometry with Simultaneous Localization and Mapping techniques. This software fuses the incoming positioning data from the sensors, processes it in real time, and constructs a highly detailed and accurate 3D representation of the surrounding environment. Unlike static pre-generated maps, these 3D models can be created dynamically on demand, ensuring that the output reflects the most up-to-date conditions of the operational area. Each map is georeferenced, meaning that all features are placed and oriented correctly within a global coordinate system, allowing for seamless integration with other spatial datasets. Once generated, the 3D maps can be stored locally for immediate use or uploaded to the cloud, where they can be accessed remotely, shared across platforms, or further analyzed for applications such as route planning, infrastructure inspection, or long-term environmental monitoring. Figure 5-6 presents the positioning aspects for T5 use case.

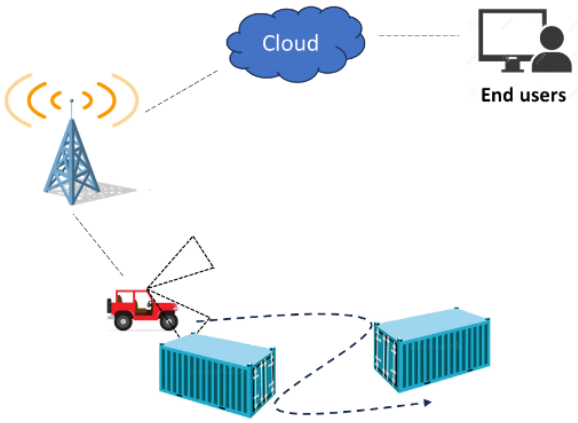

Figure 5-6 Positioning aspects for the T5 use case

5.2 IoT Connectivity and Infrastructure

This enabler covers the blocks that ensure seamless interconnection between IoT devices, gateways, and communication infrastructures. It focuses on integrating heterogeneous devices and protocols into a unified ecosystem by leveraging 5G (including 5G RedCap) and LPWAN technologies, enabling scalable, energy-efficient, and reliable data exchange. The enabler addresses both the hardware components (sensors, gateways, devices) and the communication technologies that allow them to interoperate, aggregate data, and connect with edge or cloud platforms for further processing.

5.2.1 Description of the enabler

IoT Gateways

In the context of AMAZING-6G, an IoT gateway acts as a bridge between Internet of Things (IoT) devices and the cloud or centralized data systems, enabling smooth communication, data processing, and device management. Since many IoT devices use low-power, short-range protocols like Zigbee, Bluetooth, MODBUS, etc, they often can't directly connect to the internet. The IoT gateway translates these diverse communication protocols into standard internet protocols such as TCP/IP, allowing data to be sent to cloud platforms for storage, analytics, or further processing. This protocol conversion ensures interoperability among devices from different manufacturers and helps build scalable IoT ecosystems. Also, through an IoT gateway, traffic by multiple devices, cameras, sensors can be aggregated and transmitted to the cloud for further processing and analysis.

Beyond simple connectivity, IoT gateways incorporate some local processing and filtering capabilities, also known as edge computing. By initial processing data locally before sending it to the cloud, they reduce latency, save bandwidth, and enhance reliability in situations with intermittent internet connectivity. This combination of connectivity, processing, and security functions makes IoT gateways a critical component in applications such as health, PPDR, energy and transport.

5G RedCap IoT

5G NR RedCap (Reduced Capability) was introduced in 3GPP Release 17 as a scaled-down version of 5G NR. 5G RedCap features reduced energy usage, bill-of-materials, and form factors related to full-blown 5G NR, while offering lower peak data rates. 5G RedCap is aimed at IoT devices and of particular interest for wearable devices, as their small form factor also implies a limited battery capacity (i.e., energy usage is a very important KPI). Note that the advent of reduced capability modems for IoT in 5G resembles the introduction of LTE-M and NB-IoT in the 4G era. However, data rates (and likely energy usage) of 5G RedCap are significantly higher than those of LTE-M/NB-IoT, making it a particularly interesting technology for wearables which generate a significant amount of (uplink) data traffic. Uplink data rates for RedCap can be as high as 120 Mbps (FDD) or 45 Mbps (TDD). Another potentially interesting technology is eRedCap which was introduced in 3GPP Release 18 and features (uplink and downlink)

data rates of 10 Mbps [11]. Figure 5-7 illustrates how different cellular IoT technologies relate to each other and to 'legacy' cellular technologies.

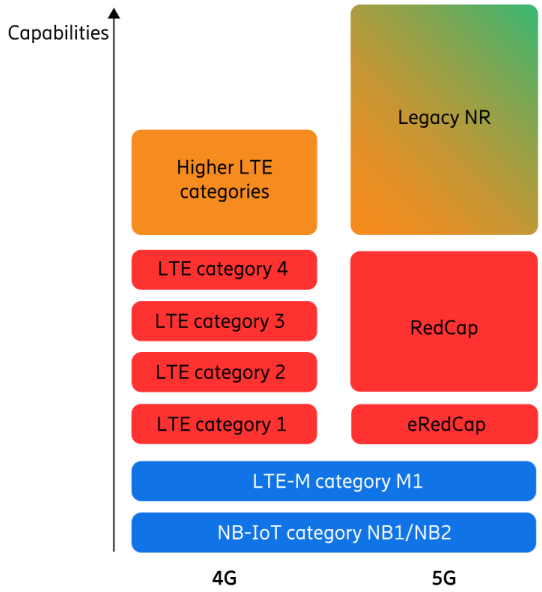

Figure 5-7 Overview of 4G/5G IoT solutions vs. legacy 4G/5G

IoT Sensors and Devices

IoT sensors and devices are the hardware that monitor the environment and convert measurements into digital information. These include: i) environmental probes to measure physical properties such as temperature, humidity, and air quality; ii) motion sensors to measure acceleration and rotation; iii) infrastructure sensors to measure pressure, vibration, and power. Advanced sensors including cameras, LiDAR, and RADAR can also be included when the goal is to perceive the overall environment rather than only specific aspects. When the trials are conducted all the devices used (including UEs, RSU, 5G access points etc.) will be explicitly mentioned (not only IoT sensors / devices).

The information gathered by sensors is sent to back-end services using various communication technologies. Common solutions include Wi-Fi or Ethernet in buildings, or BLE/Zigbee when short-range and low-data-rate links are sufficient. For outdoor use, low power wide area networks (e.g., LoRaWAN) or cellular connectivity can be used. The choice of communication solution mainly depends on the required data rate. This may range from continuous streaming (e.g., cameras, LiDAR, and other perception sensors) to a few messages per day for in-field sensors. It also depends on the needs of the back-end applications that use the data collected from IoT sensors and devices. A challenge, especially for in-field devices, is power supply. In some cases, IoT sensors and devices must rely on battery power, and this power source may need to last for long periods (e.g., a few years) because replacing sensors is expensive. Another challenge is interoperability among IoT sensors and devices from different vendors. Proprietary data formats are still common, despite ongoing community efforts to promote interoperability.

In the context of AMAZING-6G, when paired with environmental sensor nodes, an IoT gateway serves as the central hub for collecting and managing diverse environmental data streams. These sensor nodes, which measure parameters such as temperature, humidity, and air quality, utilize low-power communication protocols to extend battery life and operate in remote or distributed locations. The gateway aggregates this data, translates it into a unified format, and securely transmits it to the cloud platform. This allows environmental monitoring systems to function cohesively, even when the sensor nodes themselves are constrained by limited processing power or communication range. The device and platform are developed to monitor various parameters of the environment, e.g., by using IoT sensors for

temperature, humidity, microparticles, smoke, etc., and transmit data over available networks for further analysis and identification of potential security incidents (e.g., fires, etc.).

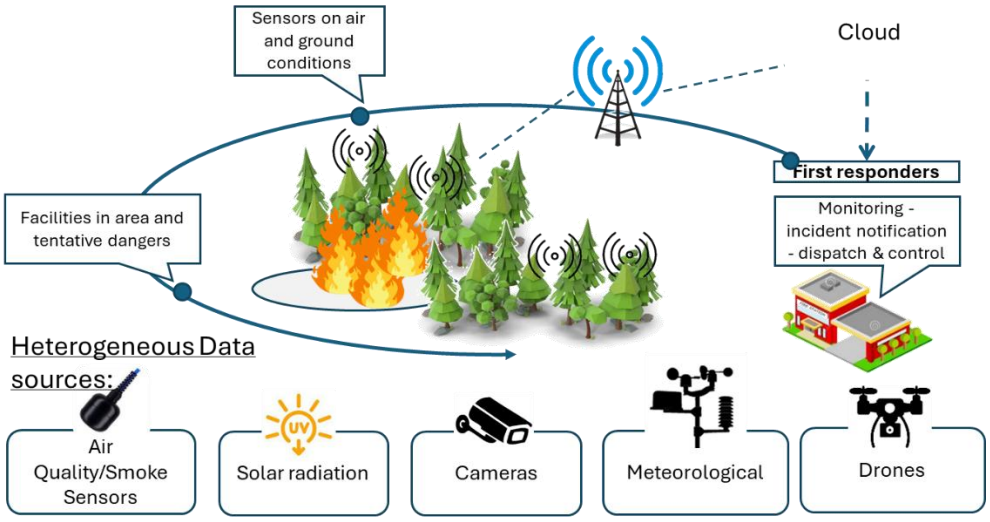

Figure 5-8 Overview of environmental IoT sensor nodes

Similarly, IoT gateways can be paired with solar inverters, battery inverters, and environmental sensors to collect and manage operational and environmental data streams. This integration enables both energy management and monitoring of external conditions within a unified framework.

5.2.2 Use case association and contributing partners

The table below illustrates how the IoT Connectivity and Infrastructure sub-enablers are associated with the respective use cases where they will be deployed, tested, and validated. The following section elaborates on this integration, emphasizing the specific context and requirements of each use case in relation to these sub-enablers.

Table 5-3 Mapping between IoT Connectivity and Infrastructure Enablers and AMAZING-6G UCs

	IoT Gateways	5G Redcap IoT	IoT Sensors and Devices
H1		TNO	
H2		TNO	
P1	WINGS		WINGS
T1	LINKS		LINKS
T2	LINKS		
E3	SIM		SIM
T5	WINGS, ThPA		WINGS, ThPA

Both **use cases H1 and H2** propose vital-signs patches which are attached to a patient's chest by means of an adhesive. This means that form factor and weight are severely constrained. Clinical case H1A and use case H2 require the patch to be battery-operated and therefore the research on these topics will mainly focus on assessing and minimizing energy consumption. The 5G RedCap IoT enabler will play a key role in this.

Clinical case H1B requires high-bandwidth, continuous uplink traffic with sub-second round trip delays. Initial uplink estimates are very high (86 Mbps) but it is the object of the work in WP4 to reduce this bandwidth considerably. Nevertheless, this clinical case will test the limits of the enabler in terms of uplink bandwidth.

Use case H2 requires much more moderate uplink bandwidth (2.4 kbps) but with an estimated 10 ms latency, and as such also challenges the limits of the enabler in terms of latency, especially in combination with the strict energy requirements mentioned above.

Environment parameters monitoring in the context of **use case P1** is essential. In the context of P1, a device and platform are developed which monitors various parameters of the environment e.g.by using IoT sensors for temperature, humidity, microparticles, smoke etc. and transmits data over available networks for further analysis and identification of potential security incidents (e.g. fires etc.).

Both **T1 and T2** have IoT sensors like the camera and LiDAR sensors that are installed on the RSU in T1 and on the robot in T2. The data from these sensors are used in both cases for the identification of events and for contributing to the positioning estimation.

Use case E3 focuses on real-time monitoring and control of photovoltaic installations. SIMTEL develops a custom IoT gateway that collects telemetry from inverters and environmental sensors, executes local control logic, and synchronizes with a central cloud platform. SIMTEL also provides the physical devices involved in the setup, including solar inverters and auxiliary sensors such as irradiance and temperature, enabling comprehensive energy forecasting and optimization.

The seamless interconnection between IoT devices, gateways, and communication infrastructures is crucial for **T5 use case**, since there is a number of sensors, STS subsystems, cameras, etc. that will be used for real-time monitoring and control of the STS crane.

