International Telecommunication Union

ITU-T Technical Specification

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(September 2026)

ITU-T Focus Group on AI Native for Telecommunication Networks

Use cases for AI native telecommunication networks

Summary

The WG2 deliverable “Technical Specification - Use cases for AI native telecommunication networks” provides use case descriptions, analysis of technical challenges or gaps, as well as indicate assumptions and potential requirements for each use case. The scope of WG2 deliverable aligns with the use case analysis and gap analysis for use cases outlined in the ToR. Based on the 39 use cases received by WG2, this document contains the baseline text of draft Technical Specification on Use cases for AI Native Telecommunication Network, which incorporates 21 use case contributions based on the progress of discussions and approvals in WG2 meetings.

Keywords
AI Native Network; use cases

Contributors:

Huiqing Sun
China Mobile
China
Tel: 18701380263
E-mail: sunhuiqing@chinamobile.com

Ranjana Sivaram
DoT
India
E-mail: ranjana.sivaram@gov.in

Introduction

FG-AINN WG2 deals with use cases showcasing the use of artificial intelligence native networks including use cases pertaining to both AI for Network and Network for AI. This output deliverable document presents all the use cases along with their requirements, assumptions etc. that will be used as an input to FG AINN WG3, which is involved in the development AINN architecture. A total of 39 contributions has been received and discussed and are at various stages of finalisation. This version of the document includes 21 use cases that have been approved or are nearing approval. A summary of these 39 use cases is presented in the Table below.

No.	Name of Contribution	Source	Document link (latest version)	Status
1	Sustainable Networks for Disaster Handling	Centre Tecnològic de Telecomunicacions de Catalunya (CTTC)	16-FG-AINN-I-WG2-006-R3	Approved
2	LLM agents-powered intent-based network slicing for industrial automation	Turk Telekomunikasyon Inc.	18-FG-AINN-I-WG2-005-R3	Under discussion
3	AI-native Networks as Applied in Verticals	State Grid Corporation of China (SGCC)	48-FG-AINN-I-148-R3	Approved
4	Transformer-based Interference prediction*	China Mobile Communications Co. Ltd.	FG-AINN-I-063	Under discussion
5	Network-native Personal AI Assistant service	China Mobile Communications Co. Ltd.	80-FG-AINN-I-062-v2	Approved
6	Proactive AN optimization for embodied intelligence	China Mobile Communications Co. Ltd.	FG-AINN-I-145	Approved
7	AI-Agent-Driven Intelligent Network	China Mobile Communications Co. Ltd.	79- FG-AINN-I-060-v2	Approved
8	Use case on communication and coordination for multiple AI Agents	China Mobile Communications Co. Ltd.	47-FG-AINN-I-059-R3	Approved
9	Use case – LLMs in AI native telecommunication networks	Huawei Technologies	31-FG-AINN-I-111-R1	Approved
10	Use case – AI Training as a Service in AI native telecommunication networks	Huawei Technologies	30-FG-AINN-I-110-R1	Approved
11	Use case – AI Inference as a Service in AI native telecommunication networks	Huawei Technologies	29-FG-AINN-I-109-R1	Approved
12	AI Agent for Automatic Network Configuration*	China Mobile Communications Co. Ltd.	FG-AINN-I-056	Under discussion

13	High Performance Inter-Edge-Cloud Network for Large-scale LLM Inference*	China Mobile Communications Co. Ltd.	FG-AINN-I-057	Under discussion
14	AI-Native Proactive Network Slice Marketplace for Dynamic Service Ecosystems	SRM Institute of Science and Technology	49-FG-AINN-I-104-R4	Approved
15	Use Case – Native API support for dynamic distribution of inference tasks in an Edge-Cloud continuum	Telecommunications Standards Development Society, India (TSDSI)	34-FG-AINN-I-102-R1	Approved
16	AI-Native Digital Twins for Cognitive Network Optimization and Management	Telecommunications Standards Development Society, India (TSDSI)	75-FG-AINN-I-103-R4	Approved
17	AI-driven network performance modelling and prediction	China Mobile	126-FG-AINN-I-146-final-version	Approved
18	Use case – Creation of data for machine learning in AI-native networks	Universitat Pompeu Fabra (UPF)	60-FG-AINN-I-WG2-015-R1	Approved
19	Use Case on - Cooperative Communication in Maritime AI-native Networks	SRM Institute of Science and Technology	82-FG-AINN-I-WG2-019-R4	Approved
20	Context-Aware AI-Native Traffic Prioritization Using Multi-Modal Device Intelligence*	SRMIST Kattankulathur	70-FG-AINN-I-WG2-020-R2	Under discussion
21	Smart Network Planning and Optimization for Next-Generation Wireless System using Artificial Intelligence Enabled Technique*	SRM Institute of Science and Technology, Chennai	83-FG-AINN-I-WG2-021-R3	Under discussion
22	Lightweight Trust: Enhancing Efficiency and Privacy with zk-SNARKs	SRM Institute of Science and Technology	89-FG-AINN-I-WG2-022-R6	Approved
23	Intelligent Transport Systems	University of Glasgow	FG-AINN-I-244	Approved
24	AI-driven adaptive Digital Twin models for future intelligent network optimization	SRM Institute of Science and Technology, Chennai	90-FG-AINN-I-WG2-024-R4	Under discussion
25	AI-Native Task-Oriented Robotic Teleoperation	University of Glasgow	113-FG-AINN-I-WG2-035-R1	Approved
26	Semantic Communications Services for Agents	Orange, SNS-JU 6GARROW	135-FG-AINN-I-WG2-036-R1	Approved
27	Overhead reduction through Semantic Closed Control	HPE, SNS-JU 6GARROW	136-FG-AINN-I-WG2-037-R1	Under discussion

	Loop for Private 5G/6G Management Systems			
28	Cultivating Radio Measurements into Value-Added Services for Spectrum Sharing	Alto, SNS-JU 6GARROW	137-FG-AINN-I-WG2-038-R1	Presented
29	Use cases contribution from SNS-JU 6G-TWIN project-Teleoperated Driving-R1	SNS-JU 6G-TWIN	123-FG-AINN-I-WG2-039-R1	Under discussion
30	Use cases contribution from SNS-JU 6G-TWIN project-Energy Saving	SNS-JU 6G-TWIN	167-FG-AINN-I-WG2-040-final version	Approved
31	Distributed Inference in Programmable User Planes	ORIGAMI project	132-FG-AINN-I-WG2-041-R1	Under discussion
32	NATWORK Use Cases from SNS-JU SNWG	The NATWORK consortium	162-FG-AINN-I-WG2-042-R7	Under discussion
33	Use cases Contribution from SNS-JU 6G-BRICKS project-Part 1	SNS-JU 6G-BRICKS	115-FG-AINN-I-WG2-044	Not presented
34	Use cases Contribution from SNS-JU 6G-BRICKS project-Part 2	SNS-JU 6G-BRICKS	116-FG-AINN-I-WG2-045	Not presented
35	Distributed Continuum towards Pervasive computing	SNS-JU 6G-INTENSE	134-FG-AINN-I-WG2-046-R1	Under discussion
35+1	Distributed Continuum towards Pervasive Computing	SNS-JU 6G-INTENSE	FG-AINN-I-241	Under discussion
36	Joint Communication & Sensing for Optimal User Tracking in Metaverse	SNS-JU 6G-INTENSE	FG-AINN-I-242	Under discussion
36+1	Fully autonomous Metaverse FCAPS, sensing and continuum abstraction	SNS-JU 6G-INTENSE	FG-AINN-I-243	Under discussion
37	Use cases Contribution from SNS-JU NexaSphere project	SNS-JU NexaSphere	139-FG-AINN-I-WG2-048-R1	Under discussion
38	Extended Reality Applications leveraging an Agentic AI System	InterDigital Europe Ltd	131-FG-AINN-I-WG2-049-R1	Under discussion
39	A Physically Grounded Trust Mechanism for Multi-Agent Space-Ground Orchestration	April Labs	165-FG-AINN-I-289-R2	Under discussion

*[Editors note: The contributions with No. +1 means the original version was split into 2 contributions when updating. The contributions with * were discussed, however further discussions and progress have not been possible due to lack of updated versions.]*

Proposal

The Technical Specification was developed based on the skeleton of technical specification (FG-AINN-I-WG2-002). This is base text of draft Technical Specification “Use cases for AI Native Telecommunication Network”, which is based on the 21 use cases listed in the main body of this TS. The proposed draft technical specification is attached as Annex I.

Annex I.

Table of Contents

1 Scope 9

2 References 9

3 Terms and definitions 9

4 Abbreviations 10

5 Conventions 11

6 Introduction 11

7 Use cases 11

Finalized Use Cases 11

7.1 Use case 001: Sustainable Networks for Disaster Handling 11

7.2 Use case 002: AI Inference as a Service in AI native telecommunication networks 17

7.3 Use case 003: AI Training as a Service in AI native telecommunication networks 21

7.4 Use case 004: LLM-based value-added services in AI native telecommunication networks 25

7.5 Use Case 005: AI-native Networks as Applied in Verticals 29

7.6 Use case 006: Use case on communication and coordination for multiple AI Agents 39

7.7 Use case 007: AI-Native Proactive Network Slice Manager for Dynamic Service Ecosystems 42

7.8 Use case 008: Creation of data for machine learning in AI-native networks 49

7.9 Use case 009: AI-Native Digital Twins for Cognitive Network Optimization and Management 52

7.10 Use Case 010: Native API support for dynamic distribution of inference tasks in an Edge-Cloud continuum 62

7.11 Use Case 011: Lightweight Trust: Enhancing Efficiency and Privacy with zk-SNARK 65

7.12 Use case 012: AI-driven network performance modelling and prediction 69

7.13 Use case 013: Cooperative Communication in Maritime AI-native Networks 73

7.14 Use case 014: AI-Native Task-Oriented Robotic Teleoperation 84

7.15 Use Case 015: Intelligent Transport Systems 91

7.16 Use Case 016: AI-Agent-Driven Intelligent Network 100

7.17 Use Case 017: Proactive AN optimization for embodied intelligence 103

7.18 Use Case 018: Network-native Personal AI Assistant service 105

7.19 Use Case 019: Semantic Communications Services for Agents 109

7.20 Use Case 020: Federated Cross-Domain Energy-Efficient Network Optimization 112

7.21 Use case No. A1: LLM agents-powered intent-based network slicing for industrial automation 119

8 Summary 123

Technical Specification

Use cases for AI Native Telecommunication Networks

1 Scope

This Technical Specification analyses AI native use cases for telecommunication networks. It provides use case descriptions and indicates assumptions and potential requirements for each use case. The use cases are divided into categories, and the actors, roles and interactions related to each use case are provided.

2 References

This clause lists the ITU references cited in this document.

[ITU-T Y Suppl.55] ITU-T Y.3170-series (2019), *Machine learning in future networks including IMT-2020: use case.*

[ITU-T Y.3172] Recommendation ITU-T Y.3172 (2019), *Architectural framework for machine learning in future networks including IMT-2020.*[ITU-T TR-GenAI-Telecom Networks] Technical Report ITU-T TR-GenAI-Telecom Networks (2025), *Requirements and methodology for deploying and assessing Generative AI models in telecom networks*.

3 Terms and definitions

3.1 Terms defined elsewhere

This contribution uses the following terms defined elsewhere:

1.1.1 Machine learning (ML) [ITU-T Y.3172]**:** Processes that enable computational systems to understand data and gain knowledge from it without necessarily being explicitly programmed.

1.1.2 Artificial intelligence (AI) [b-ETSI GR ENI 004]: Computerized system that uses cognition to understand information and solve problems.

NOTE 1 – [b-ISO/IEC 2382-28] defines AI as "an interdisciplinary field, usually regarded as a branch of computer science, dealing with models and systems for the performance of functions generally associated with human intelligence, such as reasoning and learning".

NOTE 2 – In computer science AI research is defined as the study of "intelligent agents": any device that perceives its environment and takes actions to achieve its goals.

NOTE 3 – This includes pattern recognition, the application of machine learning and related techniques.

NOTE 4 – Artificial-intelligence is the whole idea and concept of machines being able to carry out tasks in a way that mimics human intelligence and would be considered "smart".

1.1.3 ML model lifecycle [3GPP TR 22.850]**:** The end-to-end process typically consisting of data processing, model training, model testing, model deployment, model inference, model monitoring and model maintenance.

NOTE 1: Data processing includes collecting and preparing the data for model training and model inference.

NOTE 2: Model training includes training and validating the model before model deployment.

NOTE 3: Model testing includes testing the model before model deployment.

NOTE 4: Model deployment includes making a trained ML model available for use in the target environment.

NOTE 5: Model monitoring includes observing the performance of the model during the model maintenance process.

NOTE 6: Model maintenance includes updating the model, retraining the model and (de-)activating the model.

1.1.4- AI/ML inference [3GPP TS 28.105]**:** a process of running a set of input data through a trained ML model to produce set of output data, such as predictions.

NOTE 5: The inference represents the process to realize the AI capabilities by utilizing a trained ML model and other AI enablers if needed, hence the AI/ML prefix is used when referring to inference as compared to training and testing.

1.1.6. ML model training [3GPP TS 28.105]**:** a process performed by an ML training function to take training data, run it through an ML model algorithm, derive the associated loss and adjust the parameterization of that ML model iteratively based on the computed loss and generate the trained ML model.

1.1.7. AI native telecommunication network [FG-AINN-WG1-04]**:** a network integrating AI capable of interacting with its environment, enabling novel value-added services, and enhancing the performance of network and services.

NOTE1: Integrating AI refers to use AI technologies as enablers through telecommunication network design, deployment, operation, and maintenance.

NOTE2: Design in AI-Native telecommunication networks includes the following types: human-led design, AI-assisted design, AI-driven design.

NOTE3: Enhance performance refers to higher level of autonomy and efficiency.

NOTE4: AI-native telecommunication network environment refers to all components and entities external to the network interacting with the AI-native telecommunication network including humans, devices and applications.

NOTE5: AI-native telecommunication network value-added services refer to added-value services integrating AI and leveraging telecommunication network communication and computing capabilities. This extends the definition of value-added services based on [3GPPTR 21.905].

1.1.8. AI agent in AI-native telecommunication network [FG-AINN-I-WG1-AI Agent]**:** An entity in the AI-native telecommunication network able of interacting with the AI-native telecommunication network environment and that uses AI task-specific capabilities for achieving the goals of the AI-native telecommunication network and its environment.

NOTE 1- The AI agent AI task-specific capabilities may include, but are not limited to, decision-making, reasoning, forecasting, clustering, classification, self-learning, acquiring contextual information, natural language processing, and complex tasks decomposition.

NOTE 2**:** Interactions may include, but are not limited to, data collection, usage of tools (e.g., search engines, databases, devices, other AI agents external to the network) through related APIs, and sharing the output of the agent decisions with AI-native telecommunication network and its environment through an interface.

3.2 Terms defined here

- **AI for Network [**adapted by 3GPP TR 22.850**]:** Work to optimize the telecom network performance by employing AI/ML on telecom network-specified functionalities and capabilities.
- **Network for AI [**adapted by 3GPP TR 22.850**]**: Work to provide telecom network support for AI/ML-based applications and services.

4 Abbreviations

This document uses the following abbreviations and acronyms:

AI	Artificial Intelligence
AN	Access Network
API	Application Programming Interface
DT	Digital Twin
E2E	End to End
GRU	Gate Recurrent Unit

KPI	Key Performance Indicator
LLM	Large Language Model
LSTM	Long Short-Term Memory
ML	Machine Learning
NDT	Network Digital Twin
QoE	Quality of Experience
QoS	Quality of Service
SLA	Service Level Agreement
UE	User Equipment

5 Conventions

In this Technical Specification, in alignment with the conventions of [ITU-T Y Suppl.55] possible requirements which are derived from a given use case, are classified as follows:
The keywords "it is critical" indicate a possible requirement which would be necessary to be fulfilled (e.g., by an implementation) and enabled to provide the benefits of the use case.
The keywords "it is expected" indicate a possible requirement which would be important but not absolutely necessary to be fulfilled (e.g., by an implementation). Thus, this possible requirement would not need to be enabled to provide complete benefits of the use case.
The keywords "it is of added value" indicate a possible requirement which would be optional to be fulfilled (e.g., by an implementation), without implying any sense of importance regarding its fulfilment. Thus, this possible requirement would not need to be enabled to provide complete benefits of the use case.

6 Introduction

The advent of powerful generative artificial intelligence (AI), particularly large language models (LLMs), has driven a fundamental transformation in network architecture and technological innovation. The future network design is rapidly transitioning from “AI-assisted” to “AI-native”. The AI-native telecommunication encompasses AI for network and Network for AI, with AI-native technology serving as core enabling technology to support design of architecture, functionality and interface in the telecommunication network.
This Technical Specification provides detailed descriptions of each selected use case for further analysis of use case requirements. Each use case is outlined by articulating its primary objectives and core motivation. It dives into AI-native approaches to explain how the use case benefits from AI-native approach. Each use case will analyse the limitations of existing technologies, clarifying why traditional approaches cannot achieve the equivalent outcomes. In addition, the expected impact the AI native networks and the use case requirements will be introduced.

7 Use cases

Finalized Use Cases

7.1 Use case 001: Sustainable Networks for Disaster Handling

7.1.1 Use case summary

This use case focuses on the deployment of sustainable networks specifically designed for disaster handling. The main motivation behind this use case stems from the need for robust and responsive communication networks during emergencies, where traditional infrastructures may be compromised.

Use case id	FG-AINN-usecase-001
Use case name	Sustainable Networks for Disaster Handling

Created by	Dr. Engin Zeydan (UNITY-6G (https://unity-6g.eu/) Project Coordinator, CTTC)
Creation date	6 February 2025
Use case context	Discussions in the WG2 meeting regarding the contribution [FG-AINN-I-WG2-006], and [FG-AINN-I-WG2-006-R3] was approved in the 6th WG2 meeting (7th April, 2025).
Use case description	This use case focuses on deploying an integrated, resource-shared edge-cloud continuum network that efficiently manages and utilizes computing resources across edge networks during disaster scenarios. The use case aims to maintain continuous communication services and the rapid restoration of networks to operational states post-disaster or transitioning them to new, optimized states. Traditional technologies lack the adaptability and environmental sustainability that Artificial Intelligence (AI)--driven systems can provide, making them unsuitable for the dynamic requirements of disaster management. Utilizing a distributed AI-native approach, this use case aims to leverage intelligent, automated systems to optimize network resources, predict network load in crisis situations, and ensure reliable communication for emergency responders and affected populations also with the support of renewable energy sources. By leveraging distributed AI-native approach, the network will be equipped with decisions that can be done locally to dynamically deploy edge applications, manage traffic flows accurately, and ensure stable, adaptive communication paths between edge client applications and network services, which is crucial for disaster management and response. The expected impact of implementing AI-native networks will be enhanced operational efficiency, reduced downtime, and improved decision-making through local data-driven insights in emergency scenarios. For large-scale geo-distributed systems, determining how to appropriately introduce intelligence in networking is the key to high-efficiency operation. In this direction, the paper [b-Yao2019] explores two deployment paradigms (centralized vs. distributed) for AI-based networking. An AI-native network slicing architecture for 6G networks [b-Wu2022] is presented to enable the synergy of AI and network slicing, thereby facilitating intelligent network management and supporting emerging AI services. The paper [b-Yang2022] investigates distributed edge intelligence challenges, requirements, and trends in future 6G networks. The paper [b-Wang2022] develops an AI-native VhetNets-enabled framework to provide the anomaly detection service for ubiquitous IoT, whose implementation is assisted by intelligent network management functionalities. A comprehensive survey [b-Duan2022] is provided on the distributed artificial intelligence (DAI) empowered by end-edge-cloud computing (EECC), where the heterogeneous capabilities of on-device computing, edge computing, and cloud computing are orchestrated to satisfy the diverse requirements raised by resource-intensive and distributed AI computation. Unlike state-of-the-art (SoTA) systems that typically add AI components in a modular or post-processing fashion with static model pipelines and centralized control, the proposed AI-native architecture is designed from the outset to integrate distributed intelligence across all layers—

	particularly through the synergistic operation of the Monitoring System (MS), Analytics Engine (AE), and Decision Engine (DE), and tight integration with actuators (ACT) for taking local closed loop decisions [b-Blanco2023]. The MS ensures continuous real-time data collection from geo-distributed sensors, mobile applications, and IoT devices, enabling situational awareness even in volatile or degraded environments. The AE dynamically processes this multi-modal data, applying context-aware AI models to detect anomalies, forecast risks, or assess resource availability. This feeds into the DE, which applies policy-driven logic to determine the most appropriate actions—ranging from automated alert dissemination to infrastructure control—executed through tightly coupled actuator systems [b-Barrachina2024]. This represents a significant advancement over conventional SoTA approaches by embedding intelligence directly into the core networking and decision-making infrastructure. Despite significant recent progress in recognizing the need for AI-enhanced intelligence in such systems, there are still many unsolved complex challenges associated to the analysis of networking data through AI/ML in emerging communication and networking technologies such as 6G networks and edge-cloud computing which hinders its acceptability and adoption in the practice. This use case leverages a unified architecture from UNITY-6G project that aims to develop advanced edge–cloud computing mechanisms to tackle the multifaceted challenges inherent in integrated networks, including device heterogeneity, resource constraints, energy efficiency, communication costs, data privacy, scalability, model accuracy, and robustness. Leveraging collaborative and distributed AI techniques, UNITY-6G aims to revolutionize distributed resource management across edge-cloud resources, energy/computing resources, and integrated communication infrastructure and advocates for a fully distributed and collaborative AI-native approach spanning various domains In disaster event, parts of the network are disconnected, if the AI decisions are centralized, there is a possibility that no critical updates will be taken for the network behaviour since there is no communication with large parts of it. Specifically, in this disaster event, the "live" parts of the network are needed to adapt to the new situation, try to revive the network as soon as possible. For this reason, distributed AI engines are needed in the network to learn and analyse the current network situation and act accordingly. This use case explores the deployment of integrated, resource-shared edge cloud continuum networks for developing an effective disaster response for emergency communication scenarios. It focuses on maintaining always-on services and restoring networks to their pre-disaster or new sustainable states, leveraging distributed AI-native approaches for resilience and rapid adaptability. Utilizing distributed AI-native approach, the network can be designed for high resilience and rapid adaptability, ensuring that critical communication services for emergency response are reliable and always available.
Core concepts / characteristics of AINN (Optional)	Utilization of distributed AI along with renewable energy sources to enhance network flexibility and efficiency, ensuring minimal

	downtime during transitions between emergency and normal operations. This involves: ● Leveraging distributed AI to predict network demands and allocate resources proactively during disasters. ● Distributed AI-enabled Automated systems adjust network configurations in real-time based on changing conditions to maintain service continuity. ● Enabling real-time analytics to monitor edge host resource utilization to ensure optimal resource allocation, preventing overload and underutilization which are critical during disaster responses. ● Integration of renewable energy sources to sustain network operations independently of the traditional power grid, enhancing environmental sustainability. In a hierarchical design, the edge-cloud orchestrator in the architecture can also be used for global decision making to dynamically perceive computing resources and efficiently orchestrate, resolve local conflicts in local decisions and schedule edge applications enhances responsiveness and agility in deploying necessary services.
Actors and Roles (Optional)	- *Emergency Service Operators:* Manage and coordinate emergency response efforts using the network. - *Disaster Response Units:* Utilize the network for real-time communication and data sharing during rescue operations. - *AI System Administrators*: Oversee the AI operations that control network adaptability and resource allocation. - Network Engineers: Responsible for the physical and technical maintenance and deployment of the network infrastructure. - *Edge System Administrators*: Monitor and manage the real-time operation of edge networks, ensuring efficient resource utilization and rapid deployment of necessary applications. - *Edge-cloud orchestrator,* dynamically manages the placement and execution of services across heterogeneous computing resources, balancing latency, bandwidth, and energy constraints. This orchestration is critical for maintaining resilience in geographically distributed and resource-constrained environments. - *Cloud and edge applications* are deployed collaboratively: cloud-based analytics offer deep learning and long-term planning, while edge applications support localized, low-latency inference essential during network disruptions. - *Renewable energy sources* such as solar panels and micro-wind systems to power edge nodes and sensors often paired with battery storage and intelligent energy management, ensure continued system operation during power outages or infrastructure collapse.
Technical challenges or Gaps (Optional)	- Combining different heterogeneous technologies and network components into a cohesive system that can operate under the stress of a disaster.

	- Ensuring that the network meets performance standards under varying and often extreme conditions without degradation of service. - Integrating real-time analytics to dynamically adjust network configurations based on edge resource utilization metrics. - Ensuring SLA compliance rate is upheld during disasters, reflecting the availability and reliability of edge applications under extreme operational conditions. - Improving service deployment success rate and reducing deployment latency are essential for quick responses in emergency scenarios, requiring enhancements in the edge orchestrator's capabilities. - Effectively managing and integrating renewable energy sources to provide a reliable power supply during extended grid failures.
Use case category	AI-native networks applied in vertical industries
Reference	[b-Yao2019], [b-Wu2022], [b-Yang2022], [b-Wang2022], [b-Duan2022], [b-Blanco2023], [b-Barrachina2024]

7.1.2 Assumptions

- Network infrastructure will have the capability to integrate seamlessly with local renewable energy sources (e.g. solar panels).
- Real-time data exchange with local microgrids will be feasible and reliable.
- Emergency protocols will allow for AI-driven decision-making without compromising operational security or data privacy.

7.1.3 Use case requirements

To enable AI-native emergency and disaster response systems over next-generation networks, a layered and multi-dimensional set of requirements must be defined, spanning application, AI, network, operator, and end-user domains.

a. Functional requirements

i. Application-Level Requirements

1. Support multi-modal, real-time data ingestion from heterogeneous sources (IoT sensors, drones, user devices, satellite feeds).
2. Enable context-aware filtering, prioritization, and geo-tagging of emergency-relevant data streams.
3. Open APIs to plug in domain-specific AI models (e.g. earthquake risk mapping)

ii. Operator-Facing Requirements

1. Must be capable of real-time adaptation to fluctuating network conditions and service demands, including disaster-induced degradations with robust network service orchestration.
2. Dashboards and analytics interfaces must provide clear audit trails of AI-driven decisions and system behaviors.
3. Support for flexible, decentralized energy options—including off-grid renewable sources (e.g., solar, wind, battery-backed systems) and smart microgrids—to ensure continuity of critical services during main grid outage. Grid dependency can be a single point of failure in disaster scenarios. Thus, the system must prioritize local energy resilience in addition to sustainability.

iii. End-User Facing Requirements

1. Ensure emergency communication applications on user devices can operate in disconnected or degraded network conditions.
2. For mission-critical alerts or two-way communication (e.g., first responder coordination), ensure consistent low-latency interaction even under stress.

iv. AI and Network Layer Requirements

1. AI Layer Requirements

a. Support for distributed and hierarchical AI models operating at device, edge, and cloud levels
b. Support for modular agents that interconnect and compose flexible decision pipelines across distributed infrastructure.
c. Support integration of real-time analytics and AI inference at both edge and cloud levels, enabling low-latency decision-making and immediate response during emergency situations (e.g. to enable instantaneous anomaly detection, predictive insights, and decision-making).

2. Network Layer:

a. Dynamic QoS allocation with prioritization for emergency traffic.
b. Integration with fault-tolerant re-configuration and self-healing mechanisms.

b. Performance requirements

- Increase edge computational resource utilization by 40%, while ensuring a balanced distribution of compute workloads across edge and cloud infrastructures, enabled through AI-driven orchestration and dynamic workload placement.
- Increase usage of decentralized renewable energy sources by up to 20% (e.g., solar, wind, battery systems), enabled through distributed, AI-driven energy optimization and coordination with local microgrid operations — thereby enhancing system resilience and service continuity during main grid outages.
- Save up to 25% of energy costs by reducing CNF (Cloud-native Network Function) energy consumption through distributed AI decisions (e.g., model compression, edge offloading, adaptive sleep modes) within the edge-cloud continuum.