5.2.3 Design, development and implementation

The overall design of the “IoT Connectivity and Infrastructure” enabler is depicted in Figure 5-9. It illustrates different connectivity options for IoT devices. Generally speaking, IoT devices may be energy- and/or cost-constrained and consequently integrating a full-blown 5G-NR radio in them might not be realistic. Basically, two solution directions exist to address these issues.

One solution direction is to use different connectivity means, either wired or through short range radio technologies, to connect the individual IoT devices to a gateway and subsequently connecting this gateway to the 5GA network: Edge Gateway (SIMTEL/CSOFT, E3), IoT Gateway (WINGS/ThPA, P1/T5), and Road Side Unit (RSU) (LINKS, T1/T2). This is particularly beneficial if several IoT devices are co-located and fixed relative to each other.

Another solution direction is to directly connect the IoT device to the 5GA network, using reduced capability modem technologies such as 5G-RedCap or 5G-eRedCap: medical wearables (TNO/OUS, H1/H2). This so-called “Direct-to-Cloud” connectivity is an improvement of state-of-the-art devices which typically connect via Bluetooth through a phone, as the error-prone process of Bluetooth pairing and the need to bring a phone along (and keep it charged) is avoided. In other words, it enables hassle-free and reliably connectivity for every patient, which is particularly relevant in life threatening situations. Observe that 5G-RedCap connectivity may also provide cost-effective connectivity to gateway devices: Edge Gateway (SIMTEL/CSOFT, E3).

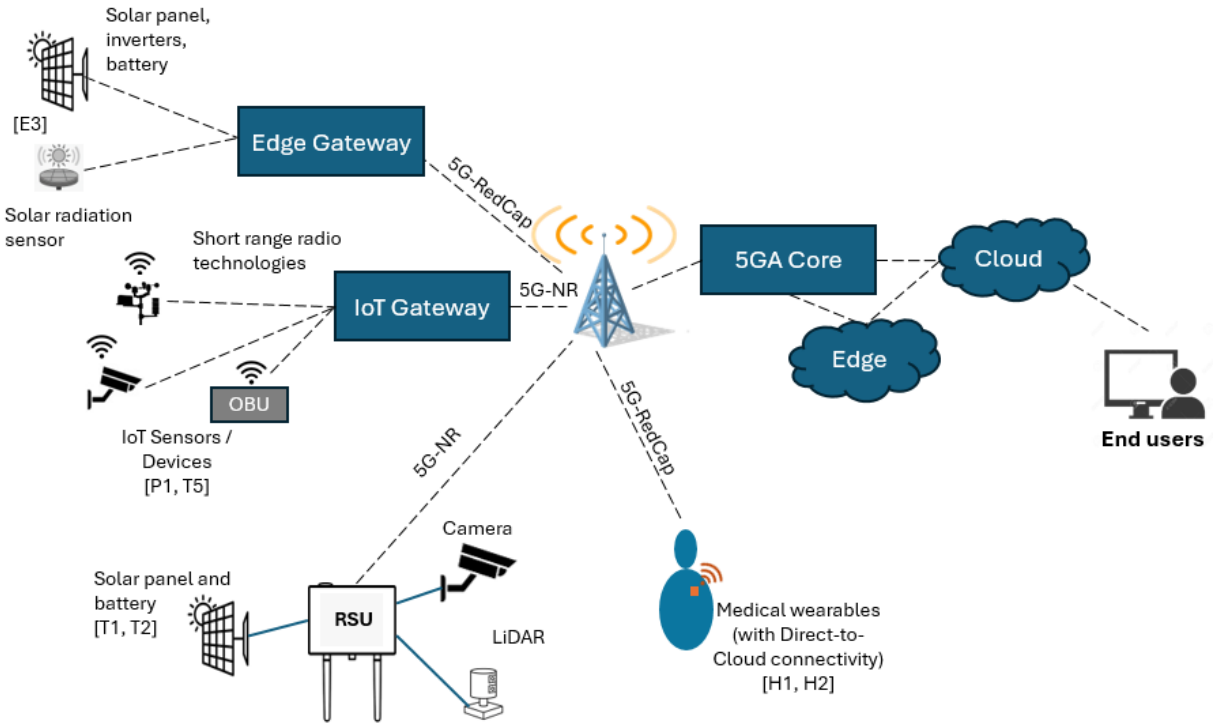

Figure 5-9 Overall design for the "IoT Connectivity and Infrastructure" enabler

For the **H1** and **H2** use cases, **5G RedCap IoT** is used, and several 5G RedCap (R17) modems have already become available on the market. TNO has obtained a Quectel RG255C with an evaluation board and managed to get it operational. Note that 5G eRedCap modems may also be of significant interest to the project; however, the first modems supporting 5G eRedCap are only expected on the market in 2027, making their timely availability within the project questionable. The focus of the work with regard to this sub-enabler will be on energy consumption, but also on coverage, as there exists a strong correlation between signal strength/quality on one hand and energy consumption on the other. Therefore, a test setup featuring detailed energy measurements, as shown in Figure 5-10 is being realized. The Qoitech Otii provides power to the evaluation board (Figure 5-11 middle) and measures the power consumption at a 1-4 kHz sampling frequency. PC-based tooling (CLI tool) is available to conduct automated, repeated tests (for statistical significance) and to extract and display the power consumption profiles like the one shown in Figure 5-11 (right). The system is built into a Faraday cage and the antenna signal can be attenuated to simulate poorer coverage conditions. The network side is represented by a 5GA network in a box; the Amarisoft 5G box. Different (types) of modems can be evaluated this way. Initial experiments comparing a full-blown 5G-NR modem (Fibocom FG150) with the Quectel RG255C shows a factor two reduction in energy consumption which is consistent with [12].

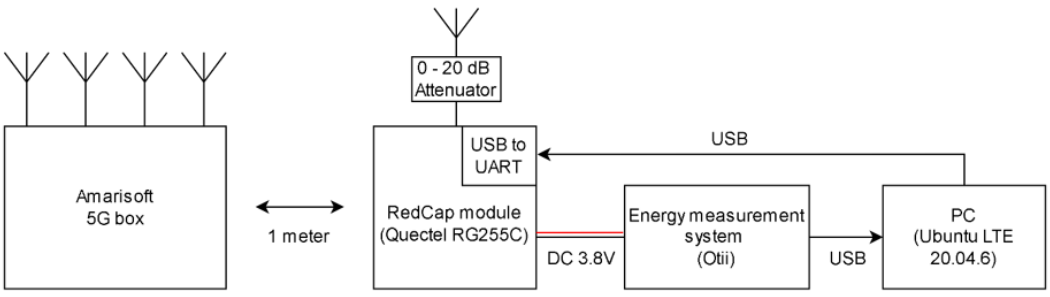

Figure 5-10 TNO 5G RedCap test setup

The test setup is portable, enabling offsite demonstrations and experiments. Figure Figure 5-11 (left) shows a portable casing comprising the Quectel evaluation kit, Otii, and attenuator in a Faraday cage. Next to the case is the Ubuntu PC. The Amarisoft 5G box is not shown but portable as well.

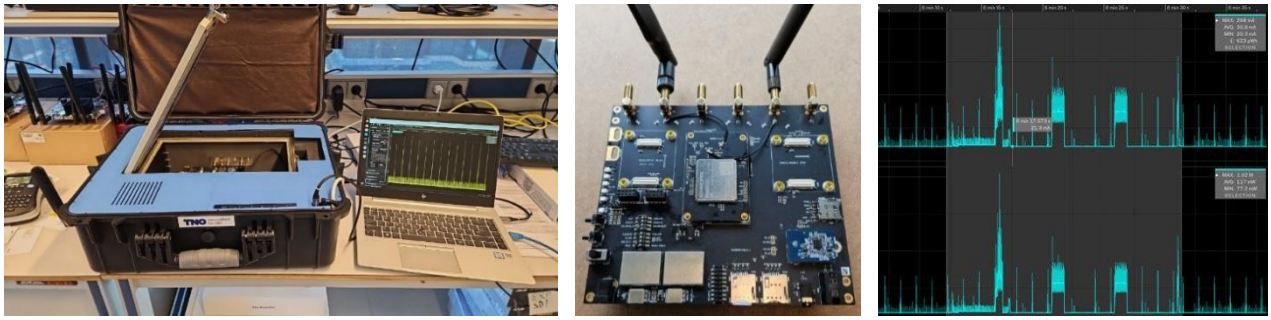

Figure 5-11 Portable TNO test setup (left), Quectel evaluation board (middle), and example power trace (right)

An inventory of alternative modems (i.e., next to Quectel RG255C) available on the market will be made, focusing on features (relative to the needs of the use cases), maturity, and energy consumption. Based on that inventory (one or two) additional modems may be purchased, brought up-and-running, and integrated into the test setup. The latter involves e.g. adapting scripting to the particulars of the AT-command set of said modems. Subsequently, the 5G RedCap performance evaluation experiments conducted within the scope of the project (i.e., WP4/WP7) may be conducted with those modems as well. In addition, the availability of 5G eRedCap modems on the market will be monitored and – if possible, within the timeframe of the project – one of those modems could be purchased, brought up-and-running, and integrated into the test setup as well to enable further experiments.

Initial experiments compared a legacy 5G-NR modem (Fibocom FG150) with the RedCap-based Quectel RG255C. The results show a factor two reduction in energy consumption, with the RedCap modem consistently consuming about half the energy of a traditional 5G modem when transmitting small, frequent packets (128 bytes at 300 ms intervals). These findings are consistent with Jörke et al. [12] and highlight the technology's potential for battery-powered IoT devices. Importantly, the results also show that most energy is consumed during active data transmission rather than idle periods, suggesting that optimizing transmission intervals can further reduce consumption.

Looking ahead, the testbed will be extended to include:

- Add capability to measure the energy consumption of HTTP and HTTPS requests.
- Perform bandwidth analysis to measure the maximal up and downlink speed.
- Add support for legacy 5G NR modems such that they can be compared with the RedCap modems.
- Add support for LTE-M and/or NB-IoT modems such that they can be compared with the RedCap modems.
- Add support for eDRX and PSM to further reduce the power consumption of non-continuous transmitting and receiving devices.

In the **P1** and **T5** use cases, an **IoT gateway** acts as a bridge between Internet of Things devices and external networks, enabling seamless communication, data processing, and management across diverse systems. Since IoT devices often operate with different communication protocols (e.g. Wi-Fi, Bluetooth etc.), the gateway serves as an aggregator of data from different sensors and translator, converting this device-specific data into a common format (e.g. json structured data) that can be transmitted over standard IP-based networks. By doing so, it ensures interoperability among heterogeneous devices and allows data to flow reliably from sensors and machines to cloud platforms, edge servers, or enterprise applications. This functionality makes the IoT gateway a critical component in large-scale IoT deployments where thousands of devices may need to communicate efficiently without overwhelming the core network. Figure 5-12 shows the IoT gateway setup.

To support this functionality, the first version of the IoT gateway device has already been developed by WINGS. During the 1st year, initial testing will be carried out in a controlled laboratory environment over

a private network. This will be followed in the 2nd year by field deployments to enable preliminary experimentation, while large-scale final trials are planned for the 3rd year.

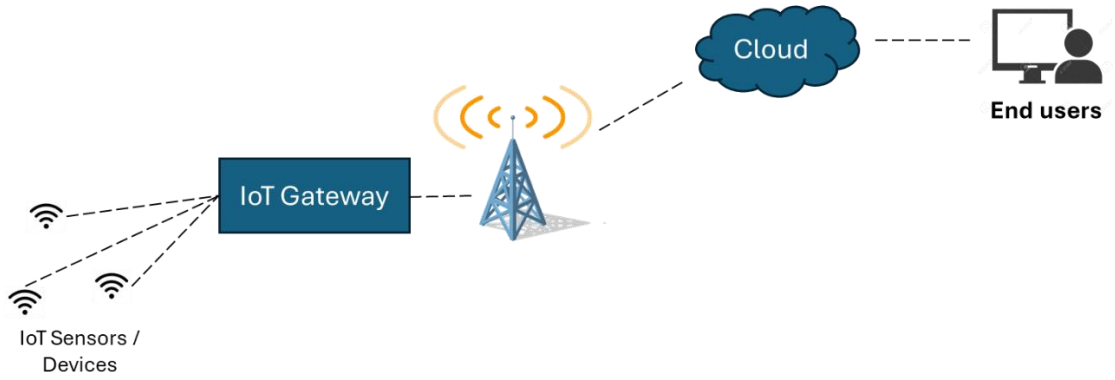

Figure 5-12 IoT Gateway setup

For the **T1** and **T2** use cases, IoT sensors and devices based on COTS are expected to be deployed. The development is applicable to how the data from the sensors are collected, processed, and used by application at the network side. The enabler will exploit 5G connectivity for sharing sensor data using well-known application layer protocols (e.g., such as RTSP for video streams). In both use cases, the IoT sensors used are camera and LiDAR sensors. Data from these sensors are sent to the edge using 5G connectivity. LINKS is actively contributing by implementing the integration of camera and LiDAR data streams into the 5G testbed, optimizing data pipelines for low-latency transmission, and validating real-time processing at the edge to support the use case requirements.

For the **E3** use case, SIMTEL, together with its affiliated entity CSOFT, will lead the design, development, and implementation of the “Solar energy monitoring, control, and prediction using B5G/6G communications and edge-cloud” scenario within the Energy domain (WP5 – E3).The use case showcases a smart gateway system that automatically monitors solar power plants with inverters, collecting real-time energy production data and transmitting it to the cloud. This enables energy companies to predict tomorrow's power generation by combining current performance data with weather forecasts, optimizing energy trading and grid management. The system provides complete visibility into solar power plants operations, supporting both day-to-day monitoring and long-term energy planning decisions.

At the core of the solution is a B5G-enabled industrial edge computer designed to interface directly with solar energy plants. The architecture integrates hardware and software components to provide a robust, secure, and intelligent end-to-end application for the green energy vertical. The phased approach foresees iterative development and validation of these enablers within the Romanian pilot.

The overall architecture is depicted in Figure 5-13: The central hardware component will be an industrial computer developed by SIMTEL. This device will feature an integrated 5G RedCap (3GPP Release 17) modem with full support for network slicing, enabling differentiated QoS for various data flows. The software foundation for this device will be provided by CSOFT, building upon their existing platform for integrating SCADA and non-SCADA communications on customer premises. The device will perform several key functions on-site:

- Data Acquisition & Processing: It will read and process real-time metrics (e.g., currents, powers) directly from solar panel systems and large-scale energy storage units.
- Local Intelligence: It will handle local alerting based on preset thresholds and execute pre-programmed logic, allowing for resilient operation even if the backhaul link is lost.
- Remote Actuation: It will serve as the gateway for remote commands, a critical feature for complying with national energy authority regulations that require energy production adjustments in under 5 minutes.

For E3, the IoT Connectivity and Infrastructure enabler supports the reliable data exchange and orchestration between the solar field devices, industrial edge gateways, and the edge-cloud platform. It ensures seamless connectivity for IoT endpoints (e.g., inverters, sensors, and controllers) distributed across large solar installations. This enabler includes the deployment of B5G/6G-compatible IoT gateways equipped with 5G RedCap (Release 17) modules and the configuration of dedicated network slices (URLLC and eMBB) over ORO's 5G/B5G infrastructure to guarantee deterministic latency, reliability, and QoS differentiation. At infrastructure level, the enabler integrates secure MQTT/TLS communication protocols, SDN-based traffic management, and edge-cloud orchestration using containerized services. This setup allows local actuation even during temporary network disruptions, enabling autonomous site operation while maintaining compliance with national grid regulations. The IoT Connectivity and Infrastructure enabler is mainly applied to E3 for real-time control, predictive analytics, and compliance monitoring, providing the data backbone that enables the forecasting engine, dashboards, and APIs to operate reliably. The implementation of the E3 use case follows a three-year phased approach ensuring a structured transition from design to full validation.

In Year 1, activities focus on the definition of the use case architecture, communication flows, and edge–cloud interfaces, as well as on ensuring interoperability with inverter systems and compliance with regulatory requirements. The procurement of industrial edge devices, sensors, and 5G/B5G connectivity equipment is planned, and laboratory environments will be prepared for integration testing. Year 2 is dedicated to the preliminary integration of the IoT Connectivity and Infrastructure enabler, linking 5G RedCap edge devices with ORO's edge-cloud infrastructure. Secure MQTT/TLS data channels and orchestration functions will be deployed and validated in laboratory conditions, while forecasting engines, dashboards, and operator interfaces will be progressively integrated. In Year 3, full-scale field trials will be carried out under real operational conditions to validate performance against key KPIs. The final phase includes consolidated analysis of trial data and evaluation of the enablers' performance and scalability, with emphasis on the IoT Connectivity and Infrastructure enabler.

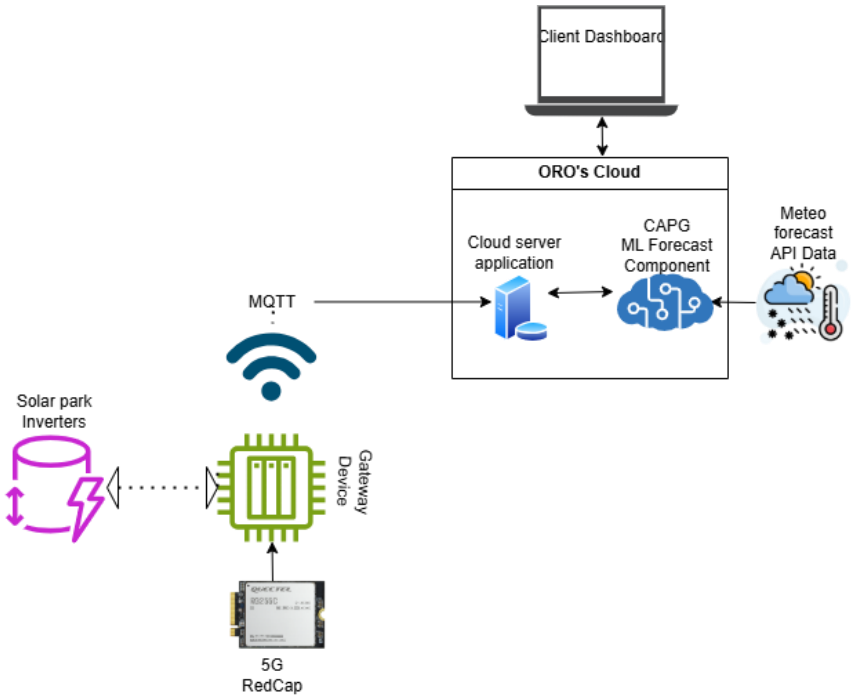

Figure 5-13 E3 System Context Diagram

The development and implementation will follow a phased approach, aligned with the timeline of WP5 tasks (T5.1, T5.2, T5.3).

- Phase 1: Hardware Prototyping and Software Integration (Months 3-12)
- Phase 2: Edge Software Finalization and Network Onboarding (Months 9-18)

- Phase 3: Edge-Cloud Platform and API Deployment for ML Processing (Months 13-24)
- Phase 4: End-to-End Integration and Testing (Months 13-33)

5.3 Data Ingestion and Telemetry

The Data ingestion and Telemetry enablers are responsible for real-time observation, analysis and interpretation of data and events occurring with the IoT environment. Provides the layer for ensuring observability, enabling decision-making, fault detection and situational response across diverse domains, such as mobility, infrastructure or eHealth.

This enabler focuses on the collection, transmission, storage, and processing of data generated within IoT environments. It addresses the telemetry chain: from sensor measurements and streaming protocols to scalable storage systems and advanced data processing pipelines. It also includes monitoring platforms that unify heterogeneous data sources, provide real-time visualization, and detect anomalies to maintain operational awareness. By ensuring continuous observability and timely interpretation of events, Data Ingestion and Telemetry enables effective decision-making, early fault detection, and responsive control in different domains.

5.3.1 Description of the enabler

Data Collection from sensors and devices

In AMAZING-6G, sensor data collection in IoT systems begins with sensor data streaming, where measurements from devices are transmitted to gateways, servers, or cloud platforms in real time or near-real time. This streaming is facilitated by lightweight communication protocols tailored for IoT environments. For example, MQTT is widely adopted for its lightweight publish/subscribe model and low bandwidth requirements, making it well suited for resource-constrained devices. CoAP operates over UDP, offering low-latency communication for resource-limited sensors. SNMP is often used for managing and monitoring networked devices, while HTTPS through REST APIs provides a straightforward, secure, web-based data transfer for scenarios requiring robust encryption and compatibility with standard web and cloud infrastructure, enabling the transfer of periodic telemetry data. These protocols ensure efficient, reliability, and secure delivery of sensor data from the edge to data processing systems.

Once transmitted, the sensor data must be stored for both immediate use and long-term analysis. Data storage can occur at multiple layers e.g. in on-premises relational or NoSQL databases for privacy-sensitive applications, or in cloud-based storage systems for scalability and accessibility. Modern storage solutions often involve time-series databases such as InfluxDB or TimescaleDB, which are optimized for handling continuous streams of timestamped sensor data.

The final stage is data processing, where raw sensor readings are transformed into actionable insights. Processing can happen at the edge (close to the data source) for low-latency decision-making, in the cloud for large-scale analytics, or in a hybrid model combining both. Data may be cleaned to remove noise or errors, aggregated to reduce volume, and enriched with contextual information (e.g., location, weather conditions). Advanced processing might involve applying statistical models, machine learning algorithms, or event-driven rules to detect anomalies, predict trends, or trigger automated responses. This processing layer turns raw data into meaningful, timely intelligence that can drive operational efficiency, improve decision-making, and enable proactive interventions.

Monitoring Platform

In AMAZING-6G, a monitoring platform in the context of IoT and sensor networks serves as the central interface for observing, analyzing, and managing system performance and environmental conditions. The platform aggregates incoming data from multiple sources, such as IoT gateways, servers, and cloud services, and present it in a unified, organized manner. By continuously tracking metrics such as device health, network performance, and environmental readings, a monitoring platform ensures that operators

have a real-time view of their systems. This visibility is essential for early detection of faults, performance degradation, or anomalies in sensor data, enabling timely maintenance and intervention.

In the project, a proprietary platform for the use cases of P1 and T5 is being developed by WINGS as part of its wi.BREATHE and wi.MOVE products. It is cloud-based, and it is possible to access the platform remotely through any electronic device such as PC / tablet / smartphone. Access to the platform is allowed only to authorized users and tiered access of users with specific roles (administrators or ordinary users) using passwords is supported. The connection to the platform is secured (SSL encryption) and the platform presents in a friendly way the overall state of the smart gateways connected to the sensors. The platform allows grouping of devices into groups and presents on a map the locations of the devices installed in the field. Graphs of the measurements are presented for a period of time selected by the user or in real time. The platform presents alerts for each sensor and sensor group. In case abnormal situations are observed, the software sends notifications (via email or SMS) to selected users and administrators. All measurements are visualized through user-friendly graphs. Also, the platform analyzes historical data of the measurements and sends alerts in case of an unforeseen change in the pattern of the measurements. The sensitivity of these alerts is set on the platform via a configurable differential rate. The platform supports the analysis and comparison of measurements.

Grafana complements the platform by providing a rich visualization layer that transforms collected metrics into interactive dashboards and graphs. It allows users to create customizable visualizations that can combine data from various sources or cloud-based monitoring services. In an IoT scenario, Grafana dashboards display real-time environmental readings, historical trends, or device uptime statistics, enabling quick identification of patterns or issues.

In E1 the Monitoring Platform described in section 3.1.3 is adopted. The choice to use a single Monitoring Platform for the collection of several types of data from different sources (network, computing resources, and IoT platforms) allows to implement unified data models across IoT-device-edge-cloud continuum and achieve a better interoperability among different domains. This facilitates the joint processing and correlation of heterogeneous metrics for more powerful elaborations, at the vertical application level (in this case, for the SB-EMS) and for management and orchestration purposes. In particular, at the resource orchestrator IoT elements are managed in combination with the associated devices and extreme edge nodes, taking into account the logical and physical relationships among them.