7.1.4 Use case specific figures

Below is the diagram showcasing the use case of sustainable networks for disaster handling based on UNITY-6G's native distributed AI-enabled architecture.

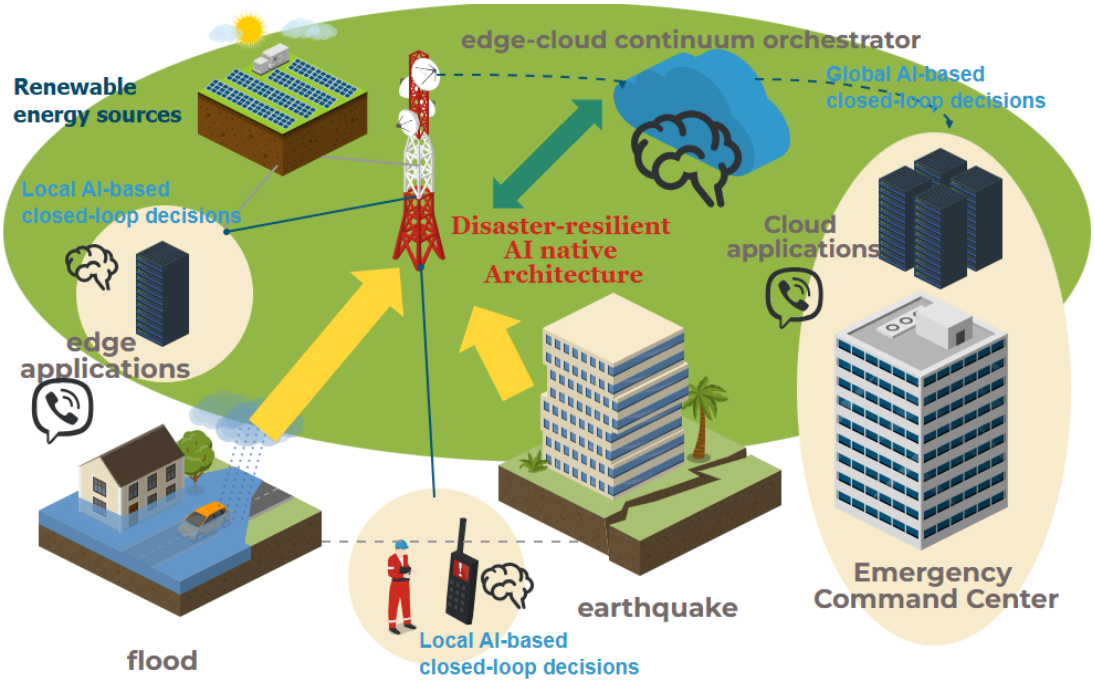


Figure 1: Use case of sustainable networks for disaster handling

7.2 Use case 002: AI Inference as a Service in AI native telecommunication networks

Inference of AI models will enable diverse applications from industry to personal life. In this use case, we introduce the concept of using the AI-native telecommunication network and its environment as a platform for collaborative and efficient AI inference.

7.2.1 Use case summary

Use case id	FG-AINN-usecase-002
Use case name	AI Inference as a Service in AI native telecommunication networks
Created by	Huawei Technologies
Creation date	April, 2025
Use case context	Submitted to the 2nd F2F meeting (April 2025) and approved in the 3rd F2F meeting (June 2025).
Use case description	The rapid diffusion of AI in our society is allowing the development of a large number of new applications, for e.g., individuals, industry, and public organization. In our vision, AI-native telecommunication networks, empowered by intelligent decision-making capabilities, will serve as a platform for delivering AI-based services. In this use case, AI-native telecommunication networks can provide AI inference as a service in a collaborative and efficient manner. AI inference as a service is a family of use cases for both AI for Network and Network for AI. In the AI for Network use cases, an AI/ML model is used to support the network functions of the AI-native telecommunication network. In the Network for AI use cases, the network monetizes its available resources and related capabilities (i.e., data collection and storage) offering new value-added services to a consumer through a dedicated interface.

Core concepts / characteristics of AINN (Optional)	• AI-native telecommunication network • AI-native telecommunication network environment • AI-native telecommunication network value-added services • AI agent in AI-native telecommunication network
Actors and Roles (Optional)	• AI/ML model inference consumer: the consumer of the AI/ML inference, e.g., a NF, an application (server), • AI/ML model repository: the repository where the AI/ML models are stored and can be collected for inference • AI/ML model data producer: the producer of the data that is used during the ML model lifecycle • AI/ML model data repository: the repository where the data used during the ML model lifecycle is stored • AI/ML model orchestrator: NF able to monitor AI/ML resource usage and availability in the AI-native telecommunication network and to take decisions accordingly on the AI/ML model inference process • AI/ML model inference participant: NF able to execute the AI inference process alone or in cooperation with other AI/ML model inference participants.
Technical challenges or Gaps (Optional)	• Current telecommunication networks support inference in AI for Network case (e.g., AI to estimate or forecast quality of experience, network load) but provide limited support of inference in Network for AI case, which can be actually done only at UE and application sides • If AI-native telecommunication network offers AI inference as a service there is the need to consider jointly the AI inference services associated requirements together with the other (e.g., connectivity) services offered by the network and optimize resource allocation accordingly
Use case category	AI training/inference as a service
Reference	[3GPP TR 22.850], [ETSI GR ENI 004], [ITU-T Y.3172], [ITU-T TR-GenAI-Telecom Networks], [3GPP TS 28.105], [b-ISO/IEC 2382-28]

7.2.2 Assumptions

- The AI-native telecommunication network and its environment have capabilities to support (at least partially) ML model lifecycle
- The AI-native telecommunication network can use both user data and network-controlled information in the inference of AI/ML models to support new value-added services

7.2.3 Use case requirements

a. Functional requirements

- It is of critical value that AI-native telecommunication networks support the complete ML model lifecycle as well as the joint optimization of connection, data collection and processing, algorithm, and models in the AI-native telecommunication network and its environment for providing AI inference as a service for both Network for AI and AI for Network;
    - It is of critical value that AI-native telecommunication networks are able to monitor the status of AI resources;
    - It is of critical value that AI-native telecommunication networks are able to monitor the AI service performance requirements, such as AI model accuracy and error, AI inference latency;
    - It is expected that AI-native telecommunication networks have intelligent mechanism to prioritize requirements between AI inference services and other services offered by the network, considering jointly communication and AI resources;
    - It is expected that AI-native telecommunication networks have intelligent mechanism for determining the AI model (and its configuration) to use for a requested AI inference as a service offered by the AI-native telecommunication network as a value-added service. Such determination takes into consideration the status of communication and AI resources;
    - It is of added value that AI-native telecommunication networks have intelligent mechanism to update the AI model (and/or its configuration) for a requested AI inference as a service offered by the AI-native telecommunication network as a value-added service, when the communication and AI resources or the service requirements change;
- It is expected that AI-native telecommunication networks have intelligent mechanism for dynamically splitting the AI Inference workload between the AI-native telecommunication network and its environment;
- It is of added value that AI-native telecommunication networks have intelligent mechanism to transfer additional layers of the AI model to the inference server to increase the model performance upon on changes on the network conditions and AI service requirements [ADG2023];
- It is expected that AI-native telecommunication networks have intelligent mechanism for monitoring and optimizing the AI inference workload and its split within the AI-native telecommunication network and its environment;
- It is of critical value that AI-native telecommunication networks support data lifecycle management of the entire data process for AI inference, including data collection, transmission, processing, storage and consumption;
    - It is of added value that AI-native telecommunication networks support data collection for AI inference with different time granularities in terms of resources, networks and services;
    - It is of added value that AI-native telecommunication networks support data processing for AI, including data processing to enhance data format, quality, privacy and security;
    - It is of critical value that functions for obtaining, storing and processing private user data are defined in accordance with enforced regulations [ITU-T TR-GenAI-Telecom Networks].

b. Performance requirements

• Latency: It is of critical value that the latency of the AI inference is limited in order to ensure effective service provisioning.

• Network overhead: It is expected that function for data lifecycle management minimize network overhead by enabling reuse of data exposed to AI and avoid duplicate data collection

• Resource usage efficiency: AI model inference may demand substantial resources. It is expected that AI-native telecommunication networks balance real-time performance with resource usage implementing resource-aware strategies and AI resource adaptations.
• Accuracy and error: It is expected that AI model inference performance should match the performance requirements of the downstream task. NOTE - Different AI inference tasks, such as regression, forecasting, and classification have distinct accuracy and error metrics.

c. Additional requirements

- It is expected that a telecommunication network supporting AI inference as a Service include an AI/ML model data repository, i.e., the repository where the data used during the ML model lifecycle is stored
- It is expected that a telecommunication network supporting AI inference as a Service include an AI/ML model repository, i.e., a repository where the AI/ML models are stored and can be collected for inference

7.2.4 Use case specific figures

Figure 2 shows three examples of AI Inference as a service provided by the AI-Native Telecommunication Network.
Figure 2(a) shows the case where the AI-native telecommunication network handles the whole AI inference workload. In this case, user data may need to be pre-processed to facilitate (e.g. reduce overhead and latency) its transmission and deal with privacy and security concerns associated with data sharing. The AI model inference may be split across the AI-Native Telecommunication Network, by AI ML orchestrator as illustrated in Figure 2, to take advantages of the diverse conditions (e.g., distance from the consumer, resource availability) of different AI/ML model inference participants (i.e., network nodes with AI capabilities). However, it should be noted that the splitting of the AI model inference introduces new challenges in terms of synchronization, security, and communication and latency overhead.
Figure 2(b) shows the case where AI inference is realized by splitting the AI model inference into multiple inference tasks across the AI-Native Telecommunication Network and its environment including devices, UEs, other devices, in and Application Servers). Inferring part of the AI model at the UE side reduces the communication requirements of the AI/ML model inference and enhances the privacy protection since only the output or the AI model at the UE side, i.e., the data features, are transmitted to other side(s), i.e., the network side.

Figure 2(c) illustrates yet another possibility of offloading the AI/ML model inference to the AI-Native Telecommunication Network. In this example, an AI-enabled application requires data from multiples sources (AI model data producers) and may deliver its results to multiple consumers. The AI/ML model inference is supported by different AI/ML model inference participants in the AI-Native Telecommunication Network as well as in its environment, which jointly achieve the processing for the AI-enabled application (e.g., multi-stream, multi-modal fusion applications). The control and fulfilment of the system requirements formed by AI model data producers and AI/ML model inference participants are performed intelligently by the AI-Native Telecommunication Network with the support of the AI/ ML model orchestrator.

Figure 2 represents the case where data, resources and models are controlled by the telecommunication network operator and used for AI for Network and for Network for AI. Additionally, other variants are possible, such as the case where: 1) the application owns the model and the data, and leases from the operator the necessary resources for the AI inference process; 2) the application owns the data and leases from the operator the necessary resources for the AI inference process including the AI model; and 3) the application leases from the operator the necessary resources for the AI inference process including the AI model taking as input network data.

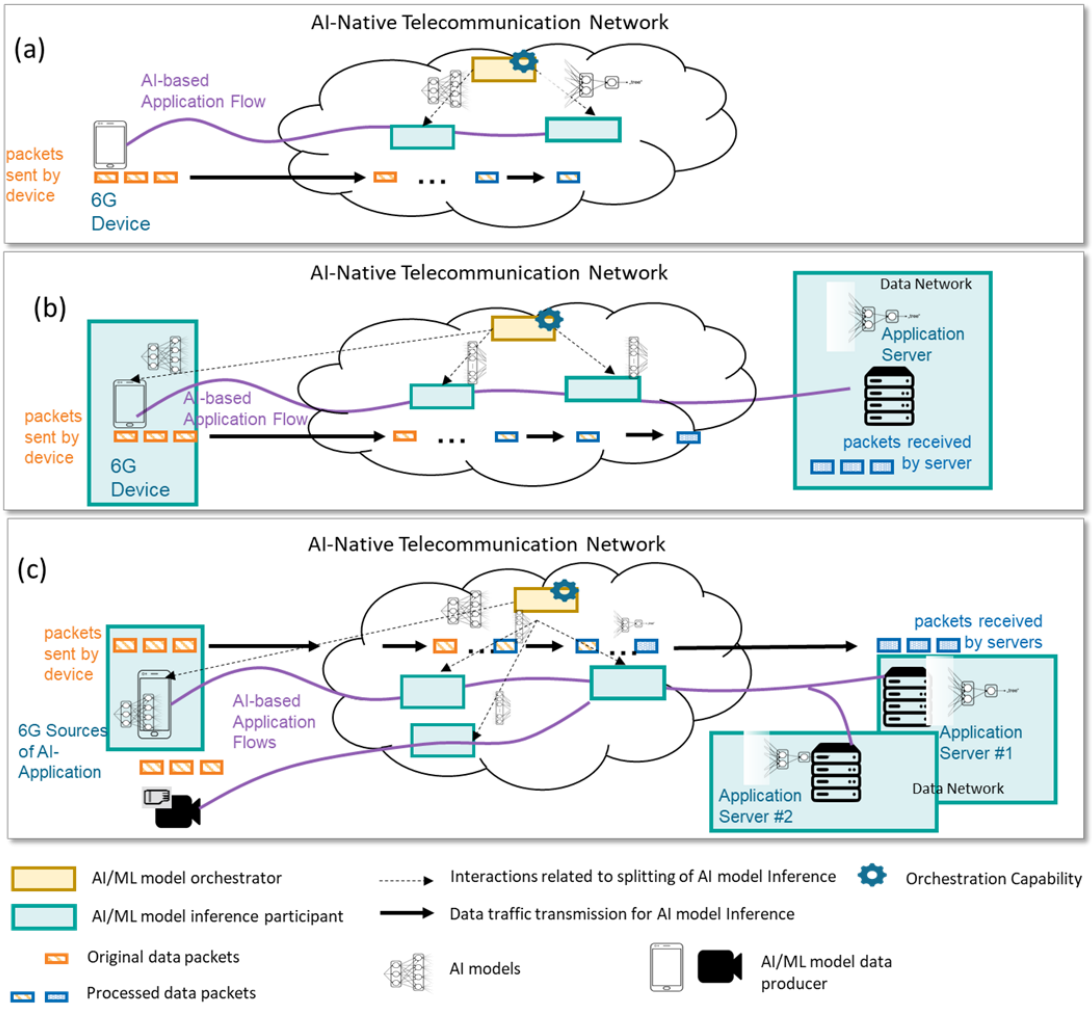


Figure 2 AI inference as a service

7.3 Use case 003: AI Training as a Service in AI native telecommunication networks

Network AI training as a service provides a way to facilitate AI model production with both optimized resource management and guaranteed privacy protection. Datasets for training be built up and used to generate powerful AI models. In this use case, we introduce the concept of using the AI-native telecommunication network and its environment as a platform for collaborative and efficient AI training.

7.3.1 Use case summary

Use case id	FG-AINN-usecase-003
Use case name	AI Training as a Service in AI native telecommunication networks
Created by	Huawei Technologies
Creation date	April, 2025
Use case context	Submitted to the 2nd F2F meeting (April 2025) and approved in the 3rd F2F meeting (June 2025).
Use case description	The data is distributed over the network and collected by sensors, cameras, radars and Lidars on various terminals. Meanwhile, with the rapid development of high-end terminals, fully exploiting this distributed on-device processing capabilities becomes a compelling potential for new application essential requirements.

	The AI-Native Telecommunication Network empowered intelligent decision-making capabilities will serve as a platform for delivering advanced AI-based services. In this use case, AI-native telecommunication networks can provide the AI Training as a service in a collaborative and efficient manner by connecting data silos to enable powerful AI model production. In our vision, AI training as a service is a family of use cases for both AI for Network and Network for AI. In the AI for Network use cases, an AI/ML model is used to support the network functions of the AI-native telecommunication network. In the Network for AI use cases, the network monetizes its available resources and related capabilities (i.e., data collection and storage) offering new value-added services to a consumer through a dedicated interface. Data can be transferred to the network side, where datasets are built up and used to generate powerful AI models in a centralized way. An alternative is training AI models locally instead of uploading the local raw data to the central processing centre. Making choice between the centralized or the distributed way about AI model training depends on several considerations. Firstly, the decision considers the processing capabilities of the involved training entities. Secondly, the privacy protection of the local data should be ensured. Thirdly, the resources for training the AI model should be optimized.
Core concepts / characteristics of AINN (Optional)	• AI-native telecommunication network • AI-native telecommunication network environment • AI-native telecommunication network value-added services • AI agent in AI-native telecommunication network
Actors and Roles (Optional)	• AI/ML model training consumer: the consumer of the AI/ML training service, e.g., a NF, an application (server) • AI/ML model repository: the repository where the AI/ML models are stored after training or update • AI/ML model data producer: the producer of the data used during the ML model lifecycle • AI/ML model data repository: the repository where the data used during the ML model lifecycle is stored • AI/ML model orchestrator: NF able to monitor AI/ML resource usage and availability in the AI-native telecommunication network and to take decisions accordingly on the AI/ML model training process • AI/ML model training participant: entity (e.g., device, NF) able to participate in the AI training process alone or in cooperation with other AI/ML model training participants
Technical challenges or Gaps (Optional)	• Current telecommunication networks support AI for network training (AI to estimate or forecast quality of experience, network load, etc) but provide limited support of network for AI training (this can be done only at UE and application sides)

	• If AI-native telecommunication network offers AI Training as a service there is the need to consider jointly the AI Training services associated requirements together with other (e.g., connectivity) services offered by the network and optimize the resource allocation accordingly
Use case category	AI training/inference as a service
Reference	[ITU-T TR-GenAI-Telecom Networks], [ITU-T Y.3172], [ETSI GR ENI 004], [3GPP TS 28.105], [3GPP TR 22.850], [b-ISO/IEC 2382-28]

7.3.2 Assumptions

- The AI-native telecommunication network and its environment have capabilities to support (at least partially) ML model lifecycle
- The AI-native telecommunication network can use both user data and network-controlled information in the training of AI/ML models to support new value-added services

7.3.3 Use case requirements

a. Functional requirements

- It is of critical value that AI-native telecommunication networks support the ML model lifecycle as well as the joint optimization of connection, data collection and processing, algorithm, and models in the AI-native telecommunication network and its environment for providing AI Training as a service for both Network for AI and AI for Network;
    - It is of critical value that AI-native telecommunication networks are able to monitor the status of AI resource usage;
    - It is of critical value that AI-native telecommunication networks are able to monitor the AI service performance requirements, such as AI model accuracy, AI service latency;
    - It is expected that AI-native telecommunication networks have intelligent mechanism to prioritize requirements between AI Training services and other services offered by the network, considering jointly communication and AI resources;
    - It is expected that AI-native telecommunication networks have intelligent mechanism for determining the AI model (and its configuration) to use for a requested AI training as a service offered by the AI-native telecommunication network as a value-added service. Such determination takes into consideration the status of communication and AI resources;
- It is expected that AI-native telecommunication networks have intelligent mechanism for dynamically split the AI Training workload between the AI-native telecommunication network and its environment, based on resource availabilities and AI Training;
- It is of added value that AI-native telecommunication networks have intelligent AI training mechanism that construct AI models able to work in different configuration to trade-off AI service performance with network resource usage [ADG2023];
- It is expected that AI-native telecommunication networks have intelligent mechanism for monitoring and optimizing the AI Training workload and its split within the AI-native telecommunication network and its environment;
- It is of critical value that AI-native telecommunication networks support data lifecycle management of the entire data process for AI Training, including data collection, transmission, processing, storage and consumption;

- It is of added value that AI-native telecommunication networks support data collection for AI Training with different time granularities in terms of resources, networks and services;
- It is of added value that AI-native telecommunication networks support data processing for AI, including data processing to enhance data format, quality, privacy and security;
- It is of added value that AI-native telecommunication networks support data management for supporting AI Training and Testing/Validation phases;
- It is of critical value that functions for obtaining, storing and processing private user data are defined in accordance with enforced regulations [ITU-T TR-GenAI-Telecom Networks].

b. Performance requirements

- Latency: It is of critical value that the latency on the exchange of the data serving as input for the AI training service, as well as the latency on exchange of information among the AI/ML Training participants is limited in order to ensure effective service provisioning.
- Network overhead: It is expected that data lifecycle management minimizes network overhead by enabling reuse of data exposed to AI and avoid duplicate data collection
- Resource usage efficiency: AI model training may demand substantial resources. It is expected that AI-native telecommunication networks balance timely AI training completion with resource usage implementing resource-aware strategies and AI resource adaptations.
- AI/ML model training performance: It is expected that AI-native telecommunication networks enforce a trade-off to balance the amount of resources used for the AI training versus the target metrics for defining the AI/ML model convergence, and therefore AI training service completion.

c. Additional requirements

- It is expected that AI-native telecommunication networks supporting AI training as a Service include an AI/ML model data repository, i.e., a repository where the data used during the ML model lifecycle is stored
- It is expected that AI-native telecommunication networks supporting AI training as a Service should include an AI/ML model repository, i.e., a repository where the AI/ML models are stored and can be collected for inference

7.3.4 Use case specific figures

As shown in **Figure *3***, based on the topology and location where the learning tasks take place, the AI/ML can be categorized to centralized learning, distributed learning and federated learning [3GPPTS28105].

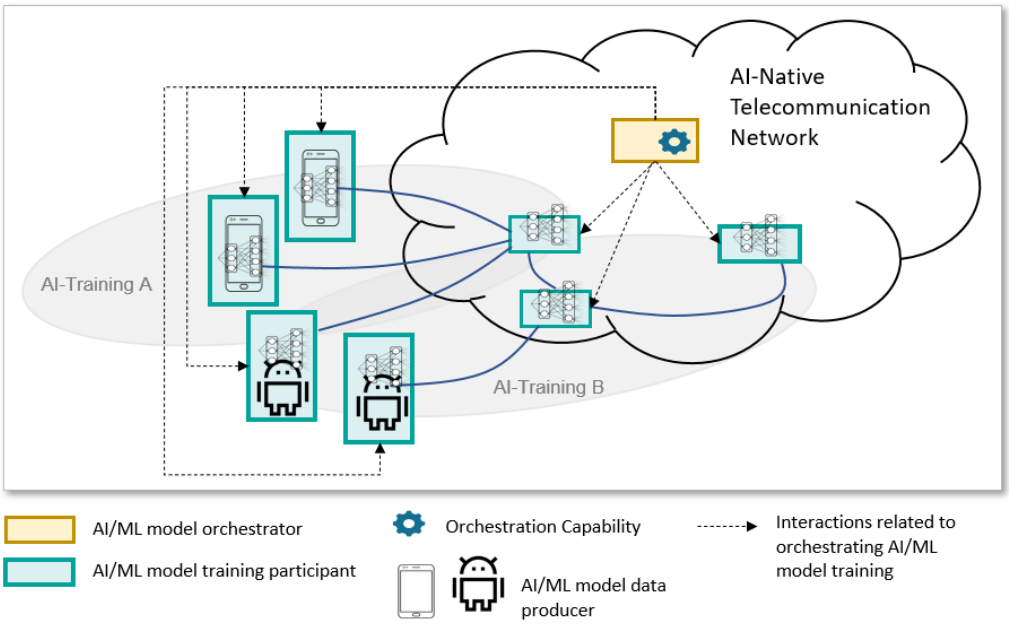


Figure 3 AI training as a service. AI-Training A and AI-Training B represent two instances of the AI Training as a Service.

7.4 Use case 004: LLM-based value-added services in AI native telecommunication networks

LLMs are becoming an important part of evolving systems in software and telecommunication engineering. They can be used to understand structure, interdependencies and complexities of language but also of interactions over a networked system.

7.4.1 Use case summary

Use case id	FG-AINN-usecase-04
Use case name	LLM-based value-added services in AI native telecommunication networks
Created by	Huawei Technologies
Creation date	April, 2025
Use case context	Submitted to the 2nd F2F meeting (April 2025) and approved in the 3rd F2F meeting (June 2025).
Use case description	LLMs are AI models with the ability to process information and demonstrate human-like text generation capabilities. In the context of AI-native telecommunication network, LLMs have three main general capabilities that can be used to offer new value-added services: • Semantic capabilities: LLMs develop an internal representation of textual data in the form of real-valued vectors called embeddings. LLMs can process and comprehend intricate information, such as the content within standard documents and infer logical conclusions from the given inputs • Intelligent Access to Knowledge: by understanding the specific intention conveyed through the prompt, an LLM can effectively apply its knowledge base to craft a response tailored to the user needs.

	• LLMs as Orchestrators: LLMs can utilize their knowledge to deconstruct complex tasks into manageable subtasks and deploy suitable (external) tools for each. In light of these capabilities, the integration of LLMs in AI-native telecommunication network arises as key tool in the context of AI for Network: • LLMs can leverage the large amount of textual data, such as network anomalies tickets, product manuals, and software to support the end-to-end network lifecycle, including requirement understanding, planning, design and simulation, implementation, tuning, evaluation and optimization. In addition, in the context of Network for AI: • The AI-native telecommunication network can leverage communication resources to support coordination between LLM based AI agents, LLM transfer and distributed training. In addition, the AI-native telecommunication network can provide data collection capabilities and resources for LLM training and assessment
Core concepts / characteristics of AINN (Optional)	• AI-native telecommunication network • AI-native telecommunication network environment • AI-native telecommunication network value-added services • AI agent in AI-native telecommunication network
Actors and Roles (Optional)	• AI/ML model repository: the server where the LLM are stored and can be collected for inference or model update • AI/ML model data repository: the repository where the data used during the ML model lifecycle is stored • AI/ML model orchestrator: NF able to monitor LLM resource usage (for inference and training) and availability in the AI-native telecommunication network and to take decisions accordingly on the LLM model inference process • Knowledge base: repository containing various types of data such as documents, databases, or search engines used by LLMs to retrieve relevant data during inference. • AI/ML model certifier: server running benchmarks to test LLM capabilities • AI agent; see clause 1.1.6
Technical challenges or Gaps (Optional)	LLM training and usage in telecommunication networks pose a number of challenges due to the size of the models and the amount of required data for training LLM based AI agents will require dedicated communication protocols and interfaces to support multi-agent operations [ETSI GR ENI 051] as well as the access to external tools [MCP] Given the large number of models existing in the literature (and their different requirements and capabilities), one challenge for the AI-native telecommunication network is to evaluate which LLMs satisfy the service specific requirements and how to select the most appropriate LLM that strikes a balance between service performance and network resource usage LLM based AI agents operating across diverse tasks or cooperating on the same task require coordination on resource usage and in the decision-making process

Use case category	LLM-based solutions
Reference	[ITU-T TR-GenAI-Telecom Networks], [ITU-T Y.3172], [ETSI GR ENI 004], [3GPP TS 28.105], [3GPP TR 22.850], [ETSI GR ENI 051], [b-ISO/IEC 2382-28]

7.4.2 Assumptions

- The AI-native telecommunication network and its environment have capabilities to support (at least partially) ML model lifecycle
- The AI-native telecommunication network can use both user data and network-controlled information in the inference of AI/ML models to support new value-added services
- The AI-native telecommunication network has access in real time to external tools to support the LLM based AI agent functionalities

7.4.3 Use case requirements

a. Functional requirements

Part of the following requirements are based on the ones reported in [ITU-T TR-GenAI-Telecom Networks]. They can be divided in AI-native telecommunication network functional requirements and LLM functional requirements.