5.3.2 Use case association and contributing partners

Table 5-4 Mapping between Data Collection and Monitoring Platform Enablers and UCs

	Data Collection	Monitoring Platform
P1		WINGS
E1	NXW	NXW
E3	SIM	
T3	TUC	

An elaborate platform for checking status of devices, sending notifications and monitoring the overall trial will be provided in the context of the **P1 use case**. The features of the platform are analyzed in the subsection above.

In the **E1 use case** the Data Collection system and the Monitoring Platform are used to collect, unify and aggregate data coming from different comfort and energy-related data sources in the smart building and REC environment, e.g., energy meters, IoT platforms and environmental sensors (e.g., for temperature, humidity, etc.), smart appliances, controllers for energy production measurements, etc.

In the **E3 use case**, SIMTEL develops a dedicated data acquisition and control device deployed on-site in a solar power plant. The device collects telemetry from photovoltaic inverters and auxiliary environmental sensors (e.g., temperature, irradiance), buffers data locally, and forwards it to the cloud platform. It also receives remote control commands and applies them to the inverters, supporting a full feedback loop.

The railway sector is moving toward B5G/6G-based signaling systems that must be resilient, safe, and efficient, and the **T3 use case** want to showcase that. Existing copper-based hardwired infrastructure is costly, vulnerable to vandalism, and difficult to repair after disasters. At the same time, GSM-R is reaching its end of life. LCX cables—already widely deployed in tunnels and stations—can serve as dual-purpose enablers for communication and sensing (ISAC) in the FRMCS era. This directly addresses the need for continuous positioning, collision avoidance, and obstacle detection, all while ensuring reliable signaling traffic transmission.

5.3.3 Design, development and implementation

The use cases E1, E3, T3, P1 rely on the data collection, ingestion and telemetry capabilities provided by the **Data Collection and Monitoring systems**.

The foundation of data ingestion and telemetry lies on the Physical Layer, where field assets such as sensors, controllers, cameras etc. continuously generate real-time operational data. This data can be integrated directly through smart or through existing systems (e.g. SCADA, ERPs). This layer forms the physical-digital interface, enabling a direct link between the working / physical environment devices and the digital platform. Figure 5-14 shows Data Collection and Monitoring Platform.

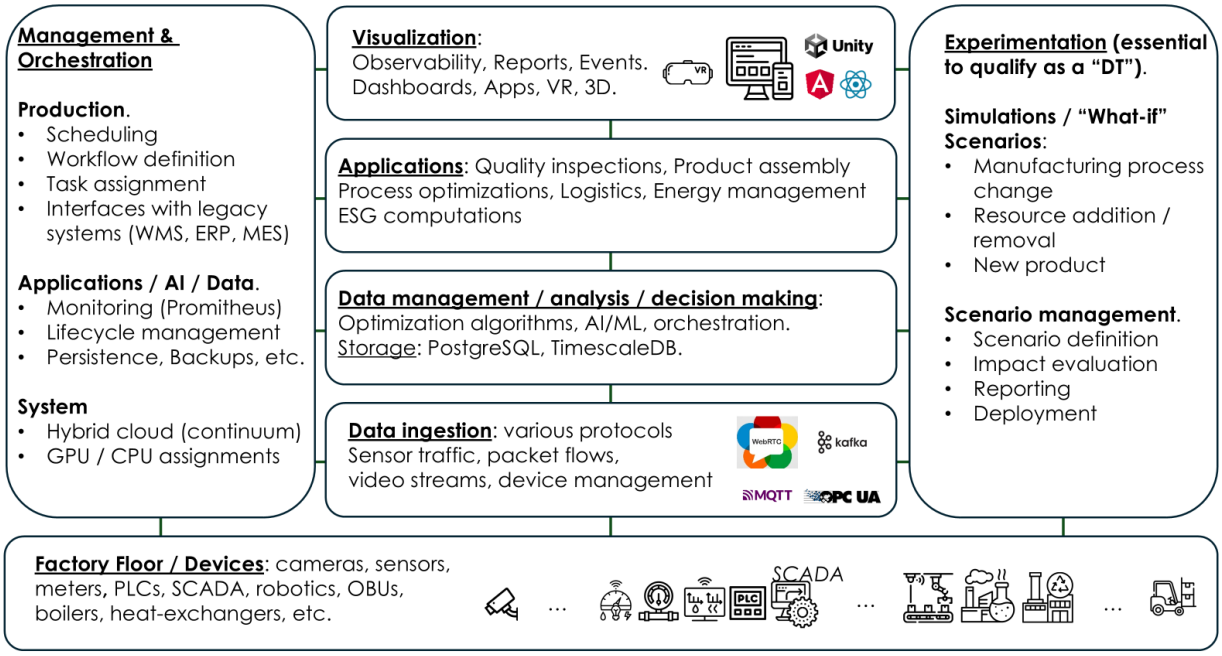

Figure 5-14 Data Collection and Monitoring Platform

Moving upwards, the Data Ingestion Layer establishes reliable two-way channels for both collecting data and pushing commands from/to field assets, enabling real-time monitoring and control. Whether assets on the factory floor, a warehouse or on a fleet of vehicles this layer leverages a variety of communication protocols such as HTTP, MQTT, OPC UA, and WebRTC to handle high-throughput streams of structured and unstructured data, ensuring data is transmitted efficiently and reliably for upstream processing.

On top of this, the Data Management/Analysis/Decision-Making Layer transforms data into actionable insights and efficient decisions. Powered by scalable databases like TimescaleDB, and enhanced with AI, ML, and various optimization algorithms, this layer enables real-time analytics, historical data mining,

reasoning and automated decision-making. It serves as the brain of the platform, orchestrating intelligent operations across all domains.

The Applications Layer builds on this intelligence to serve different business needs, ranging from quality inspections and process optimization to logistics coordination and energy management. By tailoring analytics and automation to distinct operational areas, this layer connects technology directly with business value, driving operational efficiency and sustainability.

The top layer of this architecture is the Visualisation Layer, offering insights and empowering users to monitor, diagnose, and interact with the system effectively. Through fully customisable dashboards and reports, served via web or mobile interfaces and enabled via immersive 3D and VR interfaces, this layer transforms complex data into intuitive visual narratives and seamless control functions.

A first version of the WINGS monitoring platform has already been delivered and successfully tested in a controlled laboratory environment over a private network. The next phase will extend these activities to field deployments to enable preliminary experimentation. NXW is progressing with the initial implementation of its platform components, while SIM is advancing with the preparation of the testbed and implementation; results from both will be provided in the upcoming deliverable.

The **LCX-based ISAC** enabler supports the T3 use case implemented by TUC. It uses leaky coaxial cables along railway tracks as distributed antennas to support both 5G FRMCS communication and sensing for worker safety. The LCX radiates 5G n78 (3.7 GHz) signals from an OAI-based RAN and simultaneously captures reflections and channel variations caused by human motion. Temporal dynamics, especially micro-Doppler signatures, are exploited to detect workers on or near the rails.

Development relies on LCX channel models that include leakage, distributed propagation, and human-induced Doppler effects. Two sensing modes are considered: uplink SRS measurements in cooperative mode, and downlink passive sensing with reference–surveillance correlation in non-cooperative mode. In both cases, spectrograms are generated where micro-Doppler sidebands indicate walking and limb motion. Cameras are used only during data collection to provide keypoint-based labels for training deep learning models; in operation the system runs RF-only. The implementation plan follows a practical sequence:

- Model and simulate LCX propagation at 3.7 GHz with attention to leakage and Doppler effects.
- Integrate LCX with an OAI 5G gNB and SDR hardware, enabling both uplink SRS and downlink passive radar modes.
- Deploy a rail-side testbed with LCX, synchronized SDRs, and cameras for labeled data.
- Process CSI and passive IQ data to extract micro-Doppler spectrograms using clutter removal and time–frequency analysis.
- Train deep learning models on RF spectrograms with camera supervision to classify presence, motion type, and worker count.
- Validate performance against latency, detection probability, and robustness requirements.
- Benchmark LCX-based ISAC against conventional antenna deployments in terms of coverage, energy efficiency, and cost.

This approach shows how existing LCX installations can be upgraded to provide both reliable 5G connectivity and continuous safety monitoring along tracks, enabling worker detection, micro-Doppler analysis, and resilient operation in GNSS-limited environments. LCX based ISAC is ongoing implementation, focusing on preparation of the rail-side testbed. No experimental results are available yet, and outcomes will be reported next deliverable.

5.4 IoT Service Platforms and Management

This enabler provides the capabilities required to manage and orchestrate IoT ecosystems at scale, bringing together devices, resources, and services into a platform. It decomposes the functionality of full-stack IoT platforms into modular enablers, such as device orchestration, service and resource lifecycle management, message queue fabrics, and exposure functions. Together, these elements ensure that IoT devices can be securely onboarded, monitored, and controlled; that services can be provisioned and optimized across the cloud–edge continuum; and that data and commands can flow reliably between IoT infrastructures and external applications.

5.4.1 Description of the enabler

IoT Devices Orchestration

The orchestration of IoT devices covers various steps from their initial onboarding in the system, up to their management at runtime, ensuring their integration in the whole system so that they can be properly monitored and configured through the IoT platform.

Depending on the capabilities of IoT devices and platforms, as well as policies and settings of the smart environment, the various steps can be handled in a manual, partial or full automated manner. Some IoT brokers support automatic discovery of devices. In other cases, IoT devices need to be manually registered, e.g., via GUI or REST APIs specifying their metadata, or importing bulks of devices following a script-based approach, or through proprietary device provisioning services where new devices can be self-registered using pre-provisioned security credentials, QR codes, or tokens. Once registered, IoT devices are authenticated and onboarded in the platform, connecting directly via IoT protocols (e.g., MQTT - Message Queuing Telemetry Transport, CoAP – Constrained Application Protocol, or LwM2M – Lightweight Machine-to-Machine) or through the mediation of an IoT gateway. Fine-grained policies can be defined to control which users, tenants or groups can manage each device and how (e.g., access in read mode only, in configuration or management mode).

IoT devices onboarded in IoT platforms are usually continuously monitored verifying their health and connectivity, to determine if a device is active/inactive and connected/disconnected. Sensors output and devices status are reported in the admin and telemetry dashboards and, depending on configurations, alerts and notifications are sent to the administrator in case of anomalous behavior, abnormal values, disconnections, etc.

Service and Resource Lifecycle Manager

Service and resource lifecycle management refers to the orchestration procedures to provision a service over the computing continuum, which can integrate not only edge and cloud resources, but also extreme edge nodes, devices, and IoT platforms. A detailed description of service and resource management mechanisms is provided in section 3.1.

Message queue fabric

A Kafka broker is a server responsible for receiving, storing, and serving streams of data between producers (data sources) and consumers (data processors or applications). It manages topics—logical groupings of message streams -by storing data in partitions and replicating them across multiple brokers for fault tolerance. The broker ensures high-throughput, low-latency delivery by handling message retention, maintaining offsets for consumer groups, and balancing load across partitions. In the context of AMAZING-6G, a Kafka broker acts as a reliable, scalable backbone for streaming massive volumes of real-time data from devices and gateways to analytics systems or monitoring platforms.

IoT Exposure Function

In the context of AMAZING-6G, the IoT exposure functionality serves as a critical integration layer that enables IoT platforms to interact seamlessly with external systems through northbound and southbound

APIs. Northbound APIs are designed to expose IoT data and services to higher-level business applications, dashboards, analytics engines, or third-party systems. They allow these applications to retrieve processed sensor data, query device states, or subscribe to event streams, often using interfaces communication interfaces like REST or WebSockets with Server-Sent events (e.g. REST etc.). This upward-facing communication is essential for enabling use cases such as predictive maintenance dashboards and AI-driven analytics integrations, where IoT-generated insights need to flow into decision-making systems.

Southbound APIs, on the other hand, handle communication between the IoT platform and the underlying devices, gateways, and edge nodes. They allow the platform to send configuration updates, issue control commands, or request specific data from connected devices using protocols like MQTT, CoAP, LwM2M, or proprietary interfaces, while also enabling sensors' data ingestion. This downward communication ensures that the IoT system can not only collect data but also actively manage and orchestrate devices in the field. Together, northbound and southbound APIs create a two-way exchange of information—northbound pushing insights and data upward, and southbound pushing control and configuration downward—enabling a fully interactive, adaptive IoT ecosystem that integrates tightly with both operational and business processes.

5.4.2 Use case association and contributing partners

Table 5-5 Association of IoT Service Platforms and Management enablers with UCs

	IoT Resource Orchestration IoT Sensors Orchestration	IoT Resource Registry	IoT Service Registry	Service and Resource LCM	Message Queue Fabric	IoT Exposure Function
P1	WINGS	WINGS	WINGS	WINGS	WINGS	WINGS
E1	NXW	NXW	NXW	NXW	NXW	NXW
T5						ThPa/CERTH

In the context of the **P1 use case**, enablers on IoT resource management, APIs exposure functionality and messages are essential in order to provide the necessary information to the users which will participate to the trial and provide furthers analytics and results of the project.

IoT platforms are used in the **E1 use case** for the Smart Building Energy Management System (SB-EMS). The IoT platforms allows to interconnect and interact with environmental and presence sensors, smart appliances, energy meters, and actuators for HVAC, lighting and blinds deployed in the smart building. Sensing APIs are used to collect data related to energy consumption, environmental data, presences in the rooms, and configured settings, in order to feed the energy and comfort optimization algorithms. Actuation APIs allows to send commands to control renewable energy sources, HVAC, lighting and blinds settings. They are used to execute automatic or manual commands sent by the users through the system dashboards or the smart building interfaces.

In the context of the **T5 use case**, enablers on IoT resource management, APIs exposure functionality and messages are essential in order to provide the necessary information to the users which will participate to the trial in the Port area and provide furthers analytics and results of the project.

5.4.3 Design, development and implementation

The orchestration enabler provides the foundational capabilities for large-scale coordination of heterogeneous devices, resources, and services within IoT platforms. Its architecture decomposes

monolithic IoT platform functions into modular enablers, including device and asset orchestration, resource and service lifecycle management, distributed messaging fabrics, and northbound/southbound exposure interfaces. These components collectively support secure device onboarding, continuous monitoring, and fine-grained control of IoT assets. In addition, they enable dynamic provisioning, scaling, and optimization of services across the edge–cloud continuum, while ensuring deterministic and reliable bidirectional flows of telemetry, metadata, and control commands between IoT domains and external application ecosystems.

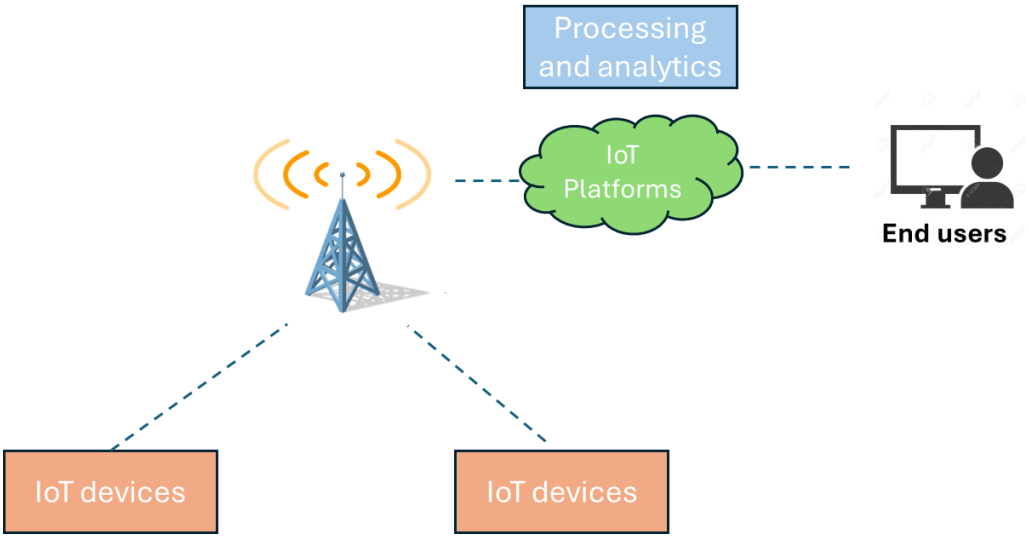

Figure 5-15 IoT Service Platforms and Management

The **IoT Devices Orchestration and Service and Resource Lifecycle Manager** provides the underlying orchestration and lifecycle management functions that enable both **P1** and **E1** use cases. The SB-EMS solution designed for **E1** is represented in Figure 5-16. Data is collected from several IoT sensors, energy meters, and smart appliances at the smart building, through the mediation of an IoT platform. They feed an SB-EMS application based on cognitive closed loops that analyze the metrics to derive predictions on rooms occupancy, comfort preferences, and energy consumption. These predictions are taken as input for decisions that jointly optimize the perceived level of comfort and the power consumption, sending suggestions to the users or directly actuating the decisions through commands issues via the actuation APIs exposed by the IoT platform.

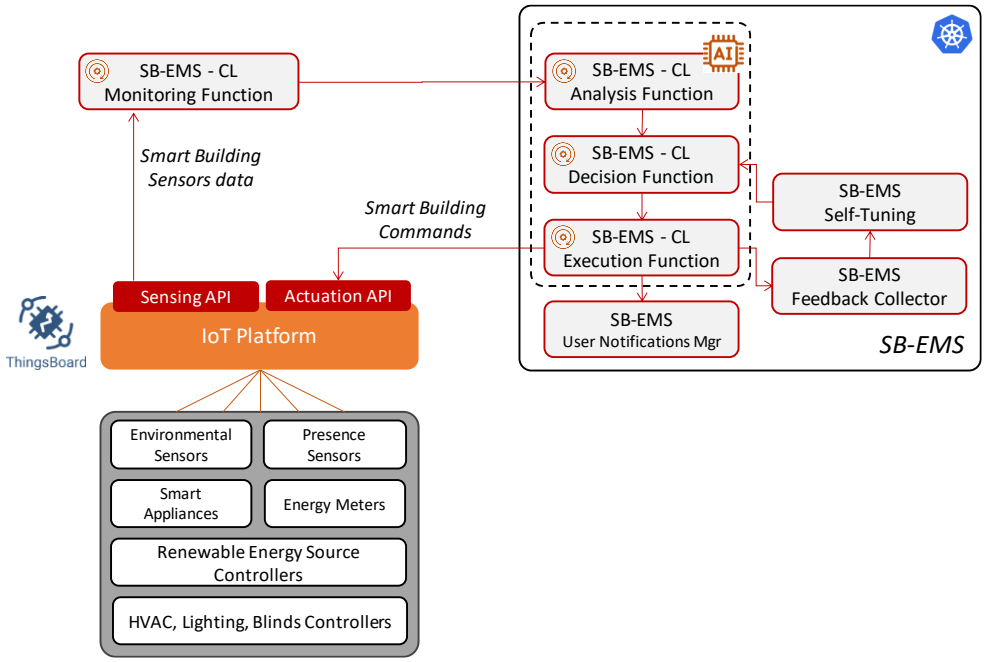

Figure 5-16 SB-EMS solution: high-level design

The IoT platform selected for the implementation is ThingsBoard[9] community edition (released under Apache 2.0 license), an open-source software which already integrates mechanisms for management of multi-protocol IoT devices, data collection, processing, and visualization. Thingsboard can be easily configured to connect to heterogeneous IoT sensors, already supporting the main IoT protocols (MQTT, CoAP, HTTP, LwM2M). Moreover, additional drivers can be created if needed to support other devices implementing proprietary protocols. The architecture is quite modular, making it suitable for integration in edge and cloud environments. Internally, it provides functionalities for managing telemetry and creating personalized dashboards for monitoring and control purposes.

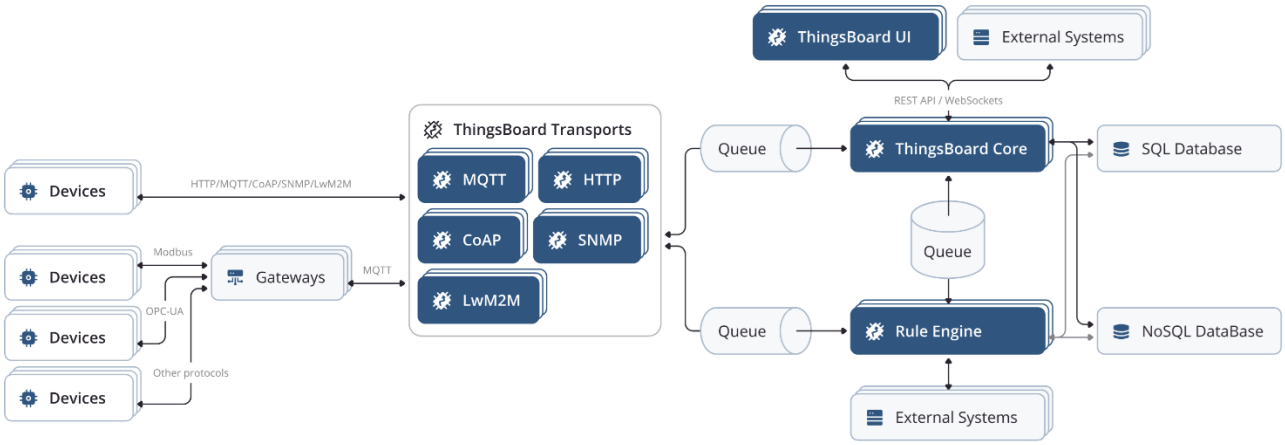

Figure 5-17 ThingsBoard architecture[10]

In E1, ThingsBoard will be used as a central IoT platform within each building, integrating with the sensors (temperature, presence, air quality, lighting) and actuators (HVAC, lighting, blinds) available in the trial site. Implementation of additional drivers is not foreseen at this stage, since the relevant protocols are natively supported. The integration with the rest of the SB-EMS application will use ThingsBoard server-

[9] https://thingsboard.io/

[10] https://thingsboard.io/docs/reference/

side APIs, in detail attributes and timeseries query API for the Sensing API and RPC API for Actuation API. The implementation started with the design and configuration of the IoT architecture, integrating sensors, actuators, and the ThingsBoard platform. It will then progress to deploying the SB-EMS solution at the ORO facility in Q1-2026, enabling cognitive closed-loop analytics and orchestration. Finally, the system will be validated and optimized for enhanced energy efficiency and user comfort.

The **message queue fabric and IoT exposure** functionality will be implemented in the E1 and P1 use cases, as they provide the integration layer that connects IoT devices, platforms, and higher-level applications

The IoT exposure functionality serves as a critical integration layer, represented by Figure 5-18, that enables IoT platforms to interact seamlessly with external systems through northbound and southbound APIs. Northbound APIs are designed to expose IoT data and services to higher-level business applications, dashboards, analytics engines, or third-party systems. Southbound APIs, on the other hand, handle communication between the IoT platform and the underlying devices, gateways, and edge nodes. The figure that follows provides a graphical representation of the physical world, the IoT devices, the ICT/monitoring layer (with the north and south interfaces), the A.I. layer for hosting algorithms and intelligence and the visualization layer for visualizing the content.