AI-native telecommunication network functional requirements:

- It is of critical value that AI-native telecommunication networks are able to monitor the status of LLM resource usage;
- It is of critical value that AI-native telecommunication networks are able to monitor the LLM service performance requirements, such as LLM model accuracy, LLM service latency and LLM service density (i.e., the number of LLM based AI agents in a given area to offer the ongoing value-added services)
- It is expected that AI-native telecommunication networks have intelligent mechanism to prioritize requirements between LLM-based services and other services offered by the network, considering jointly communication and AI resources
- It is expected that AI-native telecommunication networks have intelligent mechanisms for determining the LLM to use for a requested service offered by the AI-native telecommunication network as a value-added service. Such determination takes into consideration the status of communication and AI resources, as well as the LLM performance on the use case requirements (i.e., LLM benchmarking).
- It is expected that AI-native telecommunication networks have intelligent mechanisms for monitoring and optimizing the LLM workload within the AI-native telecommunication network and its environment;
- It is expected that AI-native telecommunication networks expose to LLMs business intents, network operator's policies, resource availabilities, user QoS and network KPIs;
- It is expected that AI-native telecommunication networks expose to LLMs the available knowledge base
- It is expected that AI-native telecommunication networks support data lifecycle management of the entire data process for LLMs, including data collection, transmission, processing, storage and consumption
    - It is of added value that AI-native telecommunication networks support data collection for LLM training and inference with different time granularities in terms of resources, networks and services;
    - It is of added value that AI-native telecommunication networks support data processing for LLMs, including data processing to enhance data format, quality, privacy and security

- It is of critical value that functions for obtaining, storing and processing private user data are defined in accordance with enforced regulations

LLM functional requirements:

- It is of added value that LLMs are capable of software development
- It is expected that LLMs comply with the expected format of their output data
- It is expected that LLMs support diverse data structures as input
- It is expected that LLMs are capable of mathematical and logical reasoning
- It is of added value that LLMs support tool calling. NOTE: Tool calling refers to the ability of the LLM to interact with external tools (e.g., code interpreter and web search engines), APIs or systems to enhance their functions.

b. Performance requirements

- Network overhead: It is expected that data lifecycle management minimizes network overhead by enabling reuse of data for LLMs and avoid duplicate data collection
- Resource usage efficiency: LLM demand substantial resources, especially during inference, which occurs continuously in telecom applications like customer service. It is expected that AI-native telecommunication networks balance timely AI training completion with resource usage implementing resource-aware strategies and LLM resource adaptations
- Latency: It is of critical value that the inference latency of the LLMs is kept under control to ensure effective service provisioning.
- Random-Access Memory (RAM): It is expected that RAM usage efficiency is large in order to comply with hardware constraints and processing efficiency. It is of added value that the amount of data stored and the associated embeddings (e.g., to support RAG) should be continuously adapted and optimized.

c. Additional requirements

- It is expected that AI-native telecommunication networks using LLMs for supporting new added-value services include an AI/ML model data repository, i.e., a repository where the data used during the ML model lifecycle is stored
- It is expected that AI-native telecommunication networks using LLMs for supporting new added-value services include an AI/ML model repository, i.e., a repository where the AI/ML models are stored and can be collected for inference
- It is of added value that AI-native telecommunication networks using LLMs for supporting new added-value services include a knowledge base, i.e., a repository containing various types of data such as documents, databases, or search engines used by LLMs to retrieve relevant data during inference

7.4.4 Use case specific figures

The top part of **Error! Reference source not found.** shows an LLM based intent-driven network. The b ottom part of Figure 4 shows network policy evaluation through LLM based AI agents.

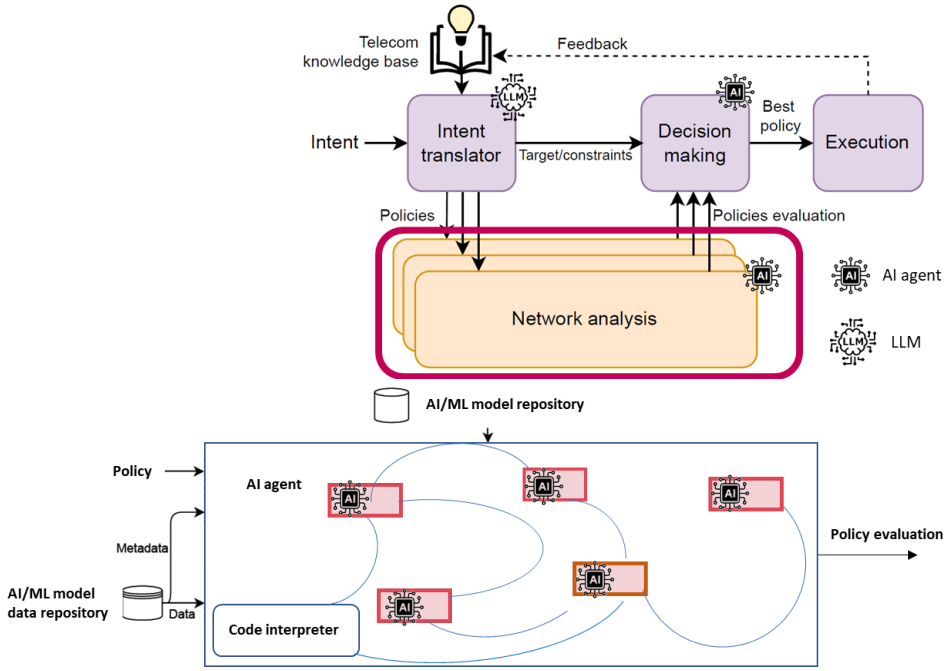


Figure 4. LLM based intent-driven network (top); AI agents for network policy evaluation (down).

7.5 Use Case 005: AI-native Networks as Applied in Verticals

This use case offers a comprehensive view of the capabilities and benefits of AI-native networks in domains that align with their architectural traits. Specifically, it delves into the detailed performance analysis of AI-native networks as applied in verticals. Then corresponding requirements for AI-native networks applied to verticals.

7.5.1 Use case summary

Use case id	FG-AINN-usecase-005
Use case name	AI-native Networks as Applied in Verticals
Created by	State Grid Corporation of China （SGCC）
Creation date	9 April 2025
Use case context	Submitted to the 2nd F2F meeting, approved in the 15th WG2 meeting (2025.08.21)
Use case description	**1. Background and Necessity** With the progress of technology, AI-native network has its advantage which support it to become one of the technical methods in verticals. And it is mainly based on the similarities between the operational characteristics of the AI-native network and the operational requirements of verticals. At present, the technology of AI-native networks has not yet had a complete supporting architecture in verticals. Therefore, it is necessary to analyse the common characteristics between the operational characteristics of the AI-native network and the operational requirements of verticals, and to analyse the technical characteristics that may be applied in verticals.

	2.Similarities between the operational characteristics of the AI-native network and the operational requirements of the verticals 2.1 Machine learning: AI-native networks have flexible AI deployment models, thus offering greater operability in machine learning and training, and can be used for targeted functional implementation. This is very similar to the local technical operation requirement patterns in many verticals. Such as the terminal independent grids (such as microgrid or distributed grid) in the power grid which have high degree of security and privacy requirements, as well as the processing requirements of monitoring data for specific locations (such as toll stations of the highway or important traffic roundabout islands) in the transportation system, etc. In these cases, different data often belongs to different operators, and the ability of data sharing is limited to some extent. 2.2 Database connection/transmission: AI-native network can support diversified data integration through the wide range of data links, to provide more powerful fundamental support capabilities to meet the machine learning and data processing requirements. This is similar to the requirement of operations such as data transmission and sharing between units with different characteristics in the verticals. 2.3 Prediction: AI-native network uses the special model such as natural language processing (NLP) and large language model (LLM) to predict demand, so that the original prediction analysis ability is strengthened. It has a high similarity with the large number of predicted demands in the verticals (such as load forecasting and fault warning in the power grid, the flight delays warning caused by weather conditions in the aviation, and the prediction of the growth of purchasing power due to special reasons in market transactions, etc.) 2.4 Knowledge base: The database in AI-native network can contain different types of operating data in multiple scenarios, and has the ability to update dynamically. This is similar to the large demand for influential knowledge in related other fields in the vertical. 2.5 Application: The AI-native network has a strong ability to deal with complex and changing needs, such as the rapid identification and classification of terminal data usage requirements, and different demands are reflected in the settings of the application, thereby meeting different user experiences. This is similar to the complex demand scenario on the user side in verticals. **3. The main optimization effects based on AI-native networks** The main optimization effects based on AI-native networks are reflected in the following aspects of network operation: a) feature specific application effects of the network, b) enhancements to the network, and c) new AI models and solutions. 3.1 Feature specific application According to the different interaction modes/data types between verticals and networks (telco network/local AI network, etc.), we can explore and discuss the specific ways to use AI-native networks as specific application to optimize the original network functions. 3.1.1 Operating data: This type of data is used to reflect the effect of the physical layer operation in verticals. Generally, it is presented in the form of data through specific communication methods to show the overall operation of the verticals, such demands are included in

	verticals that require specific communication technologies for data and information processing, such as power grid/heating/water supply. The major specific applications of AI-native networks in these verticals are monitoring/telemetry, etc. 3.1.2 Transmitting data: This type of data is mainly used to reflect the response rate of data instructions and the ability to deal with special events in verticals. The major specific applications of AI-native networks are to increase data transfer/processing speed and remote-control capabilities, and optimize the resource allocation or data transmission. Typical examples include the speed of control instructions for traffic lights and remote manual control when dealing with scenarios with heavy traffic, etc. 3.1.3 Analysis/prediction data: This type of data is mainly used for the analysis and verification of unexpected events or short/long-term operational trends in the future. The data, e.g., predictions of natural environmental factors such as weather, temperature, tides, and water flow, as well as predictions of load offsets caused by human activities in fields such as the power grid and transportation, may not in and of themselves represent the actual state of operation of the vertical, but are needed to serve as the basis for of data analysis/ AI function development. The specific applications of AI-native network are the storage of analysis data based on dynamic knowledge base, and the context-aware model training as the premise for prediction. 3.1.4 Interactive data: This type of data is used to balance the operational effectiveness between telco networks and verticals. It mainly serves as the basis for determining the service efficiency of telco networks. At the same time, it can also be used to evaluate the operational efficiency of telco networks under the influence of the characteristics of verticals and provide feedback on the demands of telco networks for verticals. The specific application of AI-native networks in this section is to optimize the service effect of the network, the most important of which is to optimize the integration of federated and distributed AI, and the AI pipeline/controller for the needs of the verticals. 3.2 Enhancements to the network In terms of network enhancement, the integration of AI functions in the network and the strengthened AI technology are used to predict and output. The role of the vertical domain here is mainly manifested as the basic reference of AI integration according to the needs and operating characteristics of verticals, or the specific AI pipeline/control methods are learned from verticals. However, the enhancement of this part is mainly the enhancement of AI-native to the original network operation, and the operation of verticals is only used as the foundation or goal. The role of this part can be achieved through the specific application of AI-native network in the previous section. 3.3 New AI models and solutions New AI models and solutions are generated and integrated in the network according to the technical requirements of the vertical. New AI models and solutions are used for specific needs of specific requirements of verticals.

	4. Specific technical requirements of AI-native network applied to verticals As can be seen from the previous analysis, the technical requirements of AI-native network applied to the verticals mainly depend on the feature specific application part of the AI-native network, which includes the following: a) cross domain requirements for AI model training and data sharing; b) the integration of AI within the network according to the technical characteristics and requirements of verticals; c) the configuration of AI pipelines. 4.1 AI model training Cross domain AI training and machine learning are the basic requirements for the application of AINN. As mentioned previously, through AI training/machine learning, we can provide basic support for data functions such as operation/analysis/prediction/interaction and form a complete network for verticals. 4.2 Integration of AI within the network A large part of the feature specific applications of AI- native network will be realized through enhanced AI integration in the network, and this part is even more important in the related applications of verticals. Depending on the needs of different verticals, AI integration in networks can include federated and distributed forms of AI deployment, as well as applications for prediction and analysis at key network nodes. Here we give a brief description with the previous classification. 4.2.1 Integration of distributed AI prediction capabilities at the terminal Among the existing technologies, in verticals, the support functions of AI for the network are mostly specific analysis and offline models, etc., and the interaction with the network on the terminal side is generally less. AI-native network combines the federated and distributed deployment mode, which can integrate AI into the terminal side of the network, to integrate the feedback of the operation effect and the local prediction, and more effectively support the operation of the network. 4.2.2 Analysis and transmission of AI integration support at key nodes An important function of AI integration is to quickly and efficiently predict and allocate resources for network operation, to improve the accuracy of data analysis and the efficiency of data transmission. Most network which supports verticals will contain the main data integration or data processing key nodes, which are also the key technical nodes to deeply integrate AI into the network. AI functions that can classify/fetch/pre-process/ allocate data are deployed at key nodes to undertake the tasks of resource allocation and transmission efficiency improvement, to support the deep integration of all aspects of the network and the optimization of analysis functions. 4.2.3 LLM Analysis with Knowledge Base The final output of the network in the verticals is mainly the analysis and prediction of the overall operation effect, which is also the role of the operation data mentioned in the previous section. With the AI integration in depth, the dynamic knowledge base and LLM analysis functions of AI-native networks can output analysis results that are

	more easily recognized and accepted by humans, thus reflecting the operation of verticals. 4.3 The configuration of AI pipelines The setup of AI pipeline is an important factor that affects the use cycle and the effect of AI control. When AI-native network is used in verticals, the specific functions and technical requirements of the AI pipeline need to learn from the operation characteristics of verticals, and the AI pipeline with specific direction should be configured. However, this part will be applied differently in different verticals, but all of them need a strong learning ability and training mechanism as a foundation. It can be seen from the above content that it is of certain application value to use AI-native network as a main technology to support the verticals.
Core concepts / characteristics of AINN (Optional)	·More flexible machine learning deployment methods: Provides targeted deployment for machine learning. ·Integrating information in knowledge bases: The main sources of analysis/prediction data. ·Data transfer and linking from multiple sources: enabling wide-area AI learning and achieve free transmission of operational data. ·Deep integration of AI features in networks: feedback the running status and needs of verticals in time through deep integration of AI features, and optimize the operation effect of the network. ·Resource allocation: The optimal resource allocation is achieved by deep integration of AI features at different locations in the network, and the specific needs of verticals are combined with the real-time interaction with the dynamic knowledge base to further optimize the resource allocation. ·AI pipelines: AI pipeline setup needs to be carried out with the specific application based on technical requirements of verticals.
Actors and Roles (Optional)	·As a feedback to operating effect of verticals ·As a content reference for the interaction between telco network and verticals ·As a basis for dynamic machine learning/AI training ·As an optimization strategy for resource allocation in verticals ·As a basis for AI pipelines
Technical challenges or Gaps (Optional)	•Make sure the data transmission is safe and complete to handle possible communication network breakdowns when there are faults in the verticals. •Have the ability to effectively combine various units or share data through the backend of the AI-native network whenever possible. •Give immediate responses to the complex and changing demands of users.
Use case category	AI-native networks applied in vertical industries

7.5.2 Environment assumptions

• The access forms of various units of the verticals can guarantee the data transmission capacity, and there is no obvious barrier to data exchange between different operating entities.
• AI-native networks are established on the premise of complying with various security regulations.

7.5.3 Requirements

7.5.3.1 Expected goals for the overall operation effect of the network

This part mainly corresponds to section 2.1.2 to 2.1.4 in the "Use case description" mentioned in chapter 2, with the aim of enhancing and optimizing network performance in verticals and the corresponding functions applied to data.

• Fully consider the applications in the network, including specific manifestations and requirements, and deeply integrate AI functions.
• Overall quality improvement of operational data.
• Deployment of distributed AI capabilities which be used to optimize the network
• Provide application-based resource allocation to improve network support capabilities
• Specific AI pipeline setup

7.5.3.2 Specific technical requirements applied in verticals

This part mainly corresponds to section 2.1.1 and a portion of section 2.1.3 and 2.1.4 in the "Use case description" mentioned in chapter 2. The requirements proposed here mainly refer to the demands that AINN should meet when aiming to enhance the operational efficiency of verticals by using its deep integration and optimization capabilities.

7.5.3.2.1 Integration of machine learning capabilities

AI-native network needs to have the ability that can transform machine learning into a supportive scenario analysis that covers the entire network or a directly related scope within the network by leveraging the integration capabilities of AI, thus replacing the work on offline data. Meanwhile, the deep integration of AI must meet the requirement of saving the time spent on analysing data for specific scenarios.

7.5.3.2.2 Data transmission capabilities

AI-native network need have compatibility enabling the AI functions to be integrated with data processing of multiple computation models/frameworks, and possess capabilities such as rapid responding/low latency /high prediction accuracy.
Note: This part has been specifically described in [FG-AINN-I-146]. NOTE-The data referred to here is that used to demonstrate the operational effect of verticals.

7.5.3.2.3 Safety

Operation safety:
a) Safety monitoring: AI-native network needs to have complete and efficient data learning and analysis capabilities, and also be able to respond immediately to ensure that latency is minimized. In this regard, AINN still needs to have the ability to quickly analysis applied the overall network operation effect, so as to identify and respond to fault information in the first instance.
b) Disaster Handling: When some faults that can be called disasters happen, the main requirements of AI-native network are to quickly restore the network functions in the disaster handling.
Note: This part has been specifically described in [FG-AINN-I-WG2-006-R3].
Data safety:
a) Compatibility: To solve the problem of incompatibility that exists among the operation units in verticals, AI-native network needs to have more flexible distributed integration capabilities and a more open data management model, so as to support local processing of data from different operation modes.
b) Privacy: To solve the problem of the latency in data transmission and storage caused by data encryption mechanisms, AI-native network needs to have the ability that can perform local data processing on terminals with high privacy requirements.

7.5.3.2.4 Integrated capabilities

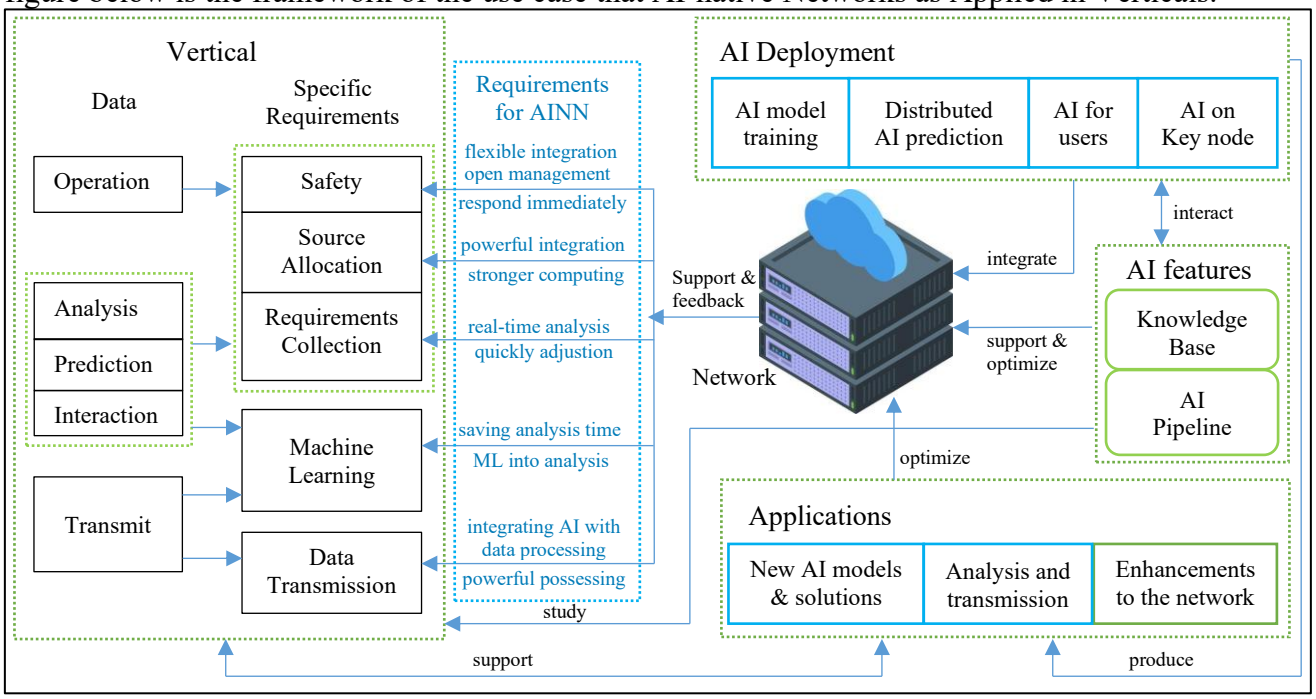
AI-native network needs to have powerful integration and computing capabilities to support large data models, and has stronger computing and learning capabilities at the key nodes of the grid in verticals. So as to integrate resource information and guide resource allocation.

7.5.3.2.5 User requirements collection

AI-native network needs to have the ability to integrate real-time analysis results of AI and quickly adjust the functions of applications within the network. This enables users' requirements for applications to be integrated into the network.

7.5.4 Framework of Use case

The figure below is the framework of the use case that AI-native Networks as Applied in Verticals.



7.5.5 Example (for power grid)

7.5.5.1 Scope

AI native network is adopted and applied to network technology within the power grid.

7.5.5.2 Object of service

Considering the operation requirements of power grid, the functions that can be used for network services and interactions with AI-native networks include the following.

•Operation data monitoring: optimize the monitoring effect of different factors.

•Grid information monitoring: holistic monitoring that integrates more access data.

•Telemetry and remote control: higher telemetry requirements.

• Terminal grid: integrating the information interaction of different operators.

• Users: determine the optimal operation mode in a timely manner and select the best operator/application.

7.5.5.3 Requirements/expected goals

7.5.5.3.1 Integrated machine learning at the terminal

AI-native network has a more flexible machine learning mechanism and strong local AI learning ability in terms of artificial intelligence machine learning. Such a mode can meet the strong privacy data protection of the terminal side, and reduce the workload of the data centre. Due to the diversity of data forms of various participating units in the power grid, the information owners come from multiple different operators, and the basic sources of machine learning are scattered. Therefore, it becomes easier to adopt local AI machine learning mechanisms to form decentralized AI learning mechanisms and integrate their respective data learning capabilities. This has a positive effect on the support for operating data and transmitting data.

7.5.5.3.2 Data monitoring based on dynamic knowledge base

Based on the functionalities of the knowledge base of the AI-native network, more comprehensive feature feedback can be achieved in the monitoring process such as the feedback from weather and human activities the monitoring mode of the traditional monitoring system with a single operation data will change. It can make a positive effect on the quality of operating data and the accuracy of prediction data.

7.5.5.3.3 Telemetry capability improvement

Based on the ability of the AI-Native network that can deploy AI capabilities at key nodes in the power grid to enhance the transmission capacity of the data link by integrating AI. At the same time, by the distributed integration of AI in the network, the telemetry function of data siloes and independent systems, such as microgrids and some independent mini-grids, is achieved. Thus, the interaction ability between the power grid and the telecom network can be improved.

7.5.5.3.4 Remote control prediction function

The strong network prediction ability of AI-native network is used to realize accurate intelligent remote control during the operation of power grid, and reduce unnecessary operations caused by data analysis and discrimination. It will make a positive significance for improving the availability of transmitting data

7.5.5.3.5 Establishment of targeted service mechanism

The prediction ability of AI-native network and the rapid analysis ability of terminal data usage demand are used to establish a targeted service mechanism based on changing user needs. It not only helps to improve the quality of operating data, but also provides direct feedback on the operational requirements of the telco network used to serve the power grid.

7.5.5.3.6 Resource allocation optimization

With the support of feature specific application capabilities, the AI native network is used to optimize the resource allocation ability and the cooperation effect of different units in the power grid. It can break the traditional mode of running through only one line from beginning to end.

7.5.5.3.7 AI pipeline setup

The AI pipeline that can bring efficient services is implemented by integrating the AI functions of various nodes in the network within the power grid, and is used for grid operation.

7.5.5.4 Specific use case requirements from power grid

7.5.5.4.1 Requirement from operational efficiency of power grid

It is critical to integrate all technical parameters related to power grid operations, forming a distinctive operational analysis framework for the power grid. This allows the optimization of any grid stage and units to serve as a way to optimize the overall network architecture's operational effectiveness. At the same time, the optimized resource allocation ability is used to reduce the process and time of analysing power grid resources, and the operation efficiency of the network is further improved.
Note: Traditionally, the power grid's operational mode facilitates interaction and energy transfer among these stages through physical connections, necessitating substantial human and material resources to uphold its safe and efficient functioning.

7.5.5.4.2 Requirement from safety

It is critical that high level of safety is assured in the power grid. By adopting distributed AI functions integrated at the terminal, it effectively avoids the problem of security mismatch between different units, also protects data privacy better.

7.5.5.4.3 Requirement from generation side

It is critical that monitoring of meteorological parameters and precise analysis of power generation facility performance are established. The power generation side mainly relies on clean energy sources/natural resources such as wind, solar, and hydropower. It is important to track the sustainable

energy potential of natural resources like wind, solar, and hydropower, then optimize their energy consumption efficiency and specific power distribution. Therefore, by adopting a dynamic knowledge base to provide basic data and leveraging optimized resource allocation and data analysis capabilities as network supportive technologies, the optimizing of power generation capacity can be well achieved.

7.5.5.4.4 Requirement from transmission side

a) It is critical that energy utilization efficiency is predicted for the upcoming specific duration ahead. Through the deep deployment of AI, especially at the key nodes of the power grid, the prediction results for the future operation of the power grid can be optimized.
b) It is critical to monitor the health status and optimize the control room's operational effectiveness for the grid's overall operational capacity. Using sensing technology and the power lines' operational state as the network's basing architecture, an energy flow display (similar to the traffic network display) is visualized, seamlessly integrating the whole power network.

7.5.5.4.5 Requirement from substation side

It is critical to determine the fault content, that the whole network's entire data is analysed and predicted, quickly identifying the impact of localized faults on overall operations. The main risk that the power grid needs to face is the inability to quickly identify and analyse the complex and changeable sources of hidden dangers. Deeply deploying AI functions within the network can significantly optimize the identification and analysis of sources of hidden dangers related to power grid operation, thereby solving this problem. Note: Substation operations face numerous security risks, including smoke, fire, and foreign object intrusion, rapidly analysing how to achieve the objective that reduce the loss to a minimum, and then improve the entire power grid's operational effectiveness are important.