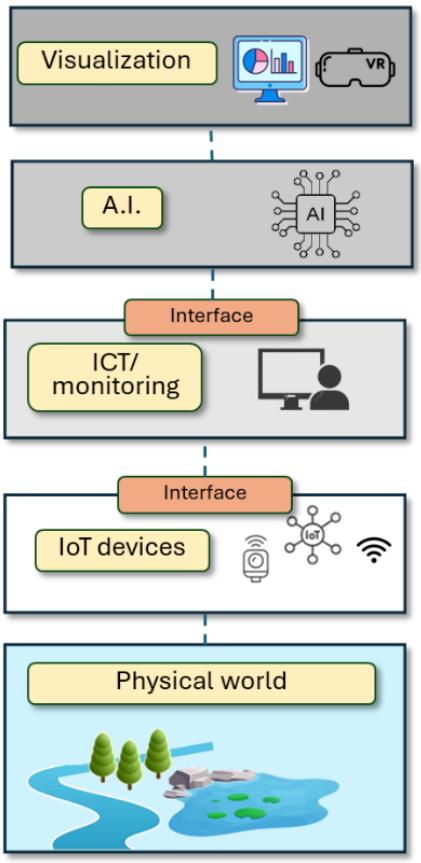

Figure 5-18 Exposure functionality between different layers

Regarding implementation initial testing is ongoing in a controlled lab environment using a private network during the 1st year. At this phase internal development and functionality testing are in progress by the different entities involved in this enablers, and preliminary results are not available yet. Field deployments are planned for the 2nd year to support preliminary experimentation and finals trials are scheduled for 3rd year.

5.5 IoT Contextual Awareness Systems

This enabler enhances raw IoT sensing data with additional contextual information to provide deeper situational understanding. It covers the collection and fusion of heterogeneous data sources, ranging from local sensors (e.g., LiDAR, radar, cameras) to external inputs such as maps, environmental conditions, and mobility datasets. Advanced data processing and AI/ML methods are applied to clean and correlate these inputs, turning low-level measurements into actionable insights. The result is a more complete awareness of ongoing conditions, enabling applications to anticipate risks, optimize system behavior, and support safe and efficient operations.

5.5.1 Description of the enabler

V2X Situational Awareness

V2X situational awareness plays a critical role in extending the system's understanding of the vehicle's environment beyond what on-board sensors alone can provide. Autonomous vehicle, or remote operation rely not only on real-time video feeds but also on fused V2X data to make safe and informed decisions.

- V2V (Vehicle-to-Vehicle) data fusion enables the vehicle to receive notifications about sudden braking or abnormal manoeuvres of nearby vehicles, enhancing anticipation of risks.
- V2I (Vehicle-to-Infrastructure) data such as traffic light status, temporary road closures, or work zone alerts can be integrated into the operator's dashboard to avoid unsafe manoeuvres.
- V2P (Vehicle-to-Pedestrian) information, such as warnings from smart devices or roadside sensors, can help the identification of vulnerable road users hidden from direct camera views.
- V2N (Vehicle-to-Network) communication provides access to aggregated traffic information and hazard notifications from a wider area, supporting strategic decisions.

By fusing these multiple V2X inputs with local sensor data (e.g., radar, lidar, cameras), a richer safety understanding of the vehicle's surroundings is achieved. This allows for the accurate interpretation of complex scenarios, reduction of blind spots, and quick reaction, which is crucial for maintaining safety.

5.5.2 Use case association and contributing partners

V2X situational awareness is essential for the **T4 use case** because it provides the remote operator with a comprehensive and reliable view of the vehicle's surroundings that goes beyond the on-board camera feeds and other sensor data (both raw and processed). By sharing data between vehicles, infrastructure, and sensors, V2X ensures that the operator can anticipate potential hazards. This enriched awareness supports safer and more effective remote decision-making, reducing reaction times and enhancing overall safety.

5.5.3 Design, development and implementation

In the context of **T4**, the TUC V2X testbed will employ a variety of sensing devices, e.g., LiDARs, cameras (both for driver monitoring, and environment perception), GNSS Receivers, Radars, IMU, as well as 5G communication equipment, to achieve V2X situational awareness.

- Radar-Sensor Setup for 360° Environment Recognition
- Radar sensors are used to achieve complete 360° environment perception, see Figure 5-19. These systems use:
- Extended Object Tracking: Objects detected by multiple sensors are tracked using their historical trajectory and visualized as coloured points, see Figure 5-20.
- Duplicate Track Merging: When objects are captured by more than one sensor, duplicates are identified and merged as "extended objects." For example, objects tracked in the neighbouring lane by blue and green tracks can be fused into a single representation.

- Generalized Probability Data Association (GPDA): This statistical method is applied to ensure robust and reliable merging of data from multiple sources.

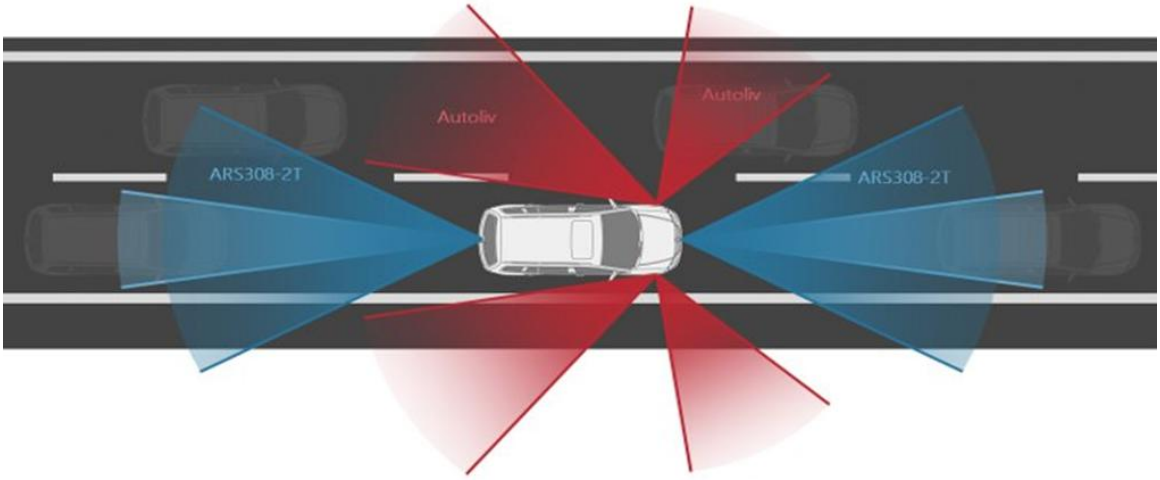

Figure 5-19 360° environment recognition with the use of radars/sensors

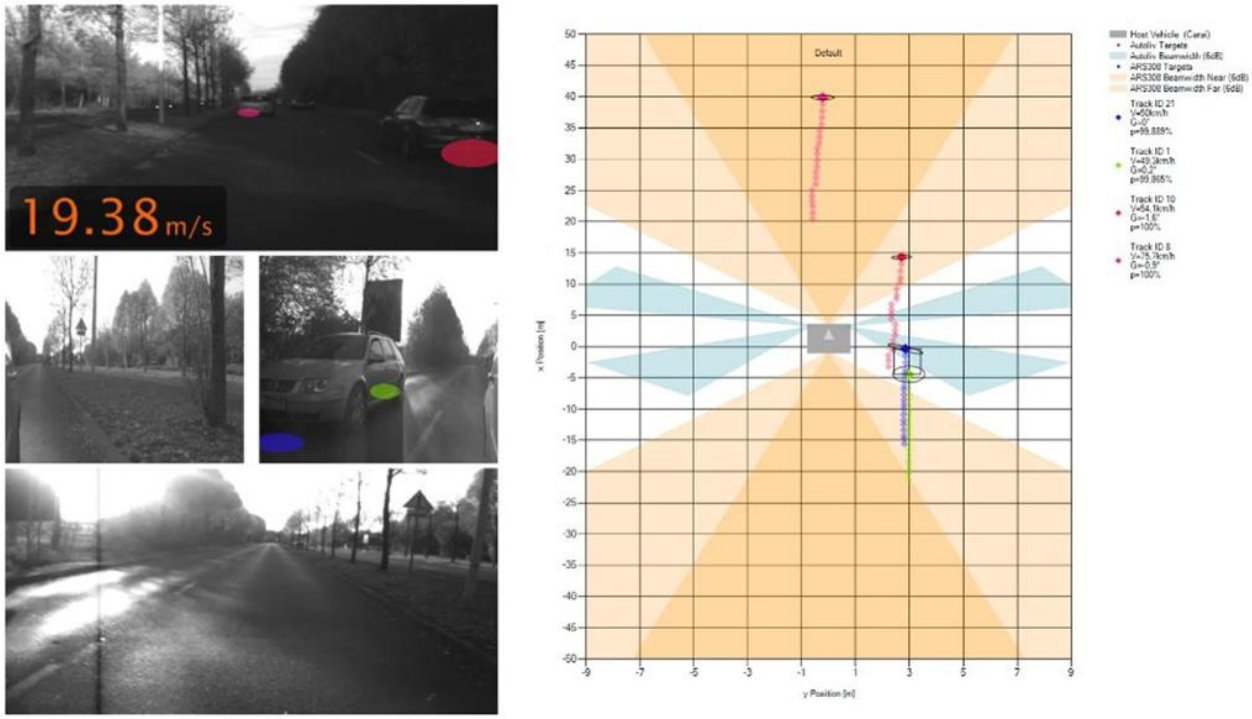

Figure 5-20 Extended object tracking

To improve driver assistance, sensors gather data both inside and outside the vehicle, see Figure 5-21. These inputs are combined to support:

- Driver Intention Estimation: Predicting what the driver plans to do next.
- Behaviour Monitoring: Understanding how the driver interacts with the vehicle and environment.
- Detection of Non-Driving Related Tasks: Identifying when the driver is distracted or engaged in secondary tasks.
- Discomfort Estimation: Detecting possible stress or unease while driving.

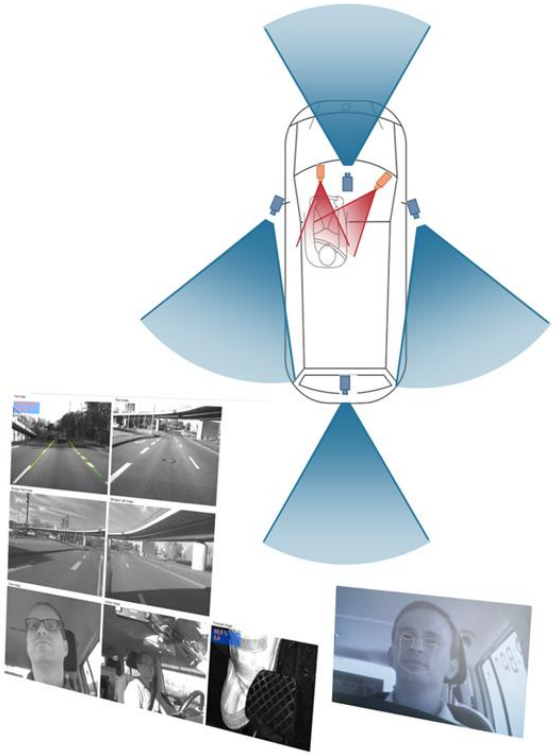

Figure 5-21 Information gathered from inside and outside the vehicle

Estimating Lane-Change Intention

Accurate prediction of a driver's intent to change lanes is critical for safety and cooperative automation, see Figure 5-22. Respective intentions can be predicted 5–3 seconds in advance.

- Predictions are based on a fusion of:
  - Driver information (e.g., head/hand movement)
  - Environmental information (e.g., nearby vehicles, road layout)
  - Vehicle trajectory (e.g., current lane, speed, direction)

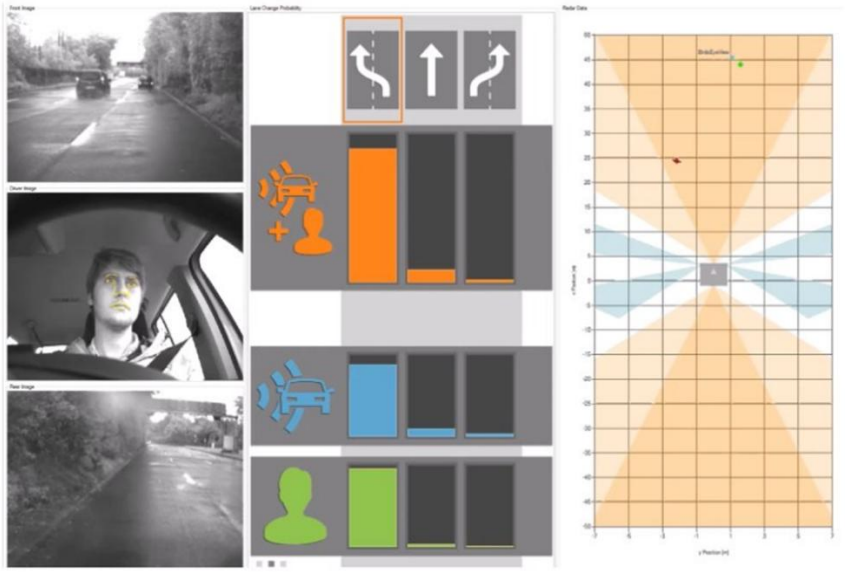

Figure 5-22 Estimation of lane change intention

Bicycle Trajectory Prediction in Cooperative Automated Driving

Ensuring cyclist safety is a key challenge for connected automated vehicles. To this end, the TUC V2X testbed employs:

- Neural Networks for Collision-Free Forecasts: Machine learning models can predict safe paths for cyclists.
- Data Fusion Approach: The model integrates multiple data types:
  - Static Data: Map and infrastructure information.
  - Dynamic Data: Vehicle movement data and cyclist kinematic information.
- Deep Neural Network (DNN): A DNN framework is used to merge static and dynamic inputs, producing reliable bicycle trajectory forecasts for connected and cooperative driving systems.

Implementations of the above described concepts are already in place in the TUC testbed. Any necessary adaptations/amendments in the context of T4 will be performed within the second year of the project and will be used in the final experimentation.

5.6 Remote Control and Operation (Actuation)

This enabler provides the mechanisms to control IoT devices, machines, and infrastructures remotely and in real time. It includes different layers of actuation: from manual control via dashboards and Remote Procedure Calls (RPCs), to automated responses triggered by rule engines based on telemetry conditions. The scope covers both human-in-the-loop teleoperation, where operators directly command devices such as cranes or robots, and automated actuation, where edge/cloud platforms issue commands to field devices with guaranteed low latency and reliability. By integrating QoS mechanisms, network slicing, and AI-driven decision support, the enabler ensures that commands are executed securely, predictably, and within strict time constraints.

5.6.1 Description of the enabler

Real-Time Actuation/Teleoperation

The Real-Time Actuation and Teleoperation enabler provides the communication and control functions required to monitor, manage, and adjust connected infrastructures in near real-time. It is based on an edge-to-cloud architecture interconnected through B5G/6G network infrastructures. Industrial-grade edge devices, typically equipped with 5G RedCap or URLLC-capable modems, interface with field equipment (e.g., PLCs, inverters, controllers) through industrial protocols such as Modbus TCP/RTU. These devices collect high-frequency telemetry data, process it locally in real time, and securely transmit updates to edge or cloud platforms.

In the opposite direction, control instructions are delivered from centralized platforms to the edge devices over Ultra-Reliable Low Latency Communication (URLLC) slices, ensuring deterministic latency and guaranteed command execution within regulatory or operational timeframes. The architecture supports both human-in-the-loop teleoperation—where remote operators can execute precise movements or adjustments with sub-5 to sub-10 ms responsiveness—and automated actuation driven by AI-based decision systems.

By integrating predictive models hosted at the edge or in the cloud, the enabler can forecast system behavior based on real-time telemetry and external data sources (e.g., weather conditions or workload predictions), and proactively issue control actions to improve efficiency, safety, or stability. To maintain resilience under variable connectivity, edge devices incorporate local buffering, offline operation, and automatic synchronization after recovery.

Together, these features make the enabler suitable for deployment in operational environments with strict requirements for security, ultra-high reliability, and response times, enabling both manual teleoperation and automated actuation across diverse vertical domains.

5.6.2 Use case association and contributing partners

Use cases E3, P5, and T5 will implement mechanisms for real-time actuation and teleoperation. Table **5-6** presents the entities involved and their corresponding use cases.

Table 5-6 Mapping between Real-Time Actuation/Teleoperation Enablers and UCs

	Real-Time Actuation/Teleoperation
E3	SIM
P5	ORO
T5	CERTH/ThPA

The **E3 use case** focuses on enabling real-time monitoring, forecasting, and remote control of solar energy systems using advanced B5G/6G communication and edge-cloud computing. Key to this setup is the system's ability to execute commands and make control decisions instantly or within milliseconds—especially in response to requests from national energy authorities or based on AI-driven predictions. Real-Time Actuation and Teleoperation are supported through a combination of Quality of Service (QoS) mechanisms and AI-as-a-Service (AIaaS), which together enable intelligent control decisions and low-latency actuation over the 5G infrastructure.

- **QoS**: The E3 use case requires a reliable mechanism to prioritize energy-related communication. QoS ensures that critical data (like control signals or real-time telemetry) is transmitted with low latency and high reliability. This is crucial when adjusting inverter parameters based on energy forecasts or emergency control instructions from the grid operator. Without QoS, latency spikes or bandwidth competition with non-critical traffic could delay actuation commands.
- **AiaaS:** AI-as-a-Service is used to deploy forecasting and control models that help optimize solar energy production. These models analyze historical data and weather forecasts to predict energy output and recommend actions. Executing these models closer to the edge allows faster response times when sending inverter reconfiguration commands. This is important for achieving sub-second reaction times, especially when quick action is required due to changing grid conditions or production inefficiencies.

The **P5 use case** focuses on field-deployable private B5G/6G networks that ensure real-time mission-critical communication capabilities for emergency response teams in remote or disaster-affected areas. These networks must support ultra-reliable low-latency communication (URLLC), dynamic network slicing, and localized actuation even when disconnected from macro-core or terrestrial infrastructure. Real-time actuation and teleoperation enablers ensure uninterrupted service delivery for MCX services (voice, video, data), support on-site decision-making, and maintain coordination across mobile and heterogeneous emergency teams.

- **QoS:** The QoS enabler is essential to ensure that mission-critical services-especially voice (MC-PTT), video (MC-Video), and telemetry (MC-Data)—receive the required prioritization.
- **Network Slicing:** Slicing enables separation of traffic for different responder categories (medical, military, firefighters), ensuring predictable performance and service isolation. Each unit gets a tailored slice matching their service needs, which is vital for managing mixed traffic types and avoiding interference across teams during high-pressure missions.
- **AIaaS:** While not central, AIaaS can support video prioritization or alert filtering in edge setups. For example, AI models might classify incoming video streams and assign them priority based on detected activity (e.g., fire or movement), enhancing bandwidth allocation and responder focus.

The **T5 use case** involves the remote-control operation of a Ship-to-Shore crane in its premises, utilizing a dedicated B5G/6G network. Unlike semi-automated processes, teleoperated crane movements involve human-in-the-loop control, where delays, jitter, or loss of precision directly affects container handling accuracy, port efficiency, and operational safety. To achieve real-time actuation and teleoperation of an STS crane instantaneous responsiveness, continuous situational awareness, and uncompromised safety must be reassured.

- **QoS:** is essential to ensure real-time responsiveness, reliability, and safety assurance required for remote human-in-the-loop operation of STS cranes, enabling precise container handling with the same effectiveness as on-cab control. This means that high performance connectivity is required, delivering data (advanced sensors and video systems) with low latency, high reliability and availability
- **AIaaS:** can coordinate actuation across all devices and sensor data streams, positioned in the edge, where all real-time control functions, sensor fusion, video encoding, and safety applications take place.

5.6.3 Design, development and implementation

The teleoperation enabler provides a framework for deterministic remote control of IoT endpoints, cyber-physical systems, and critical infrastructures, as we can see in Figure 5-23. The architecture comprises multiple actuation strata: (i) operator-driven control through Human – Machine Interfaces (HMIs), dashboards, and Remote Procedure Calls (RPCs); (ii) event-driven actuation via rule-based engines processing real-time telemetry. Both human-in-the-loop scenarios (e.g., direct manipulation of robotic assets, cranes, or unmanned vehicles) and closed-loop automation scenarios are supported. To meet stringent requirements on latency, reliability, and security, the enabler integrates Quality of Service (QoS) enforcement, deterministic transport mechanisms, and private network capabilities. Furthermore, AI/ML-assisted decision support modules are embedded to optimize control policies and ensure predictable execution of commands within constraints.

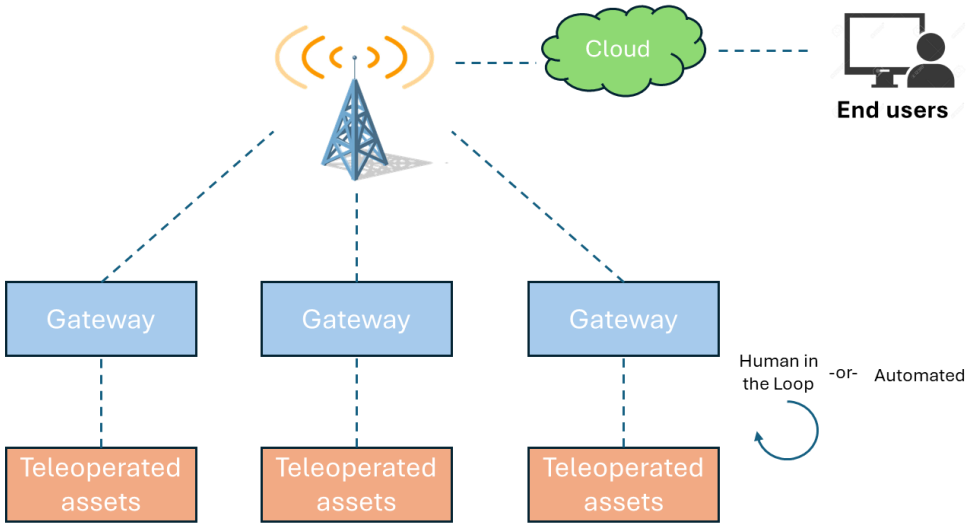

Figure 5-23 Remote control and operation

The **P5 use case** for Real-time Actuation and Teleoperation will be designed, developed, and implemented in alignment with the broader roadmap of the emergency private B5G/6G tactical communication network. During the design phase, the focus will be on defining the mission-critical requirements, mapping communication flows between field units and command centers, and specifying the interfaces among the portable RAN, standalone 5G core, and edge servers. Special attention will be given to adherence with 3GPP MCx standards, resource prioritization techniques, and the integration of multicast/broadcast capabilities to ensure ultra-low latency and highly reliable connectivity for multiple public safety stakeholders. These details will be consolidated in Deliverable D5.1 (M17).

The development phase will advance from the design outputs, closely following the tactical deployment plan. This stage will concentrate on integrating MCx services with ORO's portable B5G infrastructure while setting up secure connections over satellite and terrestrial backhaul. Orchestration functions will be hosted on the edge servers to support dynamic slice control and enable seamless transitions between centralized and localized core operations. Finally, the implementation stage will embed the enabler within the tactical bubble setup, with validation carried out through live field exercises. Testing activities will target KPIs essential for emergency communications, such as sub-10 ms latency for MCx exchanges, 99.99% service reliability under high-load conditions, and continuous operation in both satellite-supported and isolated scenarios. This stepwise process ensures coherence between design, development, and deployment, delivering a robust and field-proven solution for real-time actuation and teleoperation in mission-critical contexts.

The **E3 use case** design, development, and implementation of the real-time actuation/teleoperation enabler will be carried out in alignment with the overall use case roadmap. The first step will focus on design, including detailed specification of functional requirements, communication flows, edge-to-cloud interfaces, and compliance with regulatory constraints. This phase will also consider interoperability with different inverter models, resilience mechanisms, and integration of forecasting modules. A comprehensive description of the use case planning will be documented in Deliverable D5.1 (M17). Building on this, the project has already completed the architectural design and acquired the necessary communication devices. Preparations for preliminary integration and validation in the SIMTEL laboratory are ongoing, laying the foundation for subsequent field deployment.

The development steps will include the integration of 5G RedCap edge devices with inverter communication protocols, the setup of secure MQTT/TLS data channels, and the deployment of control orchestration functions in ORO's edge cloud platform. In the next phase, the forecasting engine, dashboards, and operator APIs will be integrated, followed by full-scale testing under real operational conditions. The final stage of the implementation will culminate with the full integration of the enabler into the solar energy monitoring and control use case (see Figure 5-24), including testing under real operational conditions. Trials are planned for the second part of the project, where system performance will be validated against target KPIs such as sub-10 ms latency, 99.99% reliability, and regulatory response compliance. This staged approach ensures that design, development, and integration remain consistent with the project's overall goals while delivering a reliable and demonstrable solution for real-time actuation in renewable energy operations.