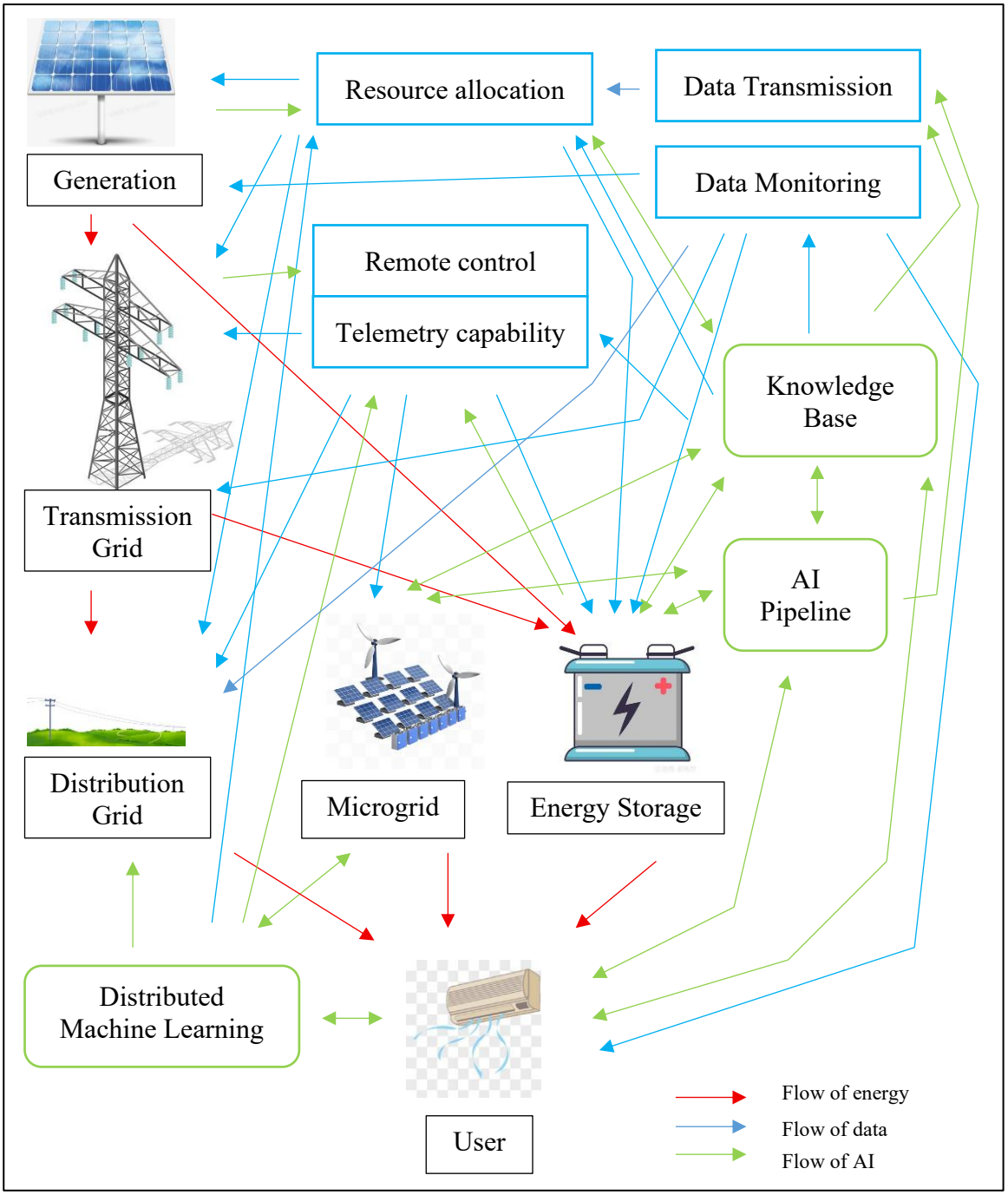
7.5.5.4.6 Requirement from distribution side

It is critical that unified control and optimization is extended to microgrids (at the distribution side primarily pertains to terminal power grids) as part of the overall power grid's operations, though they operate as independent terminal power grids.

7.5.5.4.7 Requirement from consumption side

It is expected that the power grid analyses the impact of both electricity consumption and transmission behaviours at the consumption side (user), so we can optimize power users' operational effectiveness. This is because users of electricity have gradually shifted from a unidirectional mode of obtaining electrical energy to interacting with the grid.

7.5.5.5 Framework of use case



7.5.6 Bibliography

[FG-AINN-O-08] FG-AINN-O-08, Study of concepts, characteristics and definitions of artificial intelligence native telecommunication networks
[FG-AINN-I-152] FG-AINN-I-152, Updated version of "Standardization Gap Analysis of the FG-AINN"
[FG-AINN-I-104] FG-AINN-I-104, AI-Native Proactive Network Slice Marketplace for Dynamic Service Ecosystems.
[FG-AINN-I-142] FG-AINN-I-142 Draft Technical Specification on vocabulary for Artificial Intelligence Native for Telecommunication Networks.
[FG-AINN-I-146] FG-AINN-I-146, AI-driven network performance modelling and prediction.
[FG-AINN-I-WG2-006-R3] FG-AINN-I-WG2-006-R3, Sustainable Networks for Disaster Handling.

[FG-AINN-I-WG2-015] FG-AINN-I-WG2-015, Use case – Creation of data for machine learning in AI-native networks.

7.6 Use case 006: Use case on communication and coordination for multiple AI Agents

The communication pattern of AI agents is significantly different from interaction patterns between humans and objects. Traditional network lacks the capability to interpret high-level and multi-modal instructions, which is unable to coordinate tasks of AI agents across resource-constrained devices. It is necessary for AI-native networks to address these limitations by integrating AI technology into the network architecture, enabling intent recognition, task decomposition, multi-modal transformation, and adaptive resource orchestration to optimize performance and efficiency of networks.

7.6.1 Use case summary

Use case id	FG-AINN-usecase-006
Use case name	Use case on communication and coordination for multiple AI Agents
Created by	China Mobile
Creation date	2025.4.1
Use case context	Submitted to the FG-AINN 2nd meeting, approved in the 15th WG2 meeting (2025.08.21)
Use case description	AI agents have long been recognized as a promising approach to achieving artificial general intelligence (AGI), which is expected to accomplish tasks through autonomous planning and actions. The various forms of AI agents include embodied intelligent robots, virtual assistants, and autonomous systems (e.g., autonomous drones with built-in AI agents). It is necessary for network to support seamless collaboration among multiple AI agents. However, traditional networks fail to meet the demands of AI agents due to their inability to interpret high-level and multi-modal instructions (e.g., natural language combined with maps/icons), unable to coordinate distributed tasks across resource-constrained devices (e.g., drones with limited battery power or processing capability) and adapt dynamically to real-time requirements. Considering that a user can have multiple AI agents, this use case provides detailed descriptions to illustrate how network supports communication and collaboration among different AI agents. In this use case, multiple AI agents collaborate to accomplish a task of going camping with the assistance of the AI-native network. The users' AI-agents include virtual AI assistant, autonomous drone and intelligent vehicle with built-in AI agents, etc. All AI agents of the user are connected through networks. The service flow of this use case is as follows: 1. The user makes a camping plan request to the AI assistant. After receiving the request, the AI assistant utilizes the capabilities of the AI-native network to recognize the user intents and split the task into multiple sub-tasks based on AI models, including finding a suitable camp site, planing optimal route and driving there. 2. Then, the AI assistant instructs the drone to scan a specific area via the network. The instruction includes modal information

	such as maps, icons (identifying specific areas in the map), and text ("Scan this area"), which can be regarded as a type of high-level instruction. 3. The drone is unable to identify high-level instructions (e.g. natural language, pictures, etc.) due to the limited processing capability. Therefore, the network needs to assist the drone in performing multi-modal transformation, i.e., convert the high-level instruction into simple flight control instructions by MLLMs (Multimodal Large Language Models), and then send them to the drone. 4. Subsequently, the drone flies to the specific area based on the simple flight control instructions to collect video information and asks the network to analyse the video to decide a suitable camp site and the optimal route for arrival. 5. The AI-native network determines the computing power required for the task, selects suitable AI models and computing nodes to perform the video analysis. After finishing the sub-task of analysis, the network sends location of the camp site and the optimal route to the camp site to the AI assistant. 6. The AI assistant then sends the location of the camp site and the optimal route to the intelligent vehicle, and asks the intelligent vehicle to drive there.
Core concepts / characteristics of AINN (Optional)	Refers to core concepts or characteristics of AINN applicable to the use case. The terminology will be updated to reflect the work done in WG1.
Actors and Roles (Optional)	Task control function: use LLMs to recognize the user intents and split the task into multiple sub-tasks. Multi-modal transformation: use MLLMs inside the network to transform the initial modality into a target modality that the agent can recognize. Data analysis and processing: use AI models to analyse the video and extract semantic to meet the demands of the user.
Technical challenges or Gaps (Optional)	Traditional networks fail to meet demands of communication and coordination among AI agents, including the requirements of interpreting high-level instructions, coordinating AI agents across different devices, and adapting dynamically to real-time requirements. The AI-native network should support the communication and coordination for multiple AI agents by: 1. Providing the native AI capabilities to interpret high-level and multi-modal instructions required for complex user tasks. 2. Efficiently coordinating distributed tasks across resource-constrained devices. 3. Providing the data analysis and processing capability to support the dynamic requirements of AI tasks.
Use case category	AI-agent-in AI native networks
Reference	[b-ISO/IEC 61508], [b-ISO 26262]

7.6.2 Assumptions

In this use case, multiple AI agents collaborate to accomplish specific task (e.g. going camping) with the assistance of the AI-native network. All AI agents are connected through networks.

7.6.3 Use case requirements

a. Functional requirements

- The network is required to provide intent recognition and task decomposition capability.
- The network is required to support the transformation of multi-modal information for AI agents.
- The network is required to provide the data analysis and processing capability.
  Note: The data producers can be e.g. LIDAR, cameras, and other sensors [b- ISO/IEC 61508] [b-ISO 26262].
- The network is required to provide AI models specific to different tasks, and regularly fine-tune or update the AI models to achieve better performance.
- The network is recommended to enable AI models to mutually share datasets with each other.
- The network is recommended to provide knowledge bases containing knowledge from different fields to train AI models.
  Note: The AI models can be trained based on knowledge bases to accomplish tasks in specific fields.

b. Performance requirements

- The network is required to improve the resource utilization for coordination of multiple AI agents through AI-native models.
- The network is required to realize the seamless communication of multiple AI agents.

c. AI agent-level requirements

- The AI agents are required to receive different forms of instructions issued by users.
  Note: In this use case, the user can send a camping plan request to the AI assistant by audio, context, and gesture, etc.
- The AI agents are required to support requesting network services of the AI-native network.
  Note: In this use case, the network services includes intent recognition, task decomposition, multi-modal transformation and video analysis, etc.
- The AI agents are required to perform actions according to the task requirement.
  Note: In this use case, the autonomous drone takes video of the camp site to assist the selection of camp site and the intelligent car autonomously drives users to the camp site based on the location, etc.
- The AI agents are required to support the interaction among multiple AI agents.
  Note: In this use case, the AI assistant instructs the autonomous drone to scan a specific area and asks the intelligent vehicle to drive to the camp site, etc.

7.6.4 Use case specific figures

This clause includes the figures used to describe the use case.

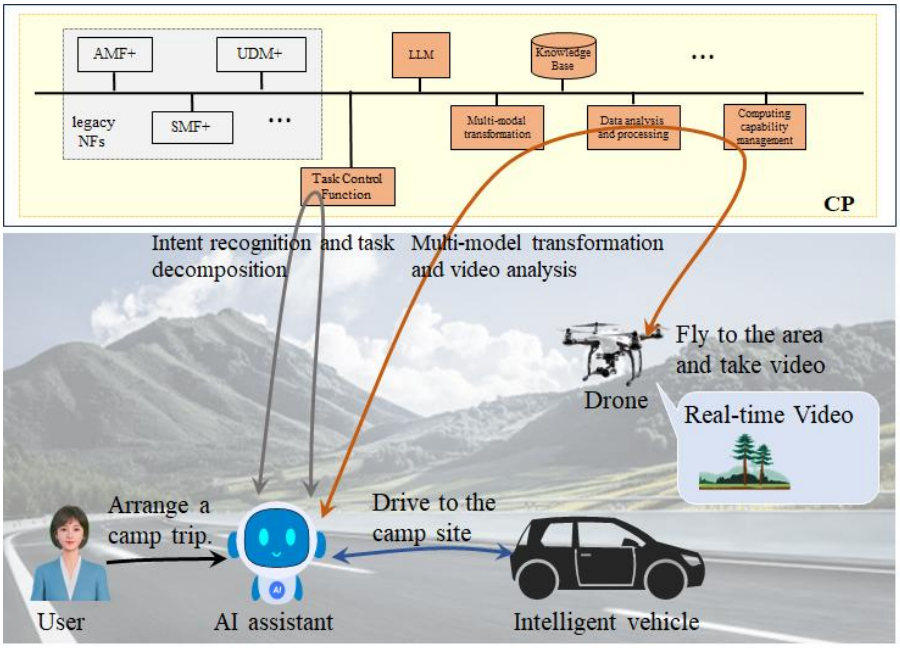


Figure 1: Network realize the communication and coordination among multiple AI agents

7.6.5 Bibliography

[b-ISO/IEC 61508] ISO/IEC 61508: Functional safety of E/E/PE safety-related systems
[b- ISO 26262] ISO 26263-9:2018 - Road vehicles — Functional safety — Part 9: Automotive safety integrity level (ASIL)-oriented and safety-oriented analyses

7.7 Use case 007: AI-Native Proactive Network Slice Manager for Dynamic Service Ecosystems

The telecommunications landscape is rapidly evolving to meet the increasingly diverse and dynamic service demands of next-generation digital ecosystems. Verticals such as manufacturing, healthcare, and autonomous systems require differentiated network behaviours that go beyond the capabilities of traditional static architectures. In response, network operators are transitioning toward AI-native paradigms, where artificial intelligence is embedded throughout the network lifecycle from design and orchestration to assurance and adaptation.

This use case introduces a proactive network slice manager designed to enable AI-native collaboration between telecom operators and enterprises. The system supports the automated provisioning of customized network slices with guaranteed service-level agreements (SLAs), based on enterprise-specific intents. It leverages intelligent policy interpretation, vendor coordination, and closed-loop service assurance to dynamically allocate network resources in real time.

While 3GPP network slicing provides the foundational capability for logical network separation, this use case addresses the limitations of manual, static slice provisioning. By incorporating AI-native behaviour such as intent recognition, demand forecasting, and SLA compliance monitoring, the proposed manager enables context-aware, self-adjusting slice delivery. The framework aligns with multi-operator service models such as those inspired by the FreeMove Alliance, supporting cross-domain SLA negotiation and slice federation.

This use case aims to define the functional requirements and service expectations for such an AI-native slice manager, including its interaction with enterprise clients, vendor systems, and assurance functions.

7.7.1 Use case summary

Use case id	FG-AINN-usecase-007

Use case name	AI-Native Proactive Network Slice Manager for Dynamic Service Ecosystems
Created by	SRM Institute of Science and Technology
Creation date	13 May 2025
Use case context	Submitted to the FG-AINN 3rd meeting in New Delhi, approved in the 17th WG2 meeting (2025.09.18)
Use case description	The flow of the use case begins when an enterprise submits its service requirements, such as latency, reliability, or bandwidth, through the Customer Interface. These high-level intents are then processed by the Intent Parser, which uses LLMs to translate natural language requests into structured SLA parameters. The translated inputs are passed to the Decision Engine, which evaluates available slice options, ranks potential vendors, and negotiates SLA terms. Once a suitable option is identified, the Slice Orchestrator takes charge, managing slice metadata, initiating deployment, and coordinating with multiple vendor systems. Before deployment into the live network, the Network Digital Twins (NDTs) simulate and validate the slice configurations to ensure compliance with SLA and performance expectations. After validation, the Network instantiates the slice, making it operational for the enterprise. The deployed slice is then continuously monitored by the Proactive Feedback Loop, which checks performance metrics and triggers automatic adjustments or redeployments if SLA violations or anomalies are detected. Simultaneously, the Analytics and Learning Engine collects usage data, detects violations, and refines AI models, feeding insights back into the Decision Engine and Orchestrator. This closed-loop process ensures real-time responsiveness, self-healing, and continuous optimization of services across dynamic multi-operator environments.
Core concepts / characteristics of AINN (Optional)	**AI-Native Network Fabric** The network is architected with embedded intelligence across all layers, enabling autonomous decision-making without relying on static configurations. Each functional component supports learning, adaptation, and intent interpretation as part of its core behavior. **Modular Distributed Intelligence** The system leverages distributed AI functions—each assigned a specific operational role, such as interpreting service intents, selecting optimal resources, and detecting anomalies. These AI components operate independently but collaborate through defined interfaces, enabling flexibility and scalability across domains.

	Closed-Loop Control and Optimization The architecture incorporates continuous monitoring and feedback mechanisms. Real-time service data is used to assess performance, detect SLA deviations, and trigger automated remediation actions such as slice scaling, migration, or vendor re-selection. **SLA-Driven Service Orchestration** All orchestration and resource allocation decisions are guided by service-level agreements (SLAs). Parameters such as latency, reliability, bandwidth, and service isolation are prioritized and enforced throughout the slice lifecycle, ensuring alignment between enterprise expectations and network behavior.
Actors and Roles (Optional)	**Enterprises** • Role: Service consumers that request tailored network slices to meet application-specific service-level requirements. • Examples: Manufacturing firms, healthcare institutions, media and content providers. **Operators** • Role: Network service providers that offer slicing capabilities across RAN, transport, and core domains. They provide resources and negotiate SLA terms with enterprises. • Examples: Telecom operators participating in federated service frameworks or multi-operator alliances. **AI Functions / Agents** • Role: Intelligent network components that autonomously interpret service intents, evaluate available resources, negotiate SLAs, detect anomalies, and coordinate slice orchestration and assurance. • Examples: Intent interpreters, SLA optimizers, assurance and anomaly detection modules. **Regulatory and Governance Bodies** • Role: Ensure that network operations and AI functions comply with regulatory standards, data protection laws, and ethical guidelines. • Examples: National and international regulators enforcing compliance with data privacy, security, and AI transparency regulations.
Technical challenges or Gaps (Optional)	The realization of an AI-native network slice manager presents several important technical and operational challenges that must be addressed to ensure scalable, secure, and compliant deployment. A key challenge lies in the coordination and scalability of autonomous decision functions across multiple enterprise requests and operator domains. Supporting intent-based, multi-agent interactions in real time requires efficient orchestration logic and coordination mechanisms that can dynamically respond to a wide variety of service needs and SLA parameters.

	Data privacy and regulatory compliance represent another critical concern, particularly in cross-operator environments where sensitive service data and policy decisions must be exchanged. The system must support privacy-preserving mechanisms and data minimization strategies that comply with regional and international frameworks such as GDPR and NIS2. Interoperability with legacy infrastructure also presents a significant gap. Many existing OSS/BSS systems were not designed to support AI-native, real-time service logic, necessitating adaptive integration layers that enable translation between legacy data models and modern policy-based architectures. The real-time enforcement of SLAs across federated domains introduces further complexity, as the system must continuously monitor key performance indicators such as latency, jitter, and packet loss, while initiating proactive adjustments to maintain compliance. This demands advanced assurance and anomaly detection capabilities embedded within the orchestration flow. Finally, ethical and governance considerations must be embedded into the system's operational logic. This includes ensuring fair resource allocation across users and vendors, avoiding algorithmic bias in SLA negotiation, and maintaining transparency in decision-making. The platform should support ongoing auditing, explainability, and compliance reporting to meet evolving regulatory expectations.
Use case category	AI-native network optimization
Reference	[3GPP TS 28.530], [3GPP TS 23.501], [ITU-T TR-GenAI-Telecom Networks], [ITU-T Y.3172], [ITU-T FG-AINN-WG1-04], [ETSI GR ENI 004]

7.7.2 Assumptions

This use case assumes that the underlying telecommunications infrastructure is sufficiently modernized to support AI-native and cloud-native operations. Operators are expected to have deployed programmable networks with the ability to host distributed intelligence, support edge-based AI inference, and expose orchestration interfaces via standardized APIs (e.g., GSMA Open Gateway).

It is also assumed that both enterprises and service providers participate in federated service ecosystems that enable real-time service discovery, SLA negotiation, and automated provisioning across domains. Industry collaborations or frameworks (e.g., operator alliances) serve as the foundation for multi-vendor interaction and slice federation.

From a governance perspective, it is assumed that national and regional regulatory environments support the use of AI-assisted service management, dynamic SLA contracts, and explainable AI decision-making, in accordance with applicable data protection and cybersecurity directives.

It is assumed that each NDT instance provides a high-fidelity, continuously synchronized virtual replica of the live network, capable of simulating traffic, resource allocation, and SLA compliance under varying conditions. Also, It is assumed that the underlying 5G network infrastructure (RAN, transport, and core) exposes sufficient telemetry, APIs, and programmability features to synchronize with NDTs and to support dynamic slice instantiation, scaling, and reconfiguration as directed by the orchestrator.

Lastly, it is expected that the AI components within the system are trained and optimized using a combination of historical network performance data, real-time telemetry, and compliant synthetic datasets where applicable, ensuring operational accuracy while preserving data privacy.

7.7.3 Use case requirements

To enable an AI-native proactive slice manager over next-generation networks, a layered and cross-functional set of requirements must be addressed. These span the enterprise application interface, AI/ML systems, network domains, slice vendors, and assurance frameworks.

a. Functional Requirements

i. Application-Level Requirements

1. It is of critical value that intent-based submission of slice requests is enabled via high-level API interfaces using natural language or structured templates.
2. It is expected that the system supports multi-domain SLA attributes (latency, throughput, reliability, energy efficiency) within a single slice request.
3. It is of added value that secure, open APIs are provided for integration with enterprise systems (e.g., hospital IT platforms, factory controllers) and allow subscription to real-time SLA dashboards.

ii. Operator-Facing Requirements

1. It is of critical value that dynamic resource allocation and orchestration are supported across all network domains.
2. It is expected that detailed audit logs of provisioning and AI decisions are available for compliance.
3. It is of added value that standardized APIs are used for interoperability with other platforms.

iii. End-User Facing Requirements

1. It is of critical value that open orchestration APIs enable end-to-end slice provisioning.
2. It is expected that dynamic QoS enforcement and traffic prioritization are present.
3. It is of added value that the system includes advanced fault detection and self-healing features.

iv. AI and Network Layer Requirements

1. AI Layer Requirements

1. It is of critical value that LLM-based intent interpretation and SLA negotiation are supported with policy alignment and domain fine-tuning.
2. It is expected that modular and reusable AI agents are enabled for traffic prediction, SLA scoring, vendor ranking, and anomaly detection.
3. It is of added value that real-time AI inference pipelines are provided for dynamic SLA adjustments and slice assurance using closed-loop feedback.

2. Network Layer Requirements

1. It is critical value that allow end-to-end slice provisioning across disaggregated infrastructure using open orchestration APIs.
2. It is expected that dynamic QoS enforcement, traffic prioritization, and slice-level failover are integrated.
3. It is added value that fault detection and self-healing mechanisms are embedded to ensure high service availability and SLA compliance.

b. Performance Requirements

- It is of critical value that customized network slices are provisioned within ≤ 5 seconds of SLA agreement under nominal conditions.
- It is expected that >95% accuracy is achieved in mapping enterprise intents to valid slice configurations via LLMs.

- It is of added value that the anomaly detection false positive rate is kept under 2% using advanced models (e.g., LSTM, CLIP-based).
- It is of critical value that SLA violations are detected and remediated within ≤ 500 milliseconds, leveraging predictive assurance mechanisms.
- It is expected that at least 99.99% slice availability is maintained in high-reliability domains (e.g., healthcare, industrial automation).
- It is of added value that slice provisioning workflows are logically isolated for compliance and auditability under multi-tenant operations.
- It is of critical value that the architecture handles 1,000+ concurrent enterprise requests via horizontally scalable systems.
- It is expected that orchestration of ≥ 10 active slice profiles is enabled simultaneously (e.g., eMBB, URLLC, mMTC, time-sensitive transport).
- It is of added value that optimal resource allocation is achieved by predicting demand and reducing over-provisioning through analytics-based optimization.

7.7.4 Use case specific figures

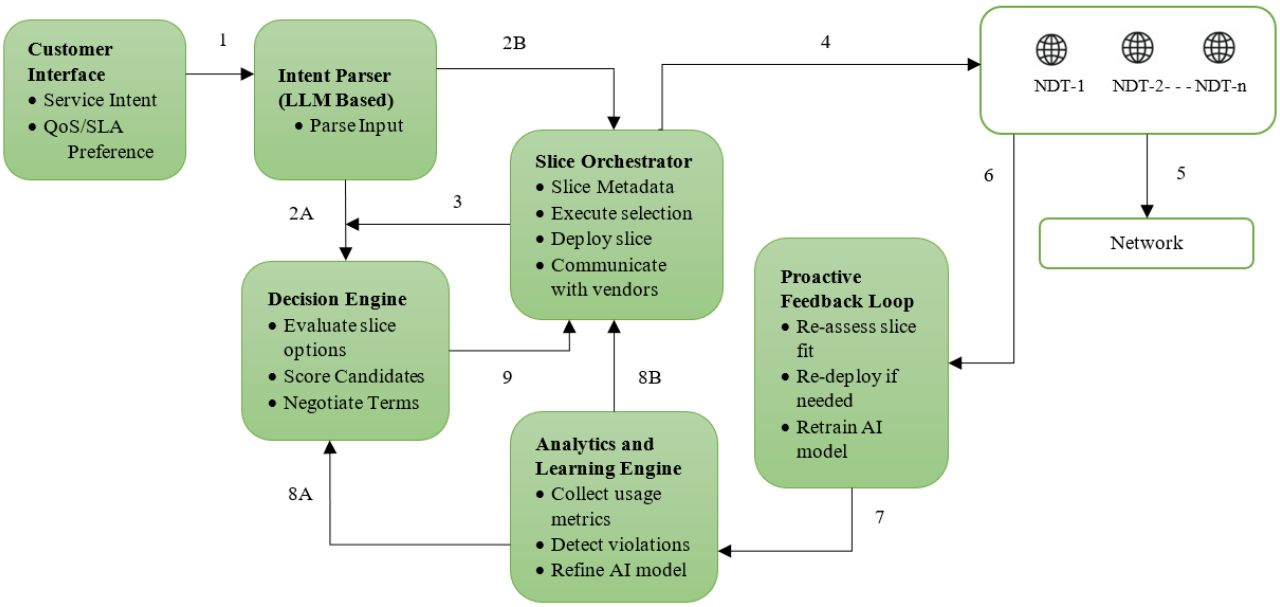


Figure 7.7-1: AI-Native Network Slice Manager Architecture

Key components and their roles in the system:

1. **Customer Interface**

Role: Acts as the entry point for customer requirements.

Input: Service intent, QoS/SLA preferences.

Output: User intent specifications sent to the Intent Parser.

2. **Intent Parser (LLM Based)**

Role: Translates natural language or high-level service intent into structured inputs for the orchestration system.

Input: Customer intent and preferences.

Output: Parsed intent forwarded to Decision Engine (2A) and Slice Orchestrator (2B).

3. **Decision Engine**

Role: Evaluates available slice options, scores candidates, and negotiates terms.

Input: Parsed intent from Intent Parser.

Output: Selected slice options and negotiation results sent to Slice Orchestrator.

4. **Slice Orchestrator**

Role: Manages slice metadata, executes selection, deploys slices, and communicates with vendors.

Input: Candidate slice options from Decision Engine & parsed intent from Intent Parser.

Output: Slice requests to NDTs (Network Digital Twins), deployed slice details to Proactive Feedback Loop.

5. **Network Digital Twins (NDT-1…NDT-n)**

Role: Provide simulation, validation, and predictive assurance before actual deployment.

Input: Slice metadata and deployment requests from Slice Orchestrator.

Output: Validated and optimized slice configurations sent to the physical Network.

6. **Network**

Role: Provides actual slice instantiation and service delivery.

Input: Deployment commands and configurations from NDTs.

Output: Operational data and performance metrics sent to Proactive Feedback Loop.

7. **Proactive Feedback Loop**

Role: Continuously monitors deployed slices, reassesses fit, redeploys if needed, and triggers AI model retraining.

Input: Performance data and metrics from Network.

Output: Updated slice adjustments, redeployment commands, and AI retraining signals.

8. **Analytics and Learning Engine**

Role: Collects usage metrics, detects violations, and refines AI models.

Input: Performance and operational data from Proactive Feedback Loop.

Output: Refined models, updated insights to Decision Engine (8A) and Slice Orchestrator (8B).

Closed-Loop Workflow Summary:

1. User provides service requirements via the Customer Interface.
2. The LLM-based Intent Parser interprets the input.
3. The Decision Engine selects the best slice option.
4. The Slice Orchestrator deploys the slice in the network.
5. The Analytics Engine monitors the slice's performance.
6. If issues are detected, the Proactive Feedback Loop re-optimizes the deployment.
7. The Security & Trust Layer ensures integrity and vendor accountability.
8. The Network Digital Twin supports simulation, mirroring, and predictive validation across the loop.

7.7.5 Bibliography

[FG-AINN-WG1-04] FG-AINN-WG1-04, Definition of AI-native telecommunication networks.

[ITU-T TR-GenAI-Telecom Networks] ITU-T TR-GenAI-Telecom Networks, Requirements and methodology for deploying and assessing Generative AI models in telecom networks, 2025.

[ITU-T Y.3172] ITU-T Y.3172, Architectural framework for machine learning in future networks including IMT-2020, 2019.

[3GPP TS 28.530] 3GPP TS 28.530, Management and orchestration; Concepts, use cases and requirements, Release 16, 2020.

[ETSI GR ENI 004] ETSI GR ENI 004 V3.1.1, Experiential Networked Intelligence (ENI); Terminology, 2023.

7.8 Use case 008: Creation of data for machine learning in AI-native networks

The substantial advantages that AI-native networks can provide above traditional communications systems rely upon having access to high-quality data. The lack of relevant data limits the performance of ML models, which reveals the need for a data generation framework tailored to AI-native networks.

However, data is not always accessible or fits the purpose for training specific machine learning (ML) models, given that existing solutions for collecting data are oriented to mass log reporting and analytics. Moreover, data are often subject to strict privacy regulations, proprietary restrictions, and scalability limitations, making it difficult to collect diverse and representative datasets.

This contribution proposes a new use case for AI-native networks aimed at defining new capabilities for creating data to be used to train, evaluate, and validate ML models. The realization of this new use case would entail defining new components and procedures for the creation and validation of data, including mechanisms to execute measurement campaigns, create synthetic data, or assessing the quality of the created data.

7.8.1 Use case summary

Use case id	FG-AINN-usecase-008
Use case name	Creation of data for machine learning in AI-native networks
Created by	Francesc Wilhelmi, Universitat Pompeu Fabra (UPF)
Creation date	August, 2025 (revised in September, 2025)

Use case context	Submitted to the 14th WG2 meeting, and approved in 17th WG2 meeting
Use case description	AI-native networks are expected to enhance the capabilities of future communications systems (e.g., enabling autonomy, increased efficiency and flexibility, higher performance, and unlocking novel functionalities) through the usage of AI. The AI approach is, however, data-centric, which comes with the need to have access to high-quality data that can be used for training, testing, and validating ML solutions. The lack of relevant data limits the performance of ML models, which reveals the need for a data generation framework tailored to AI-native networks.
Core concepts / characteristics of AINN (Optional)	• **Knowledge Base (KB):** Features specific state (e.g., target network topology and configuration) is used to generate feature specific data (e.g., what-if performance). • **Feature specific applications:** KB, AI models, and solutions are used to generate data for machine learning. • **AI technologies:** GenAI, distillation of knowledge, standardised metadata, data and model catalogues, synthetic data generation are technologies used for generating and managing data for machine learning.
Actors and Roles (Optional)	• **AI/ML data source:** entity that generates the data for ML or for creating new data. • **AI/ML data processor:** entity that processes the acquired/created data and prepares it to be part of the ML model life cycle. • **AI/ML data storage:** entity in charge of storing and maintaining (including versioning) the data for ML. • **AI/ML data validator:** entity in charge of validating the data for ML, including quality assessment and compliance. • **AI/ML data creator:** entity that generates new data (e.g., synthetic data) for ML. • **AI/ML data consumer:** entity that makes use of the data (e.g., for ML model training). • **AI/ML performance validator:** entity that assesses the performance of the ML solutions using specific created data.
Technical challenges or Gaps (Optional)	• **Quality of the data:** To ensure that the created data fits the purpose of training, evaluating, or validating ML models, new validation metrics need to be developed. Data bias is also an important challenge to be considered within this category. • **Privacy preservation:** The data generated must ensure that privacy is not compromised, especially when data are shared among multiple parties (e.g., *data as a service*) • **Scalability and efficiency:** Generating vast quantities of data (e.g., through generative AI or simulations) might require significant computation and storage resources. Creating data in a scalable and efficient manner remains an open issue.
Use case category	Data support for AI-native networks
Reference	[ITU-T Y.3172] Recommendation ITU-T Y.3172 (2019), "Architectural framework for machine learning in future networks including IMT-2020"

	[ITU-T Y.3174] Recommendation ITU-T Y.3174 (2020), Framework for data handling to enable machine learning in future networks including IMT-2020 [ITU-T Y.3181] Recommendation ITU-T Y.3181 (2022), "Architectural framework for machine learning sandbox in future networks including IMT-2020"

7.8.2 Assumptions

- The AI-native telecommunication network supports ML operations.
- The AI-native telecommunication network has access to network data.
- The AI-native telecommunication network has the necessary resources to run the processes involved in this use case.

7.8.3 Use case requirements

a. Functional requirements

Requirement a-1: Collect, process, and store various types of data. While data collection requires having control on the granularity and resolution of the data, data processing comprises ML-specific operations such as data cleaning, transformation, or labeling.

Note: Data including, but not limited to, performance-related metrics (e.g., traffic statistics, transmission errors, throughput, latency), technology specifications (e.g., protocols adopted, standard versions, types of devices, computation/storage capabilities), configuration (e.g., routing tables, MIMO configuration), network logs (e.g., failures, attacks), user/application data (e.g., application type and requirements, utilization/behavior, mobility), environmental data (e.g., off-network data, sensing data), etc.

Requirement a-2: Ability to abstract relevant information about the network and its environment.

Requirement a-3: Perform measurement campaigns on demand (leveraging the capabilities from Requirement 1). Optionally, measurement campaigns can be performed based on specific, pre-defined network configuration settings and through the injection of controlled events.

Note: Measurement campaigns should be configurable based on required network geography, timeframe, type of network data, etc.

Requirement a-4: Perform data augmentation to enrich already created datasets.

Requirement a-5: Generate synthetic data using tools such as simulators, emulators, statistical tools, or ML (e.g., generative AI).

Requirement a-6: Improve data creation pipelines based on feedback loops (e.g., including AI performance evaluation, as defined in Requirement b-1).

b. Performance requirements

Requirement b-1: Assess the quality of the data across multiple domains (e.g., performance of ML models, statistical richness, compactness of data), as well as its timeliness and freshness.

Requirement b-2: Assess the privacy aspects of the created data (e.g., differential privacy).

7.8.4 Use case specific figures

To illustrate an example of this use case, the figure below shows the iterative process for creating and refining data for ML-based Channel State Information (CSI) feedback compression in IEEE 802.11:

- **Data generation:** AI/ML data creators produce specialized datasets using diverse tools like real-world testbeds, simulators, or Generative AI.
- **Model development:** AI/ML data consumers (in this case, acting as AI/ML model providers) use these datasets to build and train ML models for CSI feedback compression.
- **Model deployment:** AI/ML model consumers integrate these trained models into their operational network solutions (e.g., commercial AP and STA devices).
- **Feedback loop:** A feedback loop is created where the model consumer reports performance metrics back to the AI/ML model provider. This information is also passed to the AI/ML data creators, leading to the creation of enhanced data for developing more robust models.

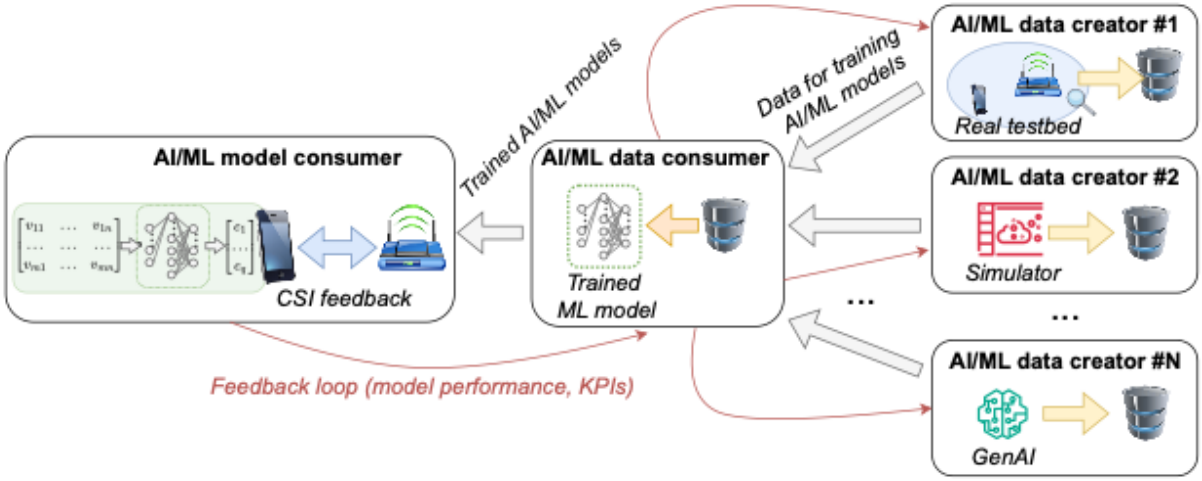


Figure7.8-1: Overall view of the components and processes within the "data creation for ML" use case.

7.8.5 Bibliography

[1] Wilhelmi, Francesc, Marc Carrascosa, Cristina Cano, Anders Jonsson, Vishnu Ram, and Boris Bellalta. "Usage of network simulators in machine-learning-assisted 5G/6G networks." IEEE Wireless Communications 28, no. 1 (2021): 160-166.

[2] Figueira, Alvaro, and Bruno Vaz. "Survey on synthetic data generation, evaluation methods and GANs." Mathematics 10, no. 15 (2022): 2733.

[3] Pandey, Chandrasen, Vaibhav Tiwari, Joel JPC Rodrigues, and Diptendu Sinha Roy. "5GT-GAN-NET: Internet traffic data forecasting with supervised loss based synthetic data over 5G." IEEE Transactions on Mobile Computing 23, no. 11 (2024): 10694-10705.

[4] IEEE 802.11/23-0290r4, "Study on AI CSI Compression."

7.9 Use case 009: AI-Native Digital Twins for Cognitive Network Optimization and Management

This use case presents an AI-native approach to telecom network management by leveraging Digital Twins and AI pipelines to enable intelligent, autonomous decision-making for energy optimization and predictive network analytics. The solution operates using real-time data streaming, forecast models, and control loops to manage network energy consumption and anticipate performance trends. AI-native principles are deeply embedded: AI pipelines manage model lifecycles, a dynamic knowledge base supports state-action decisioning, and feature-specific applications (like energy-saving xApps and traffic forecasting modules) are integrated into the operational loop. The system operates safely through

shadow mode validation and autonomous execution. This use case aims to demonstrate a working PoC in a controlled test environment with potential for scaling into field networks.

7.9.1 Use case summary

Use case id	FG-AINN-usecase-009
Use case name	AI-Native Digital Twins for Cognitive Network Optimization and Management
Created by	Telecommunications Standards Development Society, India (TSDSI), Tata Elxsi Team (Lead: Jesu Maria Antony)
Creation date	May 2025
Use case context	Submitted in May 2025, and approved in 23rd WG2 meeting with some editorial changes for references to be carried out.
Use case description	Operators face significant challenges in **managing energy efficiency, ensuring SLA compliance for network slices, and predicting demand fluctuations**, all while minimizing operational costs and carbon footprint. The **AI-Native Digital Twin [6][7]** approach provides a virtual replica of the live network that continuously synchronizes with real-time operational data. Within this twin environment, **AI pipelines are natively embedded**, enabling predictive analytics, anomaly detection, optimization, and autonomous decision-making. This allows operators to evaluate changes in shadow mode, confirm AI recommendations, and then apply them in autonomous mode with confidence. By integrating **AI models, knowledge bases, and feature-specific applications**, the use case directly addresses telco needs for **self-optimizing, self-healing, and sustainable networks**. It represents a shift from AI as an add-on to AI as a **native enabler of network intelligence**, aligned with the FG-AINN vision [1]. This use case focuses on the deployment of **AI-native digital twins** to enable cognitive optimization and management of next-generation telecommunication networks. The digital twin acts as a **live, synchronized virtual replica** of the physical network, continuously reflecting its operational state across domains (RAN, transport, and core). By embedding AI pipelines directly within the twin, operators can leverage **real-time monitoring, predictive analytics, and autonomous control** to address operational challenges that traditional management systems cannot solve. This use case describes the deployment of a digital twin-enabled AI-native system [1] that includes: **a) AI Pipelines for Network Enhancements [7]** The system includes an AI/ML pipeline layer responsible for orchestrating model lifecycle activities: data collection, training, validation, deployment, and continuous learning. These pipelines control how digital twins interface with live network telemetry, enabling the system to simulate and recommend energy-saving actions or optimal slice allocations in real time. **b) AI Models for Control & Prediction [7]** The solution uses: • **Time-Series Forecasting Models** (e.g., LSTM, GRU) to predict energy usage and traffic load trends.

	• **Anomaly Detection Models** (e.g., Autoencoders, Isolation Forests) to detect abnormal patterns in performance metrics. • **Reinforcement Learning Agents** (e.g., Proximal Policy Optimization (PPO), Deep Q-Network (DQN)) to suggest configuration changes, parameter adjustments, or scaling decisions in an autonomous yet explainable way. All models are trained in sandboxed environments via the AI pipeline, confirmed through the digital twin, and promoted to live inference stages only after meeting accuracy and safety thresholds. **c) Knowledge Base (KB)** The system builds a dynamic, feature-specific knowledge base comprising: • State-action histories per feature (e.g., "load level → scale out"). • Performance outcomes and associated feedback loops. • Policy evolution logs to track why actions were taken. This KB is stored and queried by the AI pipeline and digital twin to enhance decision-making and enable explainability and traceability. **d) Feature-Specific Applications** • **Energy Management**: AI-driven analysis of traffic patterns and load conditions to dynamically optimize energy consumption in network elements, reducing OPEX and supporting sustainability targets. • **Predictive Analytics**: Use of advanced models for traffic forecasting, anomaly detection, and fault prediction, enabling proactive actions to prevent service degradation. • **Network Slice Management**: AI-assisted orchestration and lifecycle management of slices, ensuring SLA adherence, rapid scaling, and efficient resource allocation. • **Change Management**: Shadow-mode validation of AI-driven recommendations before autonomous execution, reducing risks while accelerating adoption of automation. These applications operate via the digital twin orchestration layer and execute changes under either shadow (observational) or autonomous (active) modes.
Core concepts / characteristics of AINN (Optional)	The **AI-native approach**, as defined by FG-AINN [1], enables the development and operation of autonomous, intelligent, and context-aware networks. This use case aligns perfectly with that vision by embedding AI pipelines into every phase of network management from insight generation to action execution while continuously learning from real-world data via digital twin feedback loops. **AI Pipelines as Native Building Blocks** • AI pipelines are not external tools but are **natively embedded into the network architecture**.

	• They drive continuous model training, deployment, and lifecycle management for tasks such as energy optimization, anomaly detection, and predictive traffic forecasting. **Digital Twin as an AI-Native Environment** • The digital twin is the **central environment where AI models operate**, not just a monitoring tool. • It enables **shadow-mode validation and autonomous execution**, making AI decisions trustworthy and directly actionable. **Knowledge Base Integration** • A **continuously updated knowledge base (KB)** stores feature-specific states, action templates, and historical learnings. • This ensures the network learns from past behavior and evolves its operational policies without manual intervention. **Feature-Specific Applications** • Applications such as **energy management, predictive analytics, and slice orchestration** are built as **AI-native apps** that directly use AI pipelines and the KB. • These run on the application layer of the Telcom Network enabled through this platform. • This differs from bolt-on analytics platforms since the apps are integral to the network's functioning. **Autonomous Decision-Making Framework** • AI agents (e.g., reinforcement learning) take **real-time control decisions** within the twin and synchronize them back to the live network. • This shifts the network to **self-optimizing and self-healing operations** rather than human-driven or rule-based interventions. **Native Coexistence and Migration Support** • The design includes **mechanisms for migration and interoperability** with non-AI-native networks, allowing evolutionary deployment. • This ensures AI is not a siloed enhancement but part of the **core network management fabric**. **AI-Native Characteristics in the use case** [1] • AI Pipelines train, validate, and manage the models end-to-end. • A knowledge base evolves based on state-action-reward feedback. • The system integrates AI-native Apps for energy and predictive analytics. • Digital twin simulations serve as a safety layer, enabling **shadow mode** before autonomous execution.
Actors and Roles (Optional)	• **Telco** Network **Operator (Primary Actor)** • **Role**: Owns and operates the physical and virtual network infrastructure. • **AI-Native Digital Twin Platform** • **Role**: Acts as the **virtual replica** of the live network for monitoring, prediction, and control.

	• **AI Pipelines & Model Repository**
	  • **Role**: Core AI engine managing training, deployment, and lifecycle of models.
	• **Knowledge Base (KB)**
	  • **Role**: Centralized intelligence repository.
	• **End Users / Customers**
	  • **Role**: Beneficiaries of optimized, energy-efficient, and reliable network services.
	• **Application Layer / Service Providers**
	  • **Role**: Providers of applications that leverage optimized network resources.
	• **Regulators / Standards Bodies (External Actor)**
	  • **Role**: Ensure compliance, sustainability, and interoperability.

Technical challenges or Gaps (Optional)	1. **Data Acquisition & Quality**
	  o Ensuring **real-time, high-fidelity telemetry** from distributed network elements (RAN, Core, Transport).
	  o Managing **heterogeneous data formats** and high velocity streams.
	  o Addressing **missing, noisy, or biased data** which may impact training accuracy.
	2. **AI Pipeline & Model Lifecycle Management**
	  o Building **scalable AI pipelines** for training, validation, deployment, and retraining in near real-time.
	  o Managing **diverse AI models** (LSTM/GRU for forecasting, RL agents for slice control, anomaly detectors).
	  o Guaranteeing **explainability, interpretability, and trustworthiness** of models in production.
	3. **Digital Twin Fidelity & Synchronization**
	  o Achieving **accurate replication** of complex telco environments in the digital twin.
	  o Ensuring **low-latency synchronization** between the live network and the twin for decision-making.
	  o Balancing **shadow mode testing** versus **autonomous execution** safely.
	4. **Knowledge Base (KB) Construction & Evolution**
	  o Structuring feature-specific knowledge (states, actions, policies) in a machine-usable form.
	  o Enabling **continuous updates** as the network evolves.
	  o Avoiding **knowledge drift** and ensuring relevance across multiple use cases.
	5. **Integration with Legacy & Non-AI Native Networks [2][3]**
	  o Ensuring **backward compatibility** with non-AI-native OSS/BSS systems.

	○ Managing **co-existence and migration challenges** during incremental deployment. 6. **Operationalization & Scalability** ○ Achieving **carrier-grade reliability** with AI-driven automation [2][3]. ○ Ensuring solutions scale across **multi-vendor, multi-domain networks** [3]. ○ Aligning with **sustainability KPIs** (e.g., quantifiable energy savings, latency reduction). 7. **Governance, Security & Compliance** ○ Preventing **adversarial attacks** on AI models or manipulation of training data. ○ Guaranteeing **regulatory compliance** in AI-driven decision-making (e.g., GDPR, lawful interception). ○ Establishing **accountability in autonomous decisions** (who is responsible when AI acts?).
Open issues	• Need for standard interfaces between AI models, digital twins, and existing OSS/BSS systems • Defining robust governance and safety mechanisms for autonomous decision-making • Scalability and generalizability to national/multi-vendor deployments • Security and privacy concerns regarding data collection for model training and twin simulations • Long-term drift handling for time-series models and reinforcement agents
Use case category	AI-Native Digital Twins

7.9.2 Assumptions

a) **Network Infrastructure & Data Availability**

- The telco network elements (RAN, Core, Transport) can expose **real-time telemetry and control APIs**.
- Historical performance, traffic, and energy consumption data are available for **model training and baseline benchmarking**.
- Network elements are instrumented with the **necessary sensors and monitoring probes** to collect energy and traffic KPIs.

b) **AI Pipeline Readiness**

- A standardized **AI pipeline framework** (aligned with AI-Native Networking requirements) is available for model training, validation, deployment, and continuous updates.
- There are **sufficient computing resources** (cloud/edge) to manage both training and real-time inference.
- MLOps practices (versioning, monitoring, retraining triggers) are in place.

c) Digital Twin Environment

- A **fidelity-appropriate digital twin** exists for the target network domain(s), capable of simulating traffic patterns, energy use, and network slice behaviors.

- The twin can operate in **both shadow mode** (observing without interfering) and **autonomous mode** (executing control actions).
- Synchronization between live and twin environments should be **low-latency and dependable**.

d) Knowledge Base (KB) Integration [1]
- A knowledge base exists or should be constructed to store **feature-specific states, actions, and policies**.
- The KB is assumed to be **continuously updated** from real-world operations and feedback loops.
- Standardized interfaces exist to query and update the KB in real-time.

e) Applications & Control Layer [1]
- Applications like **energy optimization xApps**, **network slice management apps**, and **predictive analytics modules** can be integrated on top of the AI pipeline.
- Policy and orchestration frameworks (e.g., ONAP, O-RAN SMO) are available to **enforce AI-driven decisions** back into the network.

f) Operational Assumptions
- Initial deployment will be in a **controlled environment** (lab or limited field trial) before scaling to production.
- Operators allow **shadow testing mode** before transitioning to autonomous operation.
- Multi-vendor interoperability challenges are assumed to be addressed by **open APIs and standard compliance**.

g) Governance & Security
- AI model decisions are **traceable and explainable**, satisfying regulatory requirements.
- Proper **data privacy, security, and consent frameworks** are assumed to be in place for telemetry and training data.
- There are mechanisms for **human-in-the-loop validation** in the preliminary stages of deployment.

7.9.3 Use case requirements

a) Data and Digital Twin Requirements
- REQ-01: Real-time data ingestion from network elements (RAN, Core, Transport) including energy consumption, traffic load, faults, and performance metrics.
- REQ-02: Digital twin model fidelity must accurately represent network topology, configurations, and operational states at multiple granularities (cell, slice, service).
- REQ-03: Continuous synchronization between real network and digital twin, ensuring near real-time reflection of changes and events.
- REQ-04: Multi-domain integration capability (RAN, transport, core) for end-to-end network optimization.

b) AI Pipeline & Model Requirements [1]
- REQ-05: AI pipelines must support training, validation, deployment, and lifecycle management of models for traffic prediction, anomaly detection, energy optimization, and slice orchestration.
- REQ-06: Model diversity support (e.g., forecasting models like LSTM/GRU, anomaly detection models like Autoencoders, and reinforcement learning agents for optimization).
- REQ-07: Online and federated learning support to adapt models dynamically with new data while preserving data privacy.
- REQ-08: Explainability and transparency of AI models to ensure trust and regulatory compliance.

c) Knowledge Base Requirements [1]
- REQ-09: Feature-specific state collection (e.g., slice SLA states, traffic load conditions, energy usage).

- REQ-10: Policy and action template repository that allows AI agents to map network states to recommended actions.
- REQ-11: Continuous knowledge enrichment through feedback from actions executed in shadow and autonomous modes.

d) Functional and Application Requirements [1]

- REQ-12: Energy management application to dynamically adjust network power states based on AI insights.
- REQ-13: Predictive analytics application to forecast traffic demand and pre-allocate resources.
- REQ-14: Network slice management application to ensure SLA adherence, reconfigure slices, and scale resources automatically.
- REQ-15: Autonomous change management to oversee network reconfigurations with minimal operator intervention.

e) Operational Requirements

- REQ-16: Shadow mode execution to validate AI-driven recommendations before applying them autonomously.
- REQ-17: Autonomous execution mode to enable self-optimizing, self-healing network operations.
- REQ-18: Operator dashboard to provide visualization, insights, and control over AI-driven decisions.
- REQ-19: Compliance with privacy, trust, and security frameworks to safeguard network and customer data.
- REQ-20: Interoperability with legacy (non-AI-native) networks to ensure smooth migration and coexistence.

7.9.4 Use case KPIs

Here are the Key Performance Indicators (KPIs) [2][3] for the use case.

a) Energy Management KPIs

- Energy Savings (%): Reduction in energy consumption of network elements compared to baseline operations.
- Carbon Footprint Reduction (tons CO_2e): Sustainability impact due to optimized energy use.
- Dynamic Power Utilization Ratio: Efficiency in scaling power use with network traffic.

b) Predictive Analytics KPIs

- Prediction Accuracy (%): Accuracy of traffic forecasting, anomaly detection, and fault prediction.
- False Positive/Negative Rate: Reliability of AI-driven predictions.
- Time-to-Detection (ms/s): Latency from event occurrence to AI detection.

c) Network Slice Management KPIs

- Slice SLA Adherence (%): Compliance with performance, latency, and throughput guarantees.
- Slice Reconfiguration Time (s): Speed of adaptation to dynamic traffic and service demands.
- Resource Utilization Efficiency (%): Ratio of allocated vs. utilized resources per slice.

d) Operational Efficiency KPIs [4]

- Automation Ratio (%): Percentage of network operations performed autonomously by AI. [2]
- Mean Time to Adaptation (MTTA): Time taken by AI-driven digital twin to recommend and apply optimizations.
- Operator Intervention Frequency: Number of manual interventions needed vs. autonomous actions executed.

e) Business & Service Impact KPIs [4]

- OPEX Reduction (%): Cost savings achieved through AI-native automation. [2]
- Service Uptime/Availability (%): Improvement in network reliability and resilience.
- Customer QoE (Quality of Experience) Score: Measured via latency, jitter, and user satisfaction.

7.9.5 Use case specific figures

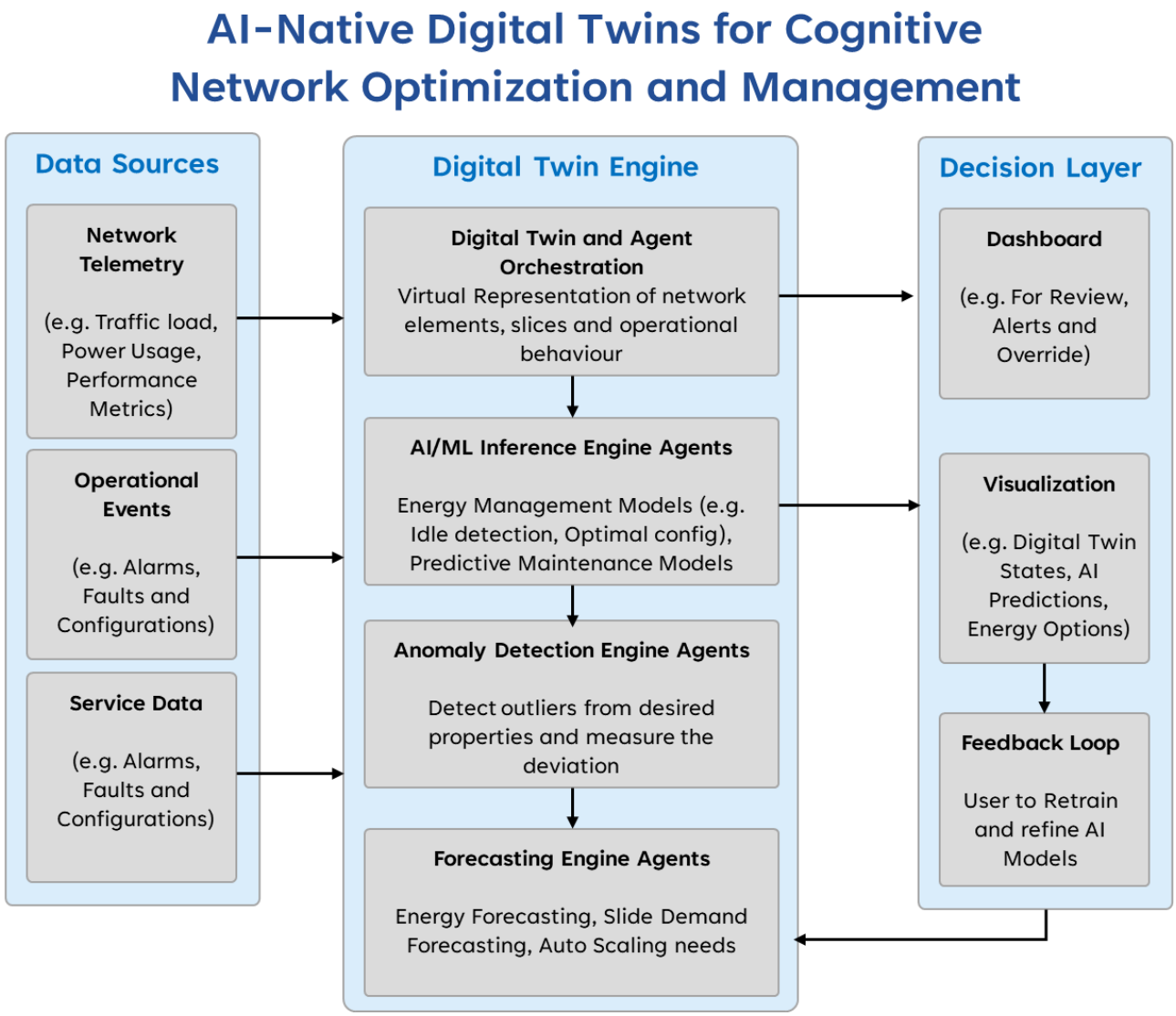


Figure 7.9-1 AI Native Digital Twins for Network Optimization and Management

The **core motivation** behind this use case is to design a system capable of **proactive, autonomous decision-making** for energy management, traffic analytics, and slice orchestration without human intervention while still maintaining traceability and explainability. This is possible using AI-native constructs such as continuous model training pipelines, reinforcement learning agents, dynamic knowledge bases, and real-time simulation environments.

Current technologies, which operate in a loosely integrated and reactive fashion, cannot deliver the level of coordination, foresight, and closed-loop control required for such optimizations. They often require manual interpretation of KPIs and reactive configuration changes, which are both time-consuming and error prone. Moreover, existing systems lack the capability to simulate outcomes in a sandbox before applying them to the live network.

Goals and Benefits

- Demonstrate **AI-native network management** using real-time data and digital twins.
- Improve **energy efficiency**, service quality, and automation via intelligent inferences.
- Enable **predictive insights** into network state, capacity, and slice health.
- Support **autonomous change management** with human-in-the-loop governance.

Sequence diagram for the use case:

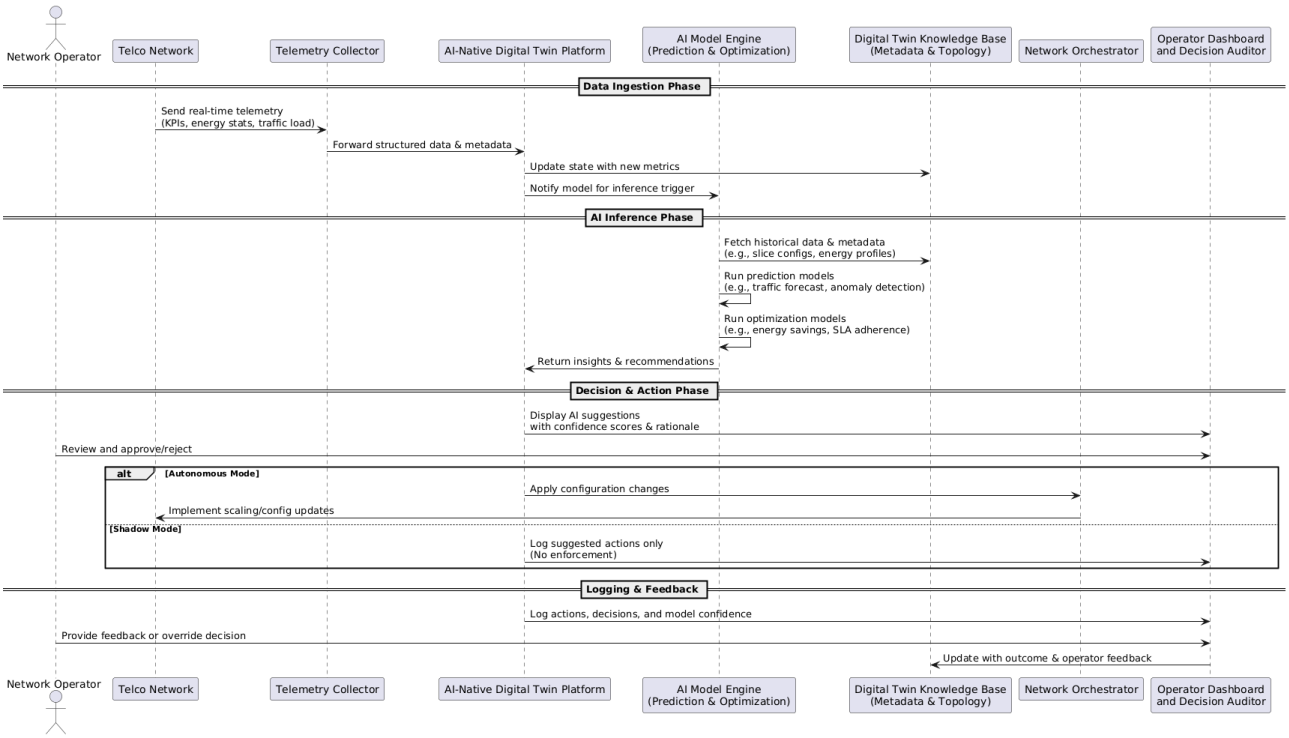


Figure 7.9-1 Sequence Diagram for the Use Case

Functional Components [6][7]

- **Digital Twin Layer**: Models network elements (e.g., base stations, slices, routers) with real-time KPIs and telemetry.
- **AI Inference Engine**: Forecasts load, detects anomalies, recommends or triggers actions.
- **Optimization Controller**: Adjusts configurations or triggers scaling based on AI outputs.
- **Operator Dashboard**: Visualizes state, inferences, and human validation steps.
- **Data Feedback Loop**: Continuously refines models with real-world performance data.

The **expected impact** of this approach includes:

- Significant energy savings through intelligent infrastructure control,
- Better quality of service via predictive traffic and slice management,
- Reduced operational overhead through automation,
- Improved system resilience and adaptability via continuous learning.

7.9.6 Bibliography

[1] **ITU-T Focus Group on AI for Autonomous and Assisted Networks (FG-AINN)**
https://www.itu.int/en/ITU-T/focusgroups/ainn/Pages/default.aspx

[FGAINN AI Native Features] https://extranet.itu.int/sites/itu-t/focusgroups/ainn/input/FG-AINN-I-065.docx
[FGAINN AI Native Features] https://extranet.itu.int/sites/itu-t/focusgroups/ainn/input/FG-AINN-I-038-R4.docx
(Provides guidance on AI-native requirements, including AI pipelines, knowledge bases, and application integration in networks.)

[2] ETSI ENI – Experiential Networked Intelligence
https://www.etsi.org/technologies/experiential-networked-intelligence
(Covers intent-based networking, closed-loop automation, and AI-driven network adaptation.)

[3] 3GPP TS 28.104 – Management and orchestration; Data analytics services
https://www.3gpp.org/ftp/Specs/archive/28_series/28.104/

(Describes service management data analytics architecture and its role in network automation.)

[4] TM Forum Autonomous Networks Whitepaper

https://www.tmforum.org/wp-content/uploads/2019/05/22553-Autonomous-Networks-whitepaper.pdf

(Describes maturity levels and architectures for building self-managing telecom networks.)

[5] Digital Twin Network Blog – TM Forum

https://inform.tmforum.org/features-and-opinion/the-role-of-digital-twins-in-autonomous-networks

(Explores how digital twins support closed-loop automation in telco networks.)

[6] Tata Elxsi Framework – Digital twins & AI/ML Framework Blog

https://www.tataelxsi.com/insights/ai-based-digital-twins-for-multi-intent-6g-networks-an-architectural-perspective

(Defines interfaces and architectures for integrating AI/ML models in Digital Twins and Network Optimization.)

7.10 Use Case 010: Native API support for dynamic distribution of inference tasks in an Edge-Cloud continuum

This contribution proposes a use case and the corresponding requirements for enabling native API support for dynamic distribution of AI based real-time inference utilizing deep neural networks (DNN).

7.10.1 Use case summary

Use case id	FG-AINN-usecase-010
Use case name	Native API support for dynamic distribution of inference tasks in an Edge-Cloud continuum
Created by	Abhijan Bhattacharyya, Ashis Sau, Suraj Mahato from Telecommunications Standards Development Society, India (TSDSI)
Creation date	4-Jun-2025
Use case context	Submitted to the FG-AINN F2F meeting in June 2025 and approved in WG2 meeting (03 July 2025).
Use case description	This use cases proposes to enable native API support for on demand DNN model transfer along with dynamic distribution of workload between the edge and the cloud. The scenario is depicted in Fig. 7.10-1 with a possible arrangement of necessary entities and the interactions amongst themselves.

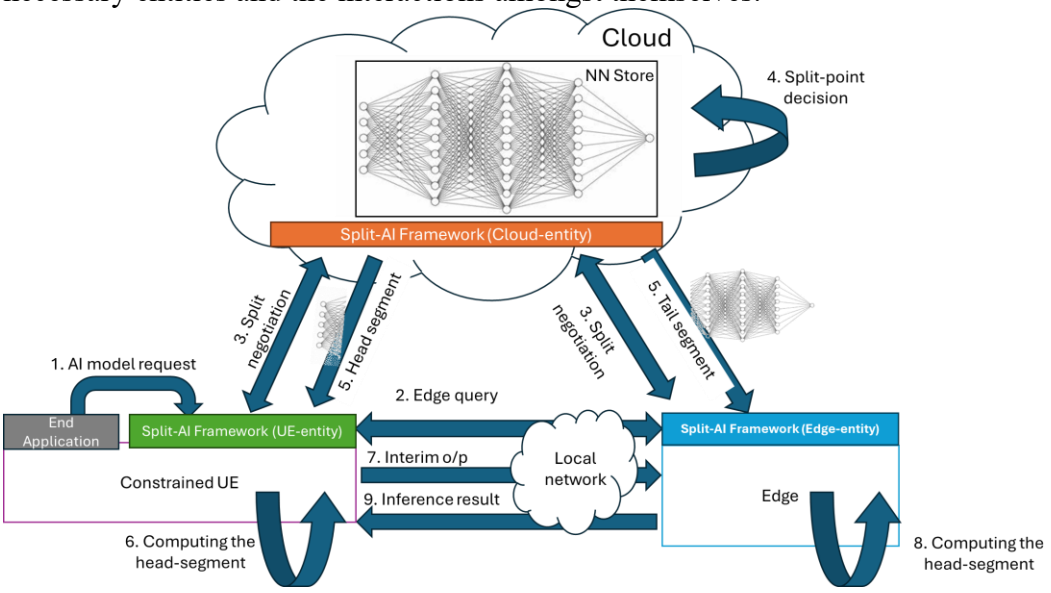


Fig. 7.10-1

	We envision that in a practical deployment the entire communication set up is equipped with a Split Decision Engine (SAF) distributed amongst the UE, the Edge and the Cloud. We consider the generic flow of operation with a constrained UE and an associated Edge. A user application in the UE requests certain inference task that needs a certain model and a desired response time with a desired level of accuracy. The UE does not have the model and needs to get from the Cloud and needs to split it due to reasons as discussed earlier. This triggers the following flow as shown in Fig. 1 above: 1. The UE component of the SAF gets the application intent and also checks UEs current and immediately future processing load, remaining energy and the channel condition connecting the Edge. 2. If authorized, the UE also performs a neighbourhood discovery to find an Edge and negotiates with a suitable one to get the service (the discovery process and negotiation is out of scope). UE components of the SAF gets the system condition for Edge. The Edge component of the SAF responds to the UE counterpart. 3. The UE component of the SAF requests the Cloud counterpart of SAF with these details along with a pointer to the Edge. The cloud performs necessary security exchanges and other communications with both the UE and the identified Edge for getting present device condition, network status and any other required information necessary to compute the splitting point. 4. The Cloud component of the SAF identifies the model and decides on the splitting point. 5. The cloud provides the head segment to UE and tell segment to Edge along with meta-information (if any). 6. The UE application performs the computation with the head-segment 7. The UE application transmits the intermediate result to the Edge. 8. Edge computes rest of the inferencing work using the tail-segment. 9. Edge transmits the inference result to the UE. The native APIs should support similar flows.
KPIs (optional)	The exact KPIs will depend on the latency and resource efficiency for application for which the distribution is performed.
Open issues	• Finding the optimal split position • The model transfer complexity with dynamic splitting
Use case category	Native API of AI-native networks
Reference	See clause 7.10.3

7.10.2 Use case requirements

7.10.2.1 Generic requirements (SAF-Gen)

i. **[SAF-Gen-REQ-01]** It is critical that APIs should allow access to low-level channel properties for all the 3 pairs of links amongst the three entities – cloud, UE and Edge.
ii. **[SAF-Gen-REQ-02]** It is critical that the APIs allow pre-configured hooks for smooth integration of the segments into the run-time application space.
iii. **[SAF-Gen-REQ-03]** It is expected that APIs should have privacy filter to allow end application restrict/ customize information exposure for split decision functions.
iv. **[SAF-Gen-REQ-04]** It is critical that the APIs should enable sufficient authorization and trust establishment amongst all the entities.

v. **[SAF-Gen-REQ-05]** It is critical that the APIs should enable secure exchanges and, especially, enable protection of the model segments from any partial or full adversarial attacks.

7.10.2.2 Requirements specific to the UE component of the SAF (SAF-UE)

vi. **[SAF-UE-REQ-01]** It is critical that the APIs on the UE component of the SAF (SAF-UE) should allow discovery of the nearest suitable Edge with necessary resources.
vii. **[SAF-UE-REQ-02]** It is critical that the APIs at the UE side should enable the UE application to advertise the instant model requirement and the desired accuracy.
viii. **[SAF-UE-REQ-03]** It is critical that APIs should provide fail-safe mechanism by enabling graceful performance degradation or termination in case adversaries due to connection-loss with Edge / Cloud, or non-availability of Edge on demand.

7.10.2.3 Requirements specific to the Edge component of the SAF (SAF-E)

ix. **[SAF-E-REQ-01]** It is critical that the SAF-E APIs should enable advertising the capabilities to enable proper discovery and selection of Edge.
x. **[SAF-E-REQ-03]** It is critical that the SAF-E APIs should enable notification of any adversaries that may hinder its operation with the tail segment and accordingly notify SAF-UE and SAF-C so that either graceful termination of the application process (at the UE), or re-initiating the split process (by involving the cloud) can be done based on situational context.

7.10.2.4 Requirements specific to the Cloud component of the SAF (SAF-C)

xi. **[SAF-C-REQ-01]** It is critical that APIs on the Cloud component of the SAF (SAF-C) should allow setting the security credentials and exchange device and network status with UE and the Edge.
xii. **[SAF-C-REQ-02]** It is critical that SAF-C APIs should enable low-overhead, reliable transfer of the head and tail segments from the ANN store through the split decision engine at the cloud
xiii. **[SAF-C-REQ-03]** It is expected that the APIs should enable loss-less compression and decompression of the head and tail segments in transit, if such loss-less model compression mechanisms are available.
xiv. **[SAF-C-REQ-04]** It is expected that in case the application indicates the desire for suboptimal accuracy, the SAF-C APIs should enable calculations for the necessary early exit strategies and intimate the same to the SAF-UE.

7.10.3 References

[Kang- Neurosurgeon] Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor Mudge, Jason Mars, and Lingjia Tang, “Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge”, Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '17), 2017, https://doi.org/10.1145/3037697.3037698.

[Mehta - DeepSplit] R. Mehta and R. Shorey, "DeepSplit: Dynamic Splitting of Collaborative Edge-Cloud Convolutional Neural Networks," 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS), Bengaluru, India, 2020, pp. 720-725, doi: 10.1109/COMSNETS48256.2020.9027432.

[Zeinali - ESAI] Behnam Zeinali, Di Zhuang, Morris J. Chang, “ESAI: Efficient Split Artificial Intelligence via Early Exiting Using Neural Architecture Search”, arXiv:2106.12549 [cs.DC], 2021.

[Amin - Auto-Split] Amin Banitalebi-Dehkordi, Naveen Vedula, Jian Pei, Fei Xia, Lanjun Wang, and Yong Zhang, “Auto-Split: A General Framework of Collaborative Edge-Cloud AI”, 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21), 2021.

[Li-Edge] En Li, Zhi Zhou, Xu Chen, “Edge Intelligence: On-Demand Deep Learning Model Co-Inference with Device-Edge Synergy”, Proceedings of the 2018 Workshop on Mobile Edge Communications, August, 2018.

[Lin-SL] Lin, Z., Qu, G., Chen, X., & Huang, K. (2023). Split Learning in 6G Edge Networks. ArXiv, abs/2306.12194.

[3GPP-TR-22.876] 3GPP TR 22.876 V19.1.0, Study on AI/ML Model Transfer Phase 2 (Rel. 19), Sep, 2023.

[FGAINN-Inference] Antonio De Domenico, Clarissa Marquezan, Riccardo Trivisonno “Use case – AI Inference as a Service in AI native telecommunication networks,” Document under consideration in FG-AINN, April, 2023.

7.11 Use Case 011: Lightweight Trust: Enhancing Efficiency and Privacy with zk-SNARK

zk-SNARK (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge) are cryptography methods that enable a prover to prove to a verifier the truth of some statement, and conceal some sensitive information in the process. zk-SNARK guarantee security, privacy, and are non-interactive properties worth their weight in gold in uses like cryptocurrencies, secure voting, biometric/authentication for devices, IoT, verifiable cloud and AI computations, etc.
For big-scale systems and low-resource environments, the use of zk-SNARK is hindered by major constraints related to their high demands on speed and memory, limiting their practical scalability.

7.11.1 Use case summary

Use case id	FG-AINN-usecase-011
Use case name	Lightweight Trust: Enhancing Efficiency and Privacy with zk-SNARK
Created by	M.Thenmozhi, G.Ramya, N.Deepa from SRM Institute of Science and Technology
Creation date	17.09.2025
Use case context	Submitted to the 16th WG2 meeting (4th of Sept. 2025), approved in the 29th WG2 meeting.
Use case description	Facial recognition systems often rely on cloud-based storage of raw biometric data, which raises significant privacy and security concerns. Integrating zk-SNARK into such systems allows verification of user identity without exposing sensitive facial data, ensuring compliance with privacy regulations and protecting against unauthorized access. Face recognition systems are known to be an effective means to deliver guarantee of safe authentication in any digital or physiological scenario. For example, face recognition systems have been employed to unlock phones, control physical access, authenticate identities for online authentication, and to monitor (know who is where). Nonetheless, conventional approaches are cloud-dependent and utilize centralized servers to manage and consolidate raw biometric information, which results in privacy and security issues owing to a

number of reasons: since biometric information is driven from storage medium to server, it is exposed to interception, overwhelmed by privacy regulations and expectations, and misaligned with extremely sensitive information related to personal privacy.
It is necessary for modern systems to support **privacy-preserving authentication** without storing or transmitting raw biometric data. Conventional methods fail to address these requirements due to their reliance on cloud services, inability to ensure full data confidentiality, and vulnerability to replay or spoofing attacks. By integrating zk-SNARK, authentication can be performed in a secure and verifiable manner where proofs, not sensitive data, are transmitted and verified [1,2].
Considering that IoT devices (e.g., Raspberry Pi, smart cameras, access points) are increasingly deployed for biometric authentication, this use case demonstrates how zk-SNARK enable **lightweight, privacy-preserving facial recognition**. The system architecture ensures that all biometric data remains local, while only cryptographic proofs are communicated across networks for verification.
The service flow of this use case is as follows:

1. The user attempts authentication through a camera connected to a lightweight IoT device (e.g., Raspberry Pi). The device captures the facial image and generates a **face embedding** using a pre-trained ResNet model
2. Instead of transmitting raw data, the embedding is converted into a **SHA-256 hash**, which serves as a secure representation of the facial features.
3. The IoT device then generates a **zk-SNARK proof** indicating that the hash captured is associated with an identity that is in the authorized database of the system [3].
4. In a zk-SNARK based 6G authentication system, the verifier (deployed at the edge server or authentication server) receives the proof and the public input (e.g., hash of the facial embedding) from the IoT device.
   The verifier, equipped with the verification key (VK) from system setup, runs the algorithm Verify(VK, proof, public input). If the proof is valid, access is granted; otherwise, it is denied. Crucially, the verifier never sees or stores the original biometric data, ensuring secure and privacy-preserving authentication.
5. The verifier confirms the proof and this verification takes milliseconds (average latency 50ms after optimizing). If the proof is valid, access is granted.
6. During this process, no raw biometric data leaves the device, therefore limiting risks to replay attacks, spoofing attempts and data leakage.

This implementation provides cryptographic privacy and real time performance and zk-SNARK provide an effective method to facilitate secure authentication for IoT use cases in smart homes, corporate access control systems, and connected devices.

Core concepts / characteristics of AINN	The AI-Native Network (AINN) provides the autonomous framework to manage secure biometric authentication. ● **Application Layer:** Local IoT devices capture faces and generate **zk-SNARK proofs** so raw data never leaves the device. ● **Service Layer:** Contains the "Automation Brain" and **Verification Module** to validate proofs and grant access in milliseconds. ● **Network/Physical Layers:** 6G infrastructure and **Edge Computing** provide the high-speed telemetry and processing power for cryptographic tasks. ● **Two-Stage Automation:** ○ **Stage 1 (Verification):** The system filters poor-quality images (e.g., blur checks) and establishes cryptographic trust. ○ **Stage 2 (Resource Action):** The network forecasts authentication loads and optimizes edge resources for real-time responsiveness.
Actors and Roles (Optional)	● **Face embedding function:** use a deep learning model (e.g., ResNet) to convert the captured facial image into a mathematical embedding. ● **Hashing function:** use SHA-256 to securely transform the face embedding into a fixed-length representation, ensuring privacy. ● **Proof generation function:** use zk-SNARKs on the IoT device to generate a zero-knowledge proof that the captured face matches an enrolled identity. ● **Verification function:** use lightweight zk-SNARK verification on the server or access system to validate the proof without accessing raw biometric data. ● **Optimization function:** apply parallel proof generation, precomputation, and circuit-level optimizations to achieve low latency and memory efficiency on constrained devices. The use case architecture is depicted in Figure 1.
Open issues (Technical challenges or Gaps) (Optional)	These gaps identify the technical hurdles specific to implementing this biometric use case: ● **Model Heterogeneity:** Difficulty managing different AI models (ResNet for faces vs. RL for circuits) across diverse IoT hardware. ● **Biometric Data Drift:** Accuracy loss over time due to changing lighting or user features, requiring complex secure retraining. ● **Inference Latency:** Standard protocols may be too slow to share real-time AI "status" with the central resource manager.

	● **Telemetry Pipeline Gaps:** Lack of standardized, high-volume paths to collect real-time data from millions of authentication nodes. ● **Programmability Limits:** If edge components aren't fully programmable, the network cannot instantly scale to handle surges in access requests
Use case category	Security and trust
Reference	[ITU-T Y.3173] Recommendation Y.3173 (2020/2025), "Framework for evaluating intelligence levels of future networks including IMT-2020" [ITU-T FG-AINN] Technical Report, "Concepts and characteristics of AI-native telecommunication networks."

7.11.2 Assumptions

In this use case, zk-SNARK are applied to enable privacy-preserving facial recognition on IoT devices (e.g., Raspberry Pi, smart cameras). The user's biometric data is processed locally to generate embeddings and cryptographic proofs, while no raw data is transmitted or stored in the cloud. All communication between the IoT device and the verifier involves only zk-SNARK proofs, ensuring secure, efficient, and private authentication across connected systems.

7.11.3 Use case requirements

a. Functional requirements

- It is of critical value for the system to support AI-native network automation integrated with Zk-SNARK-based verification.
- It is of critical value for the system to enable secure IoT data acquisition and edge-level AI inference.
- It is of added value for the system to support predictive optimization and intelligent resource management.
- It is of added value for the system to enable closed-loop feedback between AI inference and network control.

b. Performance requirements

- It is of critical value for the system to achieve ultra-low-latency cryptographic proof generation and verification.
- It is of critical value for the system to deliver real-time AI inference and decision signaling.
- It is of added value for the system to support high-throughput telemetry processing without delay.
- It is of added value for the system to ensure reliable and deterministic network control actions.
- It is of added value for the system to scale efficiently across device, edge, and core environments.

7.11.4 Use case specific figures

Figure 7.11-1 represents the overall framework of the proposed system.

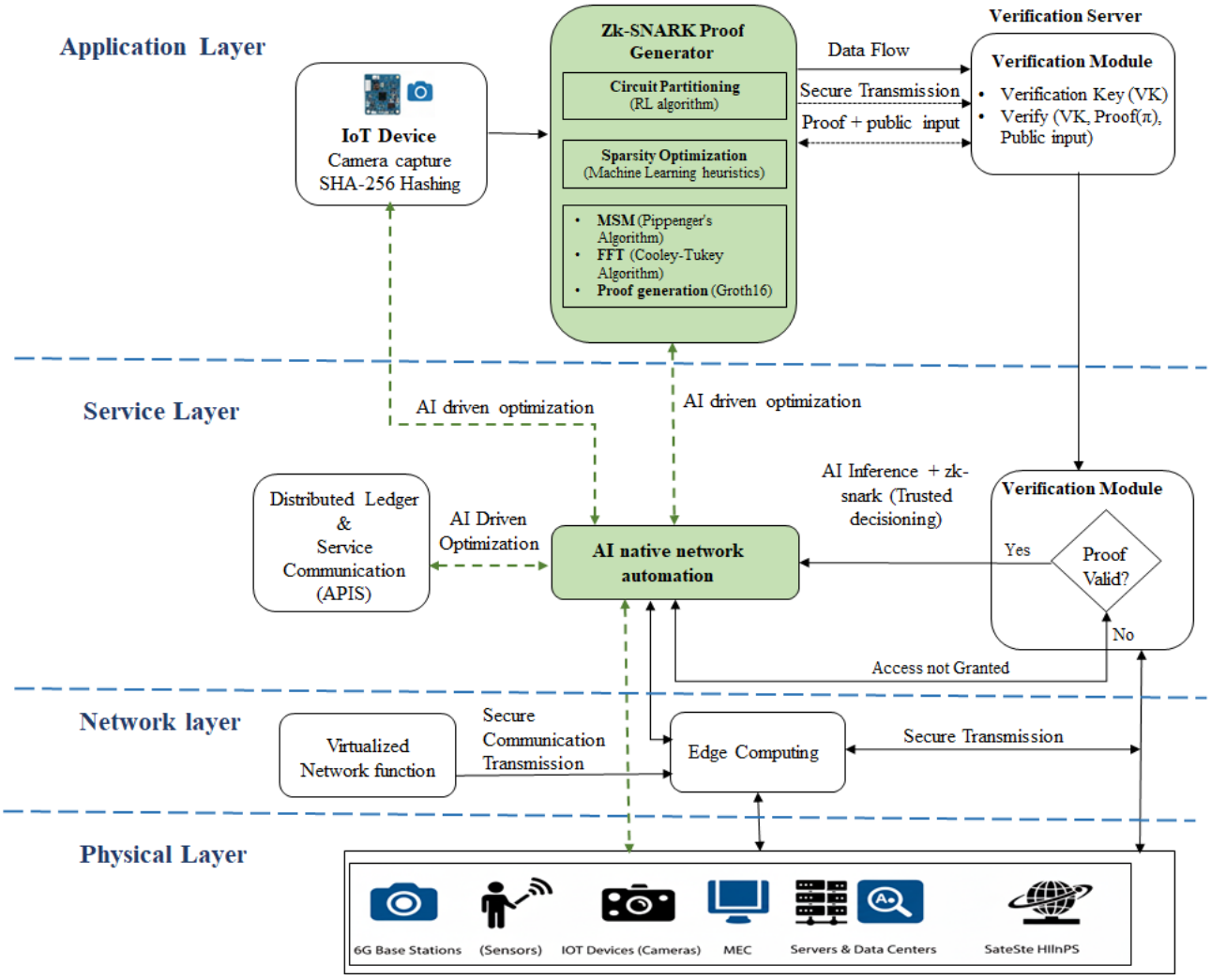


Figure 7.11-1: System architecture for privacy-preserving facial recognition using zk-SNARK

7.11.5 Bibliography

[1] Szczegielniak-Rekiel, A., Kanciak, K., & Kelner, J. M. (2025). "Zero-Knowledge Proof in 5G and Beyond Technologies: State of the Arts, Practical Aspects, Applications, Security Issues, Open Challenges, and Future Trends." IEEE Access, vol. 13, pp. 138352–138380.

[2] Ahmad, I., Gimhana, S., & Harjula, E. (2025). "Adaptive Trust Architecture for Secure IoT Communication in 6G." IEEE Networking Letters, vol. 7, no. 2, pp. 113–116.

[3] Zhang, L., et al. (2025). "Towards Native AI in 6G Standardization: The Roadmap of Semantic Communication." arXiv preprint arXiv:2509.12758.

7.12 Use case 012: AI-driven network performance modelling and prediction

Modeling and prediction of network performance metrics—such as energy consumption, user experience, network load—is critical for intelligent optimization in modern mobile networks, as it provides the quantitative foundation for adaptive decision-making and resource allocation. This requires effective data collection, augmentation, and analysis to capture complex network behaviors. Traditional methods rely on static assumptions and manual measurements, limiting scalability and adaptability. In contrast, AI enables dynamic data collection by selecting relevant KPIs, adjusting spatiotemporal granularity, and applying compression for efficient transmission. It also supports intelligent data

enhancement and advanced analytics for accurate, scalable modeling. As networks grow more complex, AI-driven modeling becomes essential for real-time, cost-effective optimization.

7.12.1 Use case summary

Use case id	FG-AINN-usecase-012
Use case name	AI-driven network performance modelling
Created by	Lehan Wang, Xu Xue and Qi Sun from China Mobile
Creation date	18.02.2025
Use case context	Submitted to the 3rd FG-AINN meeting, approved in the 32nd WG2 meeting (9 April 2026)
Use case description	Modeling and prediction of network performance metrics is essential for enabling intelligent, data-driven optimization in modern mobile networks. Effective modeling relies on a pipeline that includes data collection, augmentation, and analysis to accurately capture the complex dynamics of real-world systems. Traditional non-AI methods face key limitations: data selection is often manual and costly (e.g., through drive tests), augmentation relies on rigid assumptions that fail to reflect network variability, and analysis typically lacks the capacity to handle high-dimensional, nonlinear interactions. In contrast, AI-driven data selection can intelligently select the most relevant types of KPIs, adjust the spatial and temporal granularity of data collection, and apply real-time compression techniques to reduce redundancy and minimize transmission overhead. This process can be realized through the integration of AI Agents within the network. Agents continuously monitor network performance across different regions and orchestrate workflows: they first perform root-cause analysis to determine which KPIs related to network optimization should be reported and define appropriate reporting intervals. Subsequently, Agents leverage the collected data to model and predict network KPIs and generate optimized network configuration parameters. Importantly, these steps are autonomously orchestrated by the network using the Agent's capabilities, rather than relying on manual configuration and preset rules. This targeted and context-aware data acquisition significantly enhances modeling efficiency while reducing the cost of data handling. AI-driven techniques can also perform context-aware data augmentation, such as inferring missing values based on learned correlations, generating synthetic samples to improve model generalization, or enriching sparse measurements with external knowledge (e.g., environment maps or mobility patterns). Furthermore, AI-enabled analysis methods—such as deep learning, graph-based inference, and representation learning—can uncover complex dependencies, nonlinear behaviors, and latent patterns across massive, heterogeneous datasets that are beyond the reach of conventional statistical tools. These capabilities enable scalable, adaptive modeling and are crucial for supporting proactive optimization in 5G and beyond. As networks become increasingly complex and dynamic, AI-based modeling grounded in intelligent data handling is not only beneficial but indispensable.
Core concepts / characteristics of AINN (Optional)	AI-driven Modeling; AI Agent

Actors and Roles (Optional)	This use case involves the following actors and their associated roles: **Network Performance Metric Modeling Function** This actor represents an AI-native functional entity responsible for modeling and predicting network performance metrics. It consists of a set of modular sub-functions, including data collection, data augmentation for training and inference, modeling and prediction, and performance monitoring. These sub-functions can be flexibly instantiated, combined, or bypassed according to task requirements. The function produces predicted and enhanced performance metrics—such as network load, energy consumption, user experience, and refined user location information—which can be exposed to other network and service entities for optimization and service adaptation. **AI Agent for Modeling Orchestration and Control** This actor performs intelligent orchestration and control over the network performance metric modeling function. The AI Agent dynamically determines how the modeling sub-functions are composed and executed. It autonomously selects and adjusts key modeling parameters, including relevant KPI sets, spatiotemporal granularity of data collection, and retraining frequency, based on network context, optimization objectives, and observed model performance. Through continuous performance monitoring, the AI Agent triggers model retraining and policy updates to ensure robust and adaptive modeling under dynamic conditions. **Network and Service Data Providers / Consumers** This actor represents network and service entities—including the access network, core network, and OTT service or application layers—that participate in the use case as both data providers and consumers. On one hand, they generate and supply raw or intermediate data (e.g., radio measurements, traffic statistics, mobility information, and service-level indicators) to support data collection and augmentation within the network performance metric modeling function. On the other hand, they consume predicted and enhanced performance metrics—such as refined user location, anticipated network load, or user experience indicators—to enable network optimization actions and to support context-aware and quality-aware service provisioning. A typical example of data interaction among the above actors is as follows, also in Fig,2. First, Stage ① is data collection and reporting: data providers such as the AN collect data according to the data collection parameters configured in the metrics modeling module and provide the data to the module. In the example shown in Fig. 2, the AN uploads network data, user locations, and other relevant information to the module. Stage ② involves data analysis and prediction within the module. In the example of Fig 2, Stage ③ is the sharing of predicted network metrics with the access network and applications (e.g., future characteristics of application traffic, expected network load) so that the AN and applications can perform resource pre-allocation or dynamic service adaptation.

Technical challenges or Gaps (Optional)	Despite the growing promise of AI-driven modeling for network performance metrics, several technical challenges remain in current network infrastructure that hinder its full realization. First, there is a lack of robust model lifecycle management capabilities within operational networks. AI models trained on historical or static data may experience performance degradation as network conditions evolve; thus, continuous monitoring, validation, and online adaptation of model performance are essential to maintain accuracy and reliability. In practice, such monitoring should not be limited to model accuracy alone but should also encompass multi-dimensional indicators, including network-level KPIs (e.g., throughput, latency, reliability) and system-level metrics such as resource utilization and energy efficiency. A comprehensive view across these dimensions is crucial for ensuring that AI models remain both effective and sustainable throughout their lifecycle.Second, existing systems often lack support for distributed and edge-based model inference and training, which are critical for enabling scalable, low-latency modeling across dense and heterogeneous network deployments. Here, the edge refers to computing resources located close to data sources, including the Central Unit (CU) of the Access Network (AN), the Core Network, and Multi-access Edge Computing (MEC) platforms.Additionally, mechanisms for data versioning, labeling consistency, and feedback integration are underdeveloped, limiting the network's ability to refine and retrain models efficiently based on new observations. Addressing these gaps is essential to support the deployment of AI-native modeling pipelines that are resilient, adaptive, and responsive to the dynamic nature of 5G and future 6G systems.
Use case category	AI-native network optimization
Reference	None.

7.12.2 Assumptions

Networks are equipped with computing resources and data collection abilities that can support AI model inference and training.

7.12.3 Use case requirements

a. Functional requirements

- The network should support continuous AI model lifecycle management, including performance monitoring and adaptive retraining, to maintain modeling accuracy under dynamic network conditions.
- The network should support distributed and edge-based AI computation capabilities to enable low-latency, scalable modeling and inference across large-scale deployments.
- The network should support intelligent data orchestration capabilities, including dynamic KPI selection, spatiotemporal sampling adaptation, and efficient data compression, to construct AI-friendly data pipelines.

b. Performance requirements

- Inference latency should be low enough to support near-real-time decision-making, especially in dynamic or time-sensitive network scenarios. For network performance metric modeling and prediction tasks, latency requirements vary by network layer: milliseconds to tens of milliseconds for access network (AN)-level inference, around

100 ms for core network (CN), seconds for OMC-level operations, and 1–10 seconds at the service or application layer.

- Prediction accuracy of AI models should meet strict thresholds to ensure reliable guidance for network optimization without degrading user experience.Similarly, the required accuracy level depends on the physical meaning of the metric, its position in the network stack (e.g., AN, CN, or application layer), and the specific control or optimization actions associated with it.
- Computational efficiency should be ensured so that AI modeling and inference can run within the resource constraints of edge or distributed environments.

7.12.4 Use case specific figures

This clause includes the figures used to describe the use case.

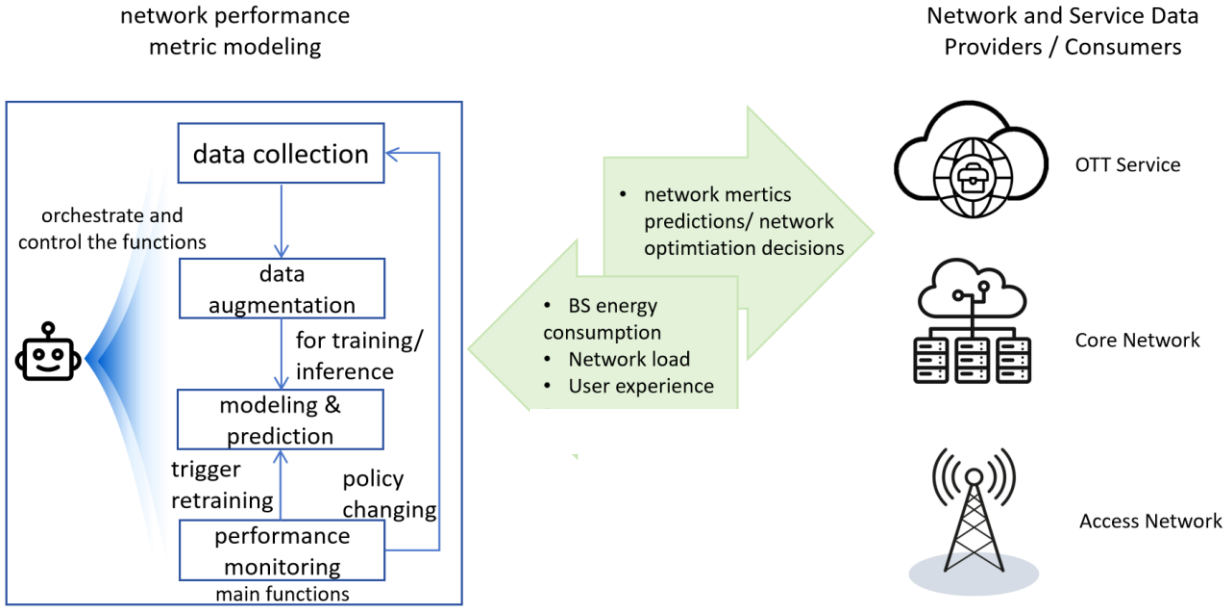


Fig 7.12-1 An overview of the use case

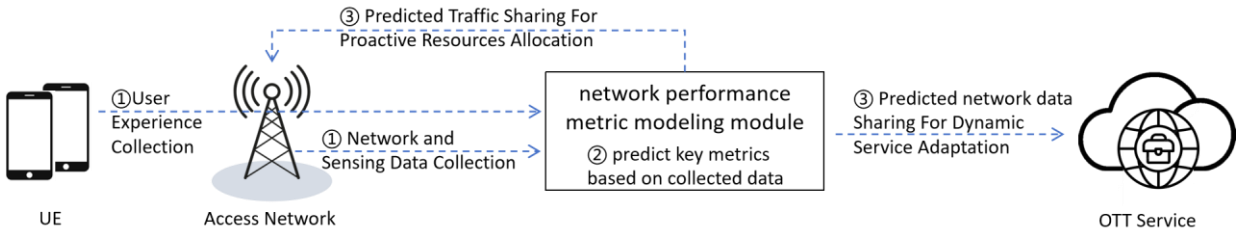


Fig 7.12-2. An example of data interaction in the use case

7.13 Use case 013: Cooperative Communication in Maritime AI-native Networks

With the advancement of autonomous maritime technologies, traditional communication systems face new demands. Conventional networks lack the ability to interpret multifaceted operational directives or fuse large volumes of sensor data, resulting in inefficient coordination and slower decision-making on the water. The introduction of AI into the network core enables dynamic intent detection, multimodal data processing, and intelligent switching between available communication channels critical for robust performance amid fluctuating conditions at sea.

7.13.1 Use case summary

Use case id	92-FG-AINN-I-WG2-019-R5

Use case name	Use Case on - Cooperative Communication in Maritime AI-native Networks
Created by	Aswini Krishnan, R I Minu, Angayarkanni V
Creation date	24.08.2025
Use case context	FG-AINN
Use case description	The proposed system supports seamless communication between underwater vehicles (ROV/USV/USN) and a land base station via ship and UAV relays. Key maritime operations such as underwater infrastructure inspection, search and rescue coordination, environmental data collection, and autonomous vessel control demand ultra-reliable, low-latency communication over highly dynamic and unpredictable channels. AI-enabled real-time channel estimation and switching facilitate optimal throughput and link reliability by dynamically adapting to rapidly changing environmental and mobility conditions. Utilizing 6G network capabilities, the system integrates heterogeneous underwater, aerial, and terrestrial links for multi-hop communication optimized by AI at multiple layers of the protocol stack. The service flow of this use case is as follows: 1. Communication request begins – A vessel starts a session such as navigation updates, distress signals, or crew connectivity. 2. Information gathering – Ships and nearby platforms collect data including vessel position, movement, link quality, and traffic type. 3. Data processing – Collected inputs (location, sensor data, channel conditions, and traffic demand) are organized into a usable format. 4. Analysis and decision-making – The system evaluates the processed information to determine link reliability, select relay nodes, and assign traffic priorities. 5. Relay selection and resource allocation – Suitable vessels or platforms are chosen to forward messages, while spectrum and power resources are distributed effectively. 6. Scheduling enforcement – Communication schedules are applied across ship-to-ship, ship-to-buoy, and ship-to-shore links to ensure smooth operation. 7. Dynamic adaptation – Routes, priorities, and resource use are updated whenever vessel mobility, sea state, or traffic patterns change. 8. Feedback loop – Performance indicators such as latency, reliability, and throughput are monitored to improve future communication decisions.
Core concepts / characteristics of AINN	• AI-native Integration: Embedded AI models performing real-time channel estimation, environment prediction, and protocol adaptation at physical, data link, and network layers.

	• Adaptive and Resilient: Continuously self-optimizing via AI techniques for dynamic channel switching in complex maritime environments. • Multi-layer Intelligence: Cross-layer AI designs enabling joint optimization of channel estimation, routing, error correction, and resource allocation. • Domain-specific AI Models: Custom deep learning architectures trained on maritime aquatic and aerial channel datasets for heightened accuracy. • Decentralized AI Coordination: Use of federated learning or distributed AI for privacy-preserving model updates across heterogeneous nodes (underwater vehicles, UAVs, and base stations).
Actors and Roles (Optional)	● Underwater Vehicles (ROV/USV/USN): Sense and collect underwater data, transmit via acoustic channels with embedded AI channel estimation. ● Ship (Surface Node): Acts as communication gateway between underwater and aerial layers. ● UAV (Aerial Relay): Provides high-bandwidth, low-latency data relaying to land base station; supports real-time AI channel state prediction. ● Base Station (Terrestrial Node): Central control and coordination hub performing higher-layer AI analytics and network orchestration. ● AI Agents: Embedded at multiple network nodes to facilitate local and global real-time decisions on channel estimation, switching, and routing. The use case architecture is depicted in Fig. 2.
Open issues (Technical challenges or Gaps) (Optional)	• **Harsh and Dynamic Channel Conditions:** Underwater acoustic channels suffer from multipath, Doppler shift, and noise; aerial channels exhibit high mobility-induced fading. • **Real-time Adaptive Channel Estimation:** Need for lightweight AI models capable of accurate, low-latency channel state prediction under limited training data. • **Protocol and AI Co-Design:** Integration of AI across PHY, MAC, and Network layers with seamless coordination for real-time channel switching. • **Heterogeneous Network Coordination:** Managing seamless multi-hop communication across underwater, aerial, and terrestrial links with diverse physical characteristics. • **Energy Efficiency and Privacy:** Balancing AI computational cost and network energy budgets, ensuring secure data sharing via federated learning or distributed AI.

	• **Standardization Gaps:** Lack of unified AI-native protocol standards tailored for maritime and underwater communications within 6G networks.
Use case category	To be included at a later stage when the families of use cases are defined
Reference	1. Khichar, S., Santipach, W., Wuttisittikulkij, L., Parnianifard, A., & Chaudhary, S. (2024). Efficient channel estimation in OFDM systems using a fast super-resolution CNN model. Journal of Sensor and Actuator Networks, 13(5), 55. 2. Cui, X., Yuan, B., Li, J., Jiang, B., Li, S., & Liu, J. (2024). Channel estimation for underwater acoustic OFDM based on super-resolution network. Internet Technology Letters, 7(6), e496. 3. Mohamed, E. M., Alnakhli, M. A., & Hashima, S. (2025). Delay-Doppler-Based Joint mmWave Beamforming and UAV Selection in Multi-UAV-Assisted Vehicular Communications. Aerospace, 12(9), 757. 4. Umoga, U. J., Sodiya, E. O., Ugwuanyi, E. D., Jacks, B. S., Lottu, O. A., Daraojimba, O. D., & Obaigbena, A. (2024). Exploring the potential of AI-driven optimization in enhancing network performance and efficiency. Magna Scientia Advanced Research and Reviews, 10(1), 368-378.

7.13.2 Assumptions

In this use case, maritime devices such as vessels, buoys, and unmanned platforms can share real-time context including location, mobility, and channel conditions. A cooperative communication framework is available to process this context along with network status for relay selection and traffic handling. The maritime network can enforce prioritization of critical communication (e.g., distress alerts, navigation data), and performance metrics are continuously monitored to refine cooperative strategies.

7.13.3 Use case requirements

a. Functional requirements

The functional requirements are categorized into:

I. AI-native maritime network functional requirements

II. AI/ML model (LLM-like intelligence) functional requirements

I. AI-native telecommunication network functional requirements:

- The system shall continuously monitor communication resource utilization, including bandwidth, power allocation, and relay node availability across underwater, aerial, and terrestrial segments.
- The system shall monitor and evaluate network performance metrics such as latency, throughput, link reliability, outage probability, and channel estimation accuracy in real time.
- The system shall support intelligent prioritization of traffic, ensuring that critical maritime services (e.g., distress signals, navigation control) are served with higher priority over non-critical data.
- The system shall implement AI-driven relay selection mechanisms that dynamically choose optimal intermediate nodes (ships, UAVs, buoys) based on channel conditions, mobility, and network load.

- The system shall provide adaptive channel selection and switching, enabling seamless transition between acoustic, RF, satellite, and UAV-assisted links depending on environmental and network conditions.
- The system shall support real-time channel estimation and prediction using AI models to improve communication reliability in highly dynamic maritime environments.
- The system shall enable cross-layer optimization, where decisions at the physical, MAC, and network layers are jointly optimized using AI techniques.
- The system shall continuously monitor and optimize network workload, including traffic distribution, relay usage, and spectrum allocation.
- The system shall expose network state information (e.g., QoS requirements, channel conditions, node status) to AI agents for intelligent decision-making.
- The system shall maintain a knowledge base of historical network data, including channel statistics, mobility patterns, and performance metrics for improved prediction and learning.
- The system shall support data lifecycle management, including data collection, pre-processing, transmission, storage, and utilization for AI model training and inference.
- The system shall enable multi-source data collection from maritime sensors (AIS, radar, GPS, environmental sensors) with different time granularities.
- The system shall support secure data processing mechanisms, ensuring data quality, integrity, privacy, and protection against unauthorized access.
- The system shall ensure that user and operational data handling complies with regulatory and privacy requirements.

II. AI/ML Model Functional Requirements:

- The system shall support AI models capable of predictive analysis, including vessel trajectory prediction and channel state forecasting.
- The system shall ensure that AI models produce outputs in structured and interpretable formats for network decision-making.
- The system shall support multi-modal data inputs, including sensor data, channel state information, mobility data, and traffic characteristics.
- The system shall enable mathematical and logical reasoning capabilities within AI models for decision-making tasks such as relay selection and resource allocation.
- The system shall support integration with external modules/tools, such as simulation engines, optimization algorithms, or network controllers, to enhance decision-making.
- The system shall allow continuous learning and model updates, including federated or distributed learning across maritime nodes.
- The system shall support real-time inference with low computational overhead, suitable for deployment on resource-constrained maritime devices.

b. Performance requirements

- The system shall achieve low outage probability and high link reliability across varying channel conditions by utilizing AI-based adaptive transmission and predictive channel estimation, as demonstrated in the system performance analysis.
- The system shall ensure high decision accuracy in relay selection by employing AI-driven predictive models, thereby reducing relay mis-selection probability and improving cooperative communication efficiency.

- The system shall maintain low communication latency with real-time adaptability by enabling proactive decision-making, dynamic channel switching, and optimized scheduling in highly dynamic maritime environments.

c. Additional requirements

- The system shall maintain a centralized or distributed repository for AI models, enabling storage, update, and deployment of models used for channel estimation, relay selection, and decision-making.
- The system shall maintain a data repository for maritime communication and sensor data, supporting storage and reuse of multimodal data collected from underwater, aerial, and terrestrial nodes.
- The system shall support a knowledge base for AI-driven decision-making, enabling improved prediction accuracy and adaptive optimization based on historical network behavior.

7.13.4 Use case specific figures

The Figure 7.13-1 and 2 represent the overall architecture framework of proposed system.



Figure 7.13-1: AI Native Cooperation Communication across OSI Layer

- Physical Layer (PHY): LSTM-based channel prediction and AI-driven multimodal fusion to dynamically estimate channel conditions and enable adaptive modulation and intelligent channel switching before link degradation occurs.
- Data Link Layer (MAC): System uses AI-driven traffic prioritization and predictive relay selection to dynamically schedule transmissions and allocate spectrum, ensuring low-latency delivery for safety-critical maritime communication.

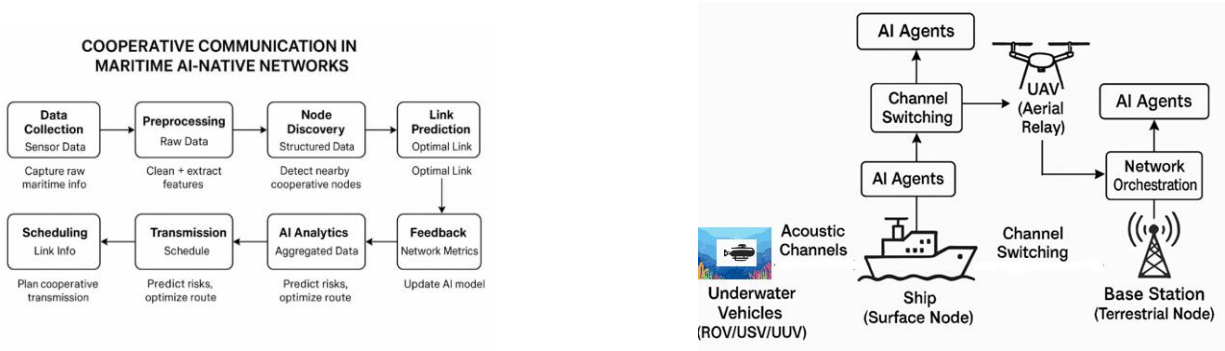


Figure 7.13-2: System for use case architecture

Fig. 2 illustrates the workflow of cooperative communication in maritime AI-native networks, where multiple heterogeneous nodes collaborate to ensure reliable connectivity.

a) Data Collection (Sensor Data)

Collects maritime environmental, navigational, and communication data using onboard sensors. Provides raw multimodal inputs such as AIS, radar, LIDAR, weather, and channel-state information.

b) Preprocessing (Raw Data)

Cleans noisy sensor inputs and extracts relevant features for communication and prediction tasks. Transforms heterogeneous data into structured and normalized formats suitable for AI models.

c) Node Discovery (Structured Data)

Identifies nearby cooperative vessels, buoys, or relay nodes using processed signal and location data. Enables topology awareness by estimating neighbour availability, mobility, and link-state conditions.

d) Link Prediction (Optimal Link)

AI models forecast the best communication link based on channel quality, mobility patterns, and interference. Outputs optimal routing/relay decisions to maximize reliability and minimize latency.

e) Feedback (Network Metrics)

Collects real-time performance metrics such as SNR, throughput, delay, and packet error rates. Continuously updates and retrains AI models to improve future link prediction accuracy.

f) AI Analytics (Aggregated Data)

Analyzes combined multi-node data to assess network risk, congestion, and environmental uncertainties. Supports adaptive decision-making for routing, relay selection, and resource allocation.

g) Transmission (Schedule)

Executes data transmission based on AI-optimized schedules and selected cooperative links. Ensures efficient routing while minimizing energy consumption and communication delay.

h) Scheduling (Link Info)

Plans cooperative transmission timing and resource allocation for all participating maritime nodes. Optimizes channel usage and avoids collisions using dynamic link-quality–aware scheduling.

- Underwater vehicles (such as ROVs, USVs, or UUVs) collect and transmit data using acoustic channels.
- Surface ships act as intermediate nodes, equipped with agents that process incoming signals and perform channel switching for better connectivity.
- Aerial relays (UAVs) extend coverage, enabling communication between ships, underwater vehicles, and terrestrial networks when direct links are weak or unavailable.

- Base stations (terrestrial nodes) coordinate overall network orchestration, ensuring end-to-end connectivity and traffic management.

Each element (underwater, surface, aerial, and terrestrial) is supported by intelligent agents that monitor link conditions, manage switching, and enforce communication policies. This cooperative setup allows maritime networks to maintain seamless communication even under challenging conditions such as mobility, interference, or harsh weather. The overall network communication architecture is illustrated in Fig.3.



Figure 7.13-3: illustrates Network Communication Architecture

7.13.5 Bibliography

[1] International Telecommunication Union, "Focus Group on AI-Native Networks (FG-AINN): Use Cases and Architecture Framework," ITU-T, Ongoing.
[2] International Telecommunication Union, "Architectural Framework for Machine Learning in Future Networks Including IMT-2020," ITU-T Recommendation Y.3172, 2018.
[3] International Telecommunication Union, "Framework for Evaluating Intelligence Level of Future Networks," ITU-T Recommendation Y.3176, 2021.
[4] International Telecommunication Union, "Trustworthiness Evaluation for IMT-2020 and Beyond with Autonomous Networks," ITU-T Recommendation Y.3062, 2025.
[5] H. Kaushal and G. Kaddoum, "Underwater Optical Wireless Communication," *IEEE Access*, vol. 4, pp. 1518–1547, 2016.
[6] Z. Zeng *et al.*, "A Survey of Underwater Optical Wireless Communications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 204–238, 2017.
[7] X. Cui *et al.*, "Channel Estimation for Underwater Acoustic OFDM Based on Super-Resolution Network," *Internet Technology Letters*, 2024.
[8] E. M. Mohamed *et al.*, "Joint UAV Selection and Beamforming in Multi-UAV-Assisted Communications," *Aerospace*, vol. 12, no. 9, 2025.
[9] U. J. Umoga *et al.*, "AI-Driven Optimization for Network Performance Enhancement," *Magna Scientia Advanced Research and Reviews*, 2024.
[10] 3GPP, "Study on Integrated Sensing and Communication (ISAC) for 6G Systems," 3GPP TR, 2024.

Annex: Mathematical Modelling:

Transmitted Signal

$$S = \sqrt{\alpha_c P} x_c + \sum_{k=1}^{K} \sqrt{\alpha_k P}\, x_k$$

Here, P stands for the transmitted power at the transmitted side, S for the superimposed signal that is transferred from the BS, and x_c and x_k are the common and private part. The power allocations for the

divided message part is represented as α_c and α_k with the condition $\alpha_c + \sum_{k=1}^{K} \alpha_k = 1$. The signal transmitted from the S (BS) to IRS is given by,
Received signal (End to End),

$$y_{S-D} = h_1 g_1^H \varphi_1 S + n$$

Where, h_1 is the channel coefficient and n indicates the noise.
The overall SINR for common and private is given by,

$$\gamma_c^{S-D} = \frac{\alpha_c |h_1 g_1^H \varphi_1|^2}{\sum_{k=1\ k\neq k_1}^{K} \alpha_k |h_1 g_1^H \varphi_1|^2 + \frac{1}{\rho}}$$

$$\gamma_k^{S-D} = \frac{\alpha_k |h_1 g_1^H \varphi_1|^2}{\sum_{k=1\ k\neq k_1}^{K} \alpha_k |h_1 g_1^H \varphi_1|^2 + \frac{1}{\rho}}$$

Results

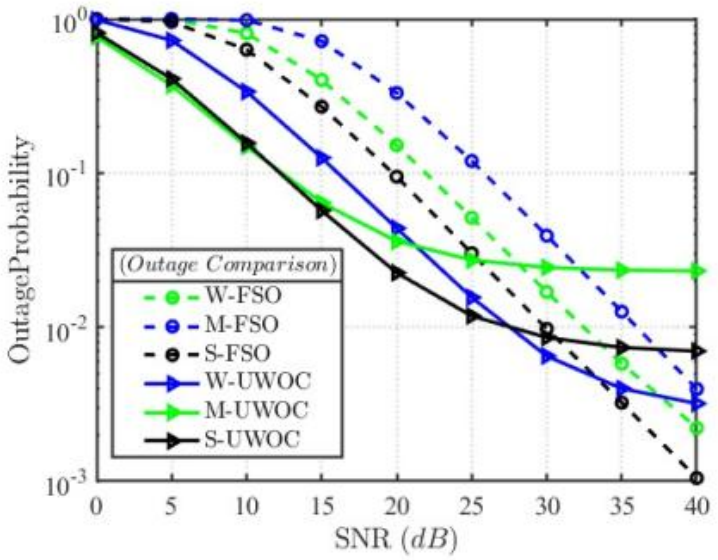


Figure 7.13-4: Outage analysis of the system with FSO and UWOC

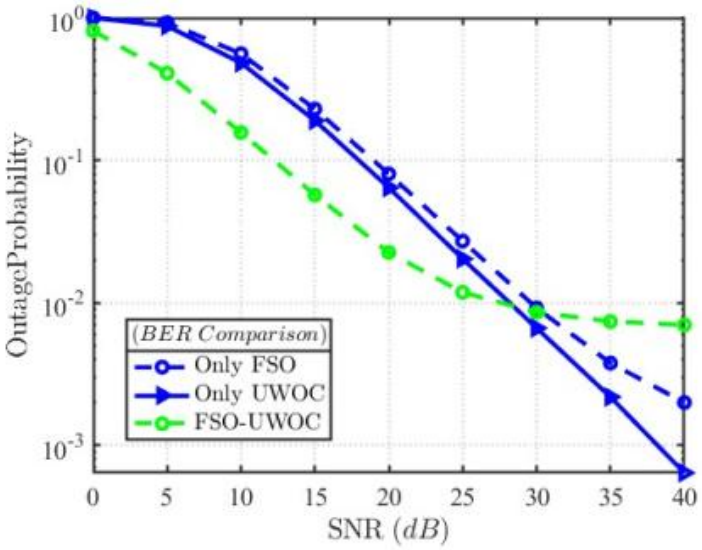


Figure 7.13-5: Outage analysis of mixed FSO-UWOC

The **Figure 7.13- 6** illustrates the outage probability versus SNR for a conventional full-power transmission scheme and AI-based adaptive schemes with different values of the control parameter α\alphaα (0.3, 0.6, and 0.9). For all schemes, the outage probability decreases monotonically as SNR increases, confirming the expected improvement in link reliability at higher SNR.

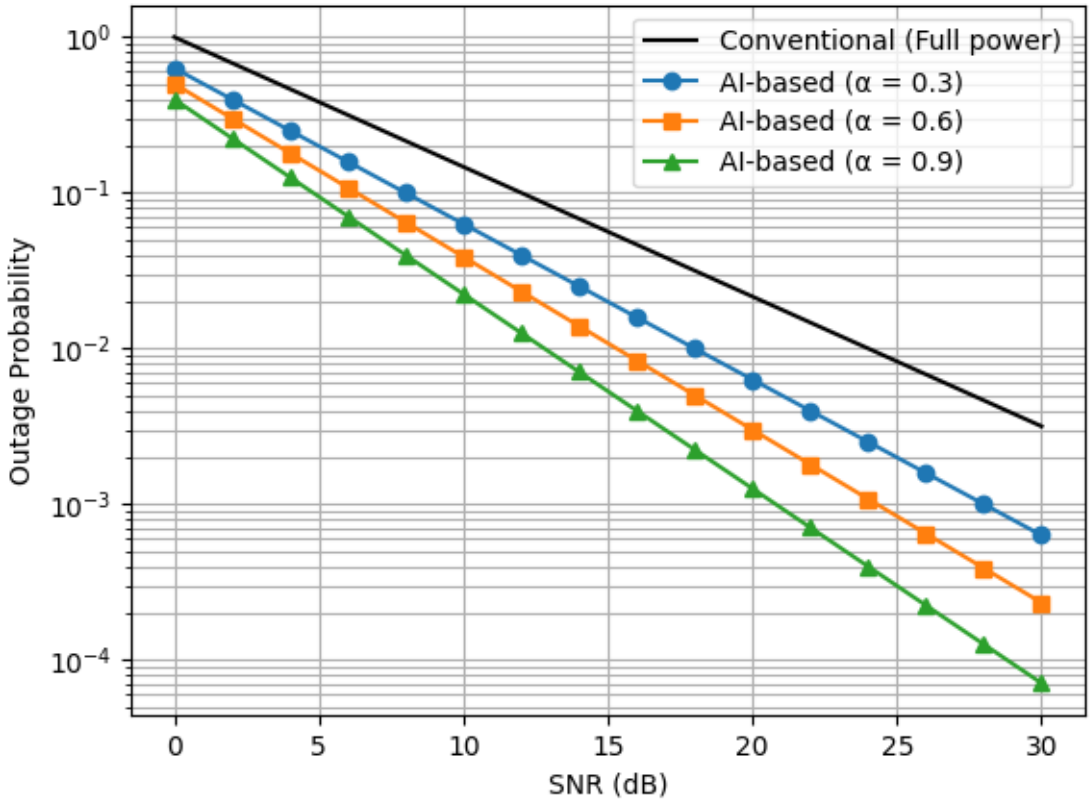


Figure 7.13-6: Comparison of conventional and AI based outage probability analysis

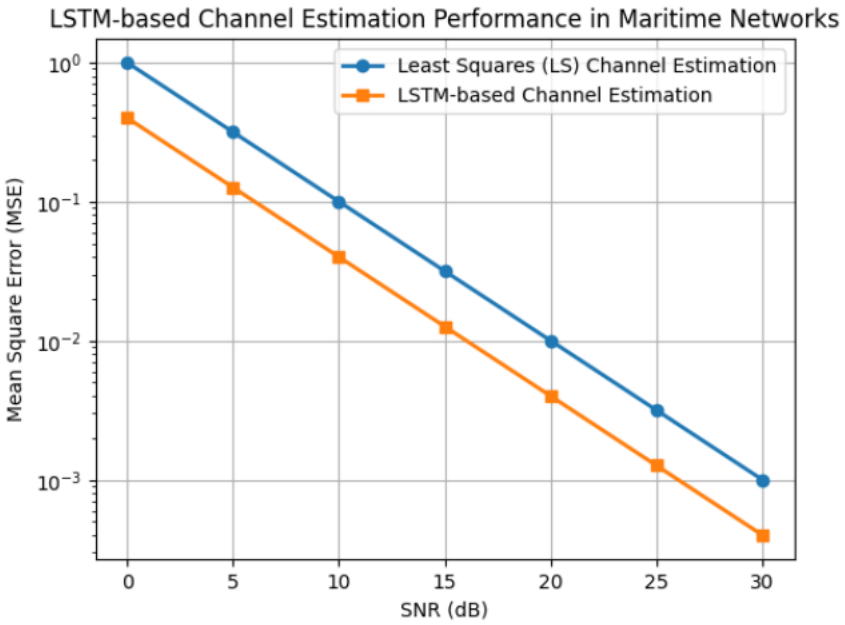


Figure 7.13-7: LSTM based channel estimation performance in maritime analysis

Figure 7.13-7 demonstrates that the proposed LSTM-based channel estimator significantly outperforms the conventional Least Squares (LS) estimation method across all SNR values. As the SNR increases, the Mean Square Error (MSE) of both techniques decreases; however, the LSTM model achieves a much lower MSE, particularly in low and moderate SNR regions.

This improvement is due to the LSTM network's ability to learn temporal channel variations and capture long-term dependencies in time-varying maritime fading channels. Unlike LS estimation, which relies purely on instantaneous pilot information, LSTM utilizes sequential learning to predict and refine channel states more accurately.

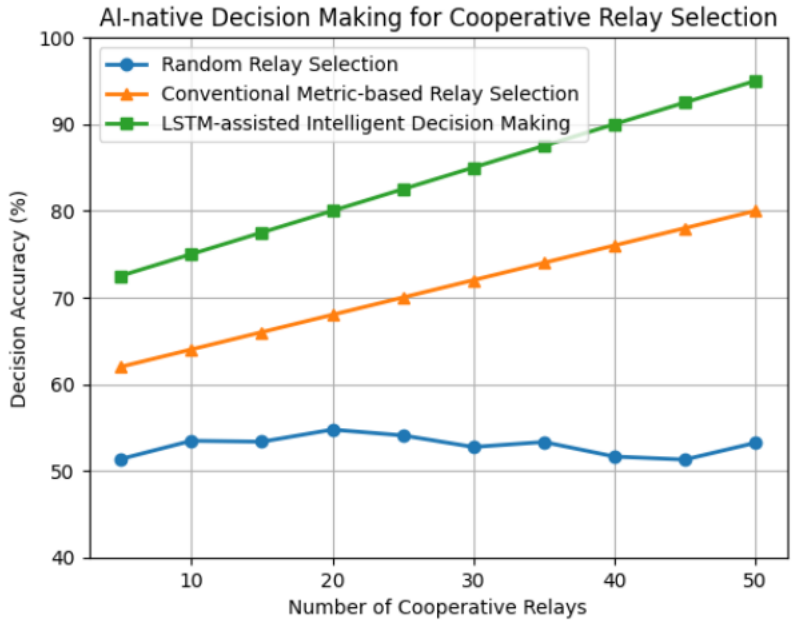


Figure 7.13-8: AI native decision making for cooperative relay selection

Figure 7.13-8 presents the relay selection performance under different decision strategies, including random relay selection, conventional metric-based selection, and the proposed LSTM-assisted AI-native decision model. The results indicate that as the number of available relays increases, the AI-based approach achieves significantly higher decision accuracy compared to conventional methods. Random selection fails to consider channel conditions, resulting in lower reliability, while metric-based selection relies only on instantaneous channel information and cannot capture dynamic variations effectively. In contrast, the LSTM model learns complex channel behaviour and relay performance patterns over time, enabling optimal relay selection under varying SNR conditions. Consequently, the AI-native decision mechanism reduces mis-selection probability and improves cooperative efficiency, leading to enhanced system reliability and performance. These findings demonstrate the effectiveness of integrating deep learning techniques, specifically LSTM, into relay decision frameworks for intelligent maritime communication networks.

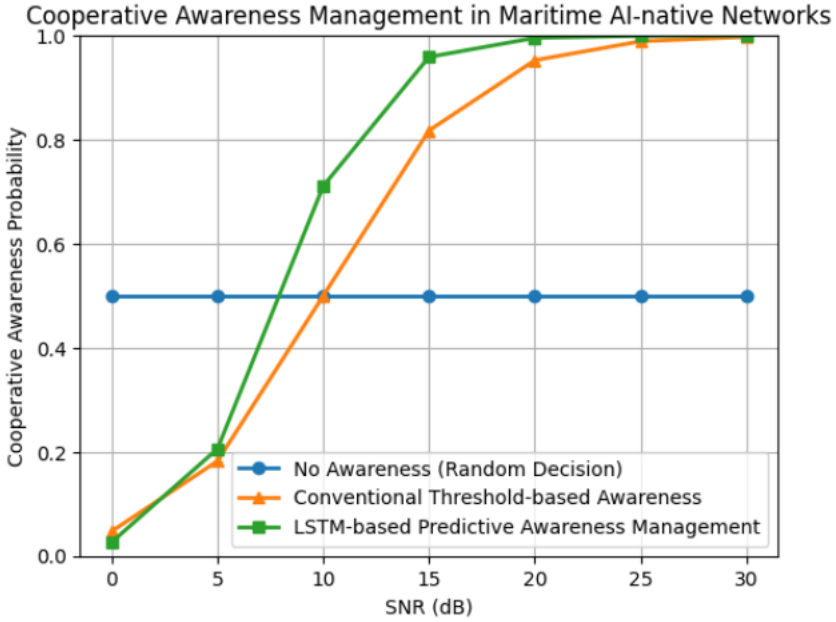


Fig. 8 Cooperative awareness management in maritime network

Fig. 8 illustrates the performance of cooperative awareness management in the maritime network environment. The results compare the conventional threshold-based cooperation mechanism with the proposed LSTM-based predictive awareness model. It can be observed that the conventional scheme activates cooperation only after the channel quality degrades below a predefined SNR threshold, which may result in delayed response and temporary performance degradation. In contrast, the LSTM-based model predicts future channel conditions by learning temporal variations in the maritime fading environment and enables proactive cooperation activation. As the SNR increases, the cooperative activation probability improves smoothly under the AI-driven scheme, demonstrating better stability and reliability. The predictive capability of the LSTM model ensures timely relay engagement, thereby reducing outage probability and enhancing overall link reliability. These results confirm that AI-based cooperative awareness management significantly improves decision efficiency and network robustness compared to traditional reactive approaches.

7.14 Use case 014: AI-Native Task-Oriented Robotic Teleoperation

Emerging applications such as remote robotic teleoperation, industrial automation, and tactile internet services demand ultra-low latency, ultra-high reliability, and continuous availability [1]. Conventional networks treat AI as an add-on (e.g., traffic prediction or offline optimization), which is insufficient

when network impairments directly affect physical safety and task success [2]. This use case motivates AI-native telecommunication networks, where intelligence is embedded end-to-end to enable resilient, task-aware robotic teleoperation under dynamic network conditions.

7.14.1 Use case summary

Use case id	FG-AINN-usecase-013
Use case name	AI-Native Task-Oriented Robotic Teleoperation
Created by	Burak Kizilkaya, Olaoluwa Popoola, Shuja Ansari, Yusuf Sambo, Muhammad Imran from University of Glasgow
Creation date	06/02/2026 (Revised on 19/03/2026 and 23/04/2026)
Use case context	Submitted to the 29th WG2 meeting (05/03/2026), approved in the 34th WG2 meeting (23/04/2026).
Use case description	This document focuses on the capabilities and services of an AI-native telecommunication network required to support task-oriented robotic teleoperation. The network is designed as an intelligent, service-centric platform that integrates communication, computation, and control to enable real-time, reliable, and adaptive operation under dynamic conditions. Rather than treating teleoperation as the primary subject, this use case serves to illustrate how advanced network functionalities such as predictive resource management, task-aware traffic handling, and distributed AI execution are designed as native services to applications. The AI-native network provides a set of tightly integrated services, including real-time network state monitoring, task-context awareness, predictive performance modelling, and autonomous adaptation. These services operate across cloud, edge, and device domains, enabling the network to continuously evaluate both communication conditions and service-level requirements. Teleoperation is used here as an example application that uses these services to demonstrate how network-native intelligence can directly influence application performance and safety. The core actors include the human operator, the teleoperated robotic system equipped with multimodal sensors and actuators, and an AI-native communication infrastructure that tightly integrates communication, computing, and control [3-5]. The network continuously observes its own state (self-awareness) (e.g., latency, jitter, packet loss, congestion) and reasons about task context (task-awareness) (e.g., control stability, operator intent, safety risk), predicts future conditions (i.e., predictive intelligence) and autonomously adapts its behaviour (autonomous adaptation). The use case targets scenarios where network performance directly affects physical task execution and safety, such as remote healthcare procedures and industrial manipulation in hazardous or inaccessible environments [3-5]. In such scenarios, conventional networks that treat AI as an overlay are insufficient, as delayed or degraded communication can lead to instability, task failure, or unsafe robot behaviour. In an AI-native approach, the network supports task-aware traffic handling, predictive communication, and intelligent mode switching between teleoperation and autonomous execution. When network conditions degrade or violations of task-specific latency or reliability constraints are anticipated, the AI-native network autonomously

	coordinates mitigation actions, such as adaptive resource allocation, edge-based prediction, control abstraction, or temporary transitions to local autonomy, while maintaining transparency to the human operator. The AI-native network capabilities demonstrated in this use case are fundamental to its operation. In particular, the use case relies on several core AI-native functionalities, including: • continuous network self-awareness and introspection, • task-aware and semantics-driven traffic handling, • predictive and proactive network behaviour, and • autonomous adaptation across communication, computation, and control domains. These capabilities are tightly integrated and collectively enable the network to maintain service continuity, stability, and safety under dynamic conditions. The following sections elaborate these characteristics in a structured manner. The purpose of this use case is to highlight the service capabilities and architectural implications of AI-native networks, particularly how they enable resilient, low-latency, and high-reliability operation for demanding applications. It emphasizes the role of the network as an active, intelligent system that provides task-aware and predictive services, rather than a passive communication medium.
Core concepts / characteristics of AINN (Optional)	The following AI-native network characteristics are all required to support the robotic teleoperation use case described in this document. Due to the tight coupling between network performance and physical task execution, the absence of any of these capabilities would directly impact system stability, safety, or service continuity. Therefore, the listed characteristics below are considered as interdependent set of characteristics. 1. Native Intelligence as a Network Primitive AI functions are treated as first-class network primitives, comparable to routing, scheduling, or mobility management. Learning, inference, and decision-making capabilities are continuously available within the network and can be invoked in real time to support latency-critical robotic teleoperation. 2. Network Self-Awareness and Introspection The network maintains continuous awareness of, 1) its own operational state (latency, jitter, packet loss, congestion), 2) the service context (teleoperation session state), and 3) the physical task impact (control stability, safety margins). This run-time introspection allows the network to reason about the consequences of communication impairments on robotic task execution, rather than reacting only to QoS metrics. 3. Task-Aware and Semantics-Driven Operation AI-native network explicitly incorporates task semantics into network decision-making. Traffic is differentiated not only by application type but by task criticality, control sensitivity, and safety relevance (e.g., haptic feedback versus video streams). This enables fine-grained prioritization and adaptation aligned with teleoperation requirements. 4. Predictive and Proactive Network Behaviour

	Rather than reacting to performance degradation, the AI-native network predicts future network and control states using learned models. Anticipated latency violations or reliability risks trigger proactive actions such as resource reallocation, predictive buffering, control abstraction, or rerouting, thereby preserving task stability. 5. Autonomous Adaptation and Resilience AI-native network supports closed-loop autonomous adaptation across communication, computation, and control. When network conditions fall outside acceptable task-specific bounds, the network coordinates mitigation strategies, including: • adaptive sampling and encoding, • multi-path or multi-access usage, • edge-assisted prediction, • controlled transitions between teleoperation and autonomous robot operation. 6. Distributed Intelligence Across Cloud-Edge-Device Intelligence is distributed across cloud, edge, and endpoint domains. Latency-critical inference and control decisions are executed at the network edge or device level, while longer-term learning and policy optimization are handled in the cloud. This hierarchical intelligence distribution is essential for real-time teleoperation. 7. Tight Coupling of Communication, Computing, and Control AI-native network removes rigid boundaries between communication, computing, and control. Network resource management decisions are jointly optimized with robotic control objectives and task execution requirements, enabling end-to-end performance optimization beyond traditional layered network designs. 8. Trust, Safety, and Reliability by Design AI-native networks incorporate trust, reliability, and safety mechanisms natively, including continuous monitoring, anomaly detection, and policy enforcement. These mechanisms ensure that AI-driven adaptations remain predictable, explainable, and aligned with safety constraints, which is critical for mission-critical robotic teleoperation.
Actors and Roles (Optional)	Human Operator Role: Controls the teleoperated robotic system to perform skill-based, mission-critical tasks. Responsibilities: • Provides real-time commands and high-level guidance. • Monitors robot state, environment feedback, and system alerts. • Oversees task safety and ensures proper execution. Interaction with AI-Native Network: Receives seamless, low-latency feedback and experiences intelligent adaptation without being burdened by network impairments. Teleoperated Robotic System Role: Executes the tasks using received control commands human, equipped with multimodal sensors and actuators. Responsibilities:

	• Converts operator commands into physical actions. • Collects and streams sensor data (vision, haptics, etc.) to the operator and AI-native network. • Engages autonomous functions when network constraints or safety concerns arise. Interaction with AI-Native Network: Leverages predictive control and mode switching between teleoperation and autonomy. AI-Native Communication Infrastructure Role: Provides intelligent, task-aware network services that integrate communication, computation, and control. Responsibilities: • Continuously monitors network conditions (latency, jitter, packet loss, congestion) and task context (control stability, operator intent, safety risk). • Predicts potential network impairments and proactively adapts resources to preserve task performance. • Coordinates autonomous mitigation strategies such as edge-based prediction, adaptive sampling, multi-path routing, and temporary autonomous control transitions. • Ensures safety, trust, and reliability throughout the teleoperation session. Interaction with Other Actors: Acts as a mediator between human operator and robot, maintaining seamless operation despite network variability.
Technical challenges or Gaps (Optional)	• Network Rigidity and Limited Adaptivity: Conventional communication networks are often designed with static configurations and QoS mechanisms that cannot dynamically adapt to highly variable robotic teleoperation tasks. Latency, jitter, and packet loss directly affect task execution and safety, but traditional networks lack the intelligence to anticipate or mitigate these issues in real time. • Integration of AI with Network and Robot Control: Embedding AI natively into the network requires tight coupling of communication, computation, and control, which is challenging across heterogeneous infrastructures (cloud, edge, devices). Ensuring low-latency inference, distributed learning, and coordination between multiple AI agents remains a significant technical challenge. • Task-Aware Networking: Current networks treat traffic largely by type or priority, rather than by task criticality, control sensitivity, or safety relevance. Mapping task semantics to network actions (e.g., predictive routing, adaptive sampling) is underdeveloped and lacks standardized frameworks. • Predictive and Proactive Adaptation: Predicting network impairments and their impact on robotic tasks in real time requires accurate models of network behaviour, robot dynamics, and human operator intent. Developing robust, real-time predictive mechanisms that can trigger preemptive

	mitigation actions without introducing additional latency is technically challenging. • Autonomous Mode Switching and Resilience: Safely transitioning between teleoperation and autonomous robot control in response to network degradation demands complex coordination and verification mechanisms. Maintaining task stability, safety, and operator transparency during these transitions is not fully addressed by current solutions.
Use case category	AI-native networks applied in vertical industries
Reference	see below

7.14.2 Assumptions

The following assumptions define the operational and architectural context of the use case:

- The communication infrastructure is an AI-Native Network, where AI capabilities are embedded into network control, management, and service entities rather than deployed as external optimization tools.
- AI functions for prediction, decision-making, and adaptation are available across cloud, edge, and device domains, with latency-critical intelligence deployed close to the robotic and operator endpoints.
- The network supports end-to-end observability, enabling continuous monitoring of network performance, service context, and task-level impact.
- Teleoperated robotic systems are equipped with multimodal sensing, actuation, and local control capabilities, and can support hybrid control modes (teleoperation, shared autonomy, local autonomy).
- Task semantics (e.g., control sensitivity, safety relevance, criticality of data streams) can be accessible and interpreted by the network.
- Security, trust, and policy enforcement mechanisms are integrated into the AI-native network to ensure safe and predictable operation of AI-driven adaptations.

7.14.3 Use case requirments

a. Functional requirements

The following functional requirements are all necessary to realize the AI-native capabilities described above in the context of robotic teleoperation. Each requirement directly supports one or more core characteristics and contributes to ensuring stable, safe, and resilient task execution under dynamic network conditions.

- FR1 - AI-Native Network Introspection: The network continuously observes and analyses its own operational state, including latency, jitter, packet loss, congestion, and synchronisation quality, and provide this information to AI decision-making functions in real time.
- FR2 - Task-Aware Traffic Differentiation: The network supports task-aware handling of traffic, allowing differentiation based on task semantics such as control sensitivity, safety relevance, and temporal criticality (e.g., haptic feedback versus video streams).
- FR3 - Predictive Network Operation: The network incorporates predictive models to anticipate performance degradation and task instability, enabling proactive mitigation actions before service-level violations occur.
- FR4 - Autonomous Adaptation and Control: The AI-native network autonomously adapts communication, computing, and routing strategies based on observed and predicted conditions, without requiring manual intervention or static configuration.
- FR5 - Coordinated Communication-Computation-Control: The network supports tight coordination between communication resource management, edge computing allocation, and robotic control objectives, enabling joint optimization across traditionally separate domains.

- FR6 - Intelligent Mode Switching Support: The network supports controlled transitions between teleoperation and autonomous or semi-autonomous robot operation, triggered by AI-driven assessment of network conditions, task state, and prediction confidence.
- FR7 - Distributed AI Execution: The network enables distributed AI execution across cloud, edge, and device nodes, ensuring that latency-critical inference is performed close to the endpoints while longer-term learning and policy updates are handled centrally.
- FR8 - Safety, Trust, and Policy Enforcement: The AI-native network enforces safety and trust policies, including anomaly detection, bounded decision-making, and explainable adaptation behaviour, to ensure reliable operation in mission-critical scenarios.

Collectively, these requirements enable the realization of key AI-native capabilities, including network self-awareness (FR1), task-aware operation (FR2), predictive behaviour (FR3), and autonomous adaptation (FR4–FR6), supported by distributed intelligence (FR7) and trust and safety mechanisms (FR8).

b. Performance requirements

- PR1 - End-to-End Latency: The network supports sub-millisecond end-to-end latency for latency-critical control and haptic feedback loops to ensure stable teleoperation and prevent control oscillations.
- PR2 - Reliability and Availability: The network achieves ultra-high reliability, targeting packet loss rates below 10^{-6} and continuous service availability suitable for mission-critical robotic tasks.
- PR3 - Deterministic Performance under Dynamic Conditions: The network provides predictable and bounded latency and jitter, even under dynamic traffic conditions or partial network degradation, through AI