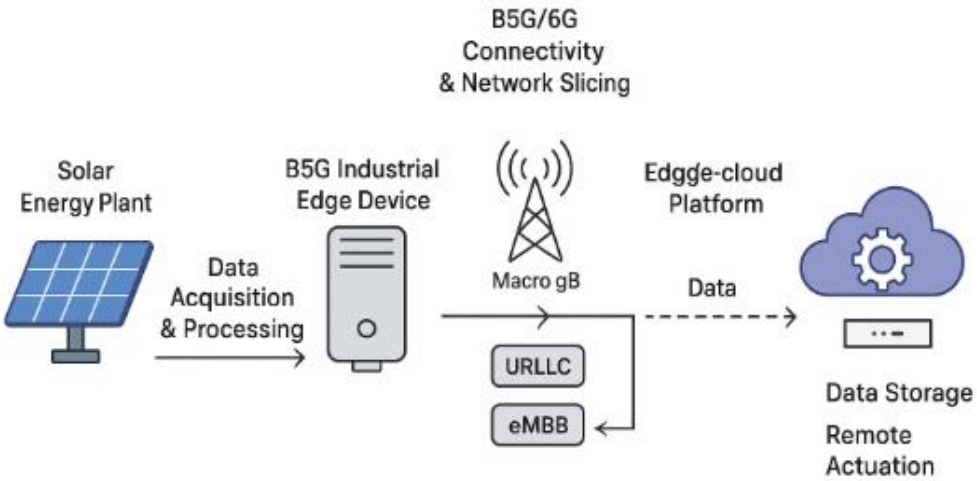

Figure 5-24 Solar energy monitoring, control and predictions using use case overview

The **T5 use case** focuses on the remote-control operation of a Ship-to-Shore (STS) crane within port premises, leveraging the capabilities of a dedicated Beyond 5G (B5G) or 6G communication network. This advanced connectivity infrastructure enables the reliable transmission of high-volume data streams required for crane teleoperation, including video feeds, sensor readings, and control signals. Unlike traditional automation,

where processes may follow pre-programmed instructions, the teleoperation of STS cranes demands real-time responsiveness to ensure that every movement corresponds precisely to operator input. This responsiveness is essential for maintaining continuous situational awareness, supporting instantaneous actuation, and guaranteeing the safe handling of heavy and valuable cargo containers in dynamic and often unpredictable port environments. A key distinction of this use case lies in its human-in-the-loop nature, where skilled operators remotely control the crane's functions rather than relying on fully autonomous systems. This setup means that even minor disruptions in network performance—such as latency, jitter, or packet loss—can have significant operational consequences. Delayed responses could compromise the precision of container movements, leading to inefficiencies in loading and unloading operations, or worse, creating hazardous situations that threaten both personnel and infrastructure. By deploying a robust B5G/6G network, the system ensures ultra-reliable low-latency communication (URLLC), minimizing risks while maximizing efficiency. As a result, the T5 use case not only demonstrates the potential of next-generation connectivity in critical industrial applications but also highlights its role in enhancing safety, productivity, and competitiveness in modern port operations. Figure 5- shows the Teleoperation of crane in T5.

The first version of the teleoperation platform is currently under development. During the 1st year, testing will take place in a controlled laboratory environment. In the 2nd year, field deployments are planned to support preliminary experimentation, followed by large-scale final trials in the 3rd year.

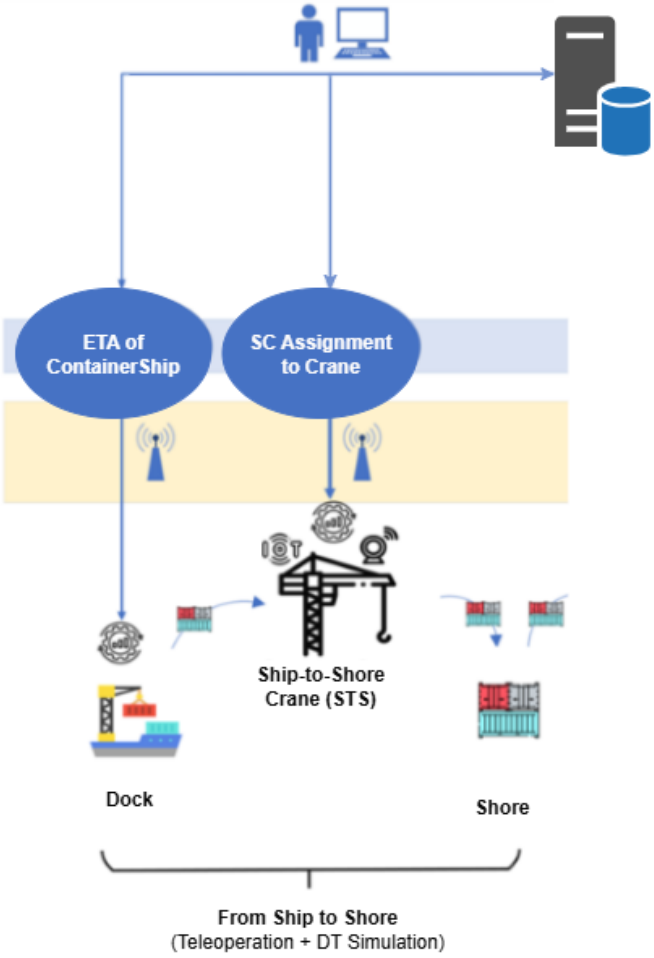

Figure 5-25 Teleoperation of crane in T5

5.7 Summary

This section has presented the leading IoT and Localization enablers developed within AMAZING-6G, highlighting their role in supporting advanced capabilities across multiple domains. The enablers address both the challenges of accurate localization and the need for scalable, interoperable IoT infrastructures. Hybrid approaches that combine 5G with GNSS and RTK correction enable centimeter-level accuracy, while cooperative positioning integrates perception data from LiDAR, radar, and cameras with V2X communications to provide reliable situational awareness in complex environments. In parallel, IoT connectivity and infrastructure components such as gateways, 5G RedCap modems, and heterogeneous sensors ensure that data can be acquired efficiently, transmitted securely, and integrated seamlessly with edge and cloud platforms. The collected information is processed through telemetry and monitoring solutions, enabling real-time analytics, anomaly detection, and decision support. In addition to these, IoT service platforms offer orchestration, lifecycle management, and the exposure of device capabilities through APIs, ensuring interoperability and integration across heterogeneous systems. Contextual awareness mechanisms further enrich sensing data with external inputs, enhancing safety, prediction, and adaptive operation in dynamic scenarios. Finally, real-time actuation and teleoperation enablers demonstrate how edge-to-cloud control loops can be implemented for critical infrastructures, from renewable energy plants to port cranes, guaranteeing responsiveness, reliability, and secure operation. Together, these enablers form a cohesive technological foundation that brings sensing, connectivity, positioning, and control into a unified framework, paving the way for advanced 6G applications.

6 Conclusions

In this deliverable we have identified the technology enablers of the project, based on the requirements (KPIs and KVIs) of the AMAZING-6G use cases, with the aim that the identified enablers may also be used for other vertical use cases with similar requirements. In order for their integration in the AMAZING-6G trials and pilots, the identified technology enablers are particularly those where there are innovation opportunities and in which the project partners have strong expertise.

The AMAZING-6G technology enablers have been grouped in the following four (4) categories:

- Communication enablers (Chapter 2), including radio, transport and core network enablers.
- Compute as a Service enablers (Chapter 3), including the enablers in the compute continuum.
- Application enablers and AI (Chapter 4), including network exposure APIs and AI-driven application services.
- IoT and localization enablers (Chapter 5), including IoT platforms, IoT devices and localization technologies.

Some of the technology enables are composed of multiple sub-enablers which have different technical focuses, use case associations but similar purposes (see Table 1-2).

We admit that some of the technology enablers are relevant to each other or even overlap. Examples are: (1) ZSM may be used for the management of both communication resources and compute resources, and in an more ideal case communication and compute resources may be jointly managed for better overall performance. (2) “Network slicing” is identified as a separate technology enabler, focusing on how network slices may be configured and implemented, while slices may be seen as part of the resources to be managed in the context “communication resource management”. (3) “Digital Twin” (both application and network levels) is identified as a separate technology in the category of “Application enablers and AI”, while it’s also relevant for the collection of IoT data. (4) ISAC is listed as one of the communication enablers, but the sensing part of ISAC is out of the scope of communication and may be related to IoT or Internet of Sense. The development progress of these enablers will be followed closely to ensure necessary coordination among the relevant enablers and involved partners.

For each of the technology (sub-)enablers, a brief description has been given introducing the concept of the (sub-)enabler. This was followed by an analysis of its relevance for the associated AMAZING-6G use cases. High-level use case independent designs have been given for all the identified technology (sub-)enablers, showing that they have potential to be used for other vertical use cases with similar requirements but not covered by the project. Aiming to be integrated in use case specific trials and pilots, we have also provided use case specific design and implementation plans for the (sub)enablers.

Being at an early stage of the project (M10), only some initial implementation and test results have been provided for a small number of the (sub)enablers. Eventual fine-tuning of the designs and final implementations will be reported in Deliverable D3.2 (M33). For the sake of earlier and in-time reporting, some may also be captured in the vertical-focused Deliverables D4/5/6.1 (M17), from the perspective of potential integration with the associated use cases. This is possible since the same partners are involved in WP3 and WP4/5/7.

7 References

[1] [https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/001/01.01.01_60/gs_ZSM001v010101p.pdf] – ETSI GS ZSM 001 – ZSM Requirements

[2] TS 122 186 - V16.2.0 - 5G; Service requirements for enhanced V2X scenarios (3GPP TS 22.186 version 16.2.0 Release 16)

[3] Zero Touch Management: A Survey of Network Automation Solutions for 5G and 6G Networks | IEEE Journals & Magazine | IEEE Xplore

[4] GS ZSM 002 - V1.1.1 - Zero-touch network and Service Management (ZSM); Reference Architecture

[5] OpenFlow : https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf

[6] Docker: https://www.docker.com

[7] LXC: https://linuxcontainers.org

[8] Kubernetes: https://kubernetes.io

[9] GR ZSM 011 - V2.1.1 - Zero-touch network and Service Management (ZSM); Intent-driven autonomous networks; Generic aspects

[10] I2BN: Intelligent Intent Based Networks | Journal of ICT Standardization

[11] RedCap/eRedCap – standardizing simplified 5G IoT devices - Ericsson

[12] Pascal Jörke, Hendrik Schippers and Christian Wietfeld, " Empirical Comparison of Power Consumption and Data Rates for 5G New Radio and RedCap Devices", 2025 IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, USA, CCNC2025_AuthorsVersion.pdf

[13] 3GPP TS 29.522, "5G System; Network Exposure Function Northbound APIs; Stage 3", v19.4.0, September 2025.

[14] GSMA, "Operator Platform: Requirements and Architecture", Version 9.0, May 2025

[15] P. Soto, M. Camelo, et al., "Network Intelligence for NFV Scaling in Closed-Loop Architectures," in IEEE Communications Magazine, vol. 61, no. 6, pp. 66-72, June 2023, doi: 10.1109/MCOM.001.2200529.

[16] N. Slamnik-Krieštorac, M. Camelo, et al., "AI-Empowered Management and Orchestration of Vehicular Systems in the Beyond 5G Era," in IEEE Network, vol. 37, no. 4, pp. 305-313, July/August 2023, doi: 10.1109/MNET.008.2300024.

[17] L. Bonati, M. Polese, S. D'Oro, S. Basagni and T. Melodia, "NeutRAN: An Open RAN Neutral Host Architecture for Zero-Touch RAN and Spectrum Sharing," in IEEE Transactions on Mobile Computing, vol. 23, no. 5, pp. 5786-5798, May 2024, doi: 10.1109/TMC.2023.3311728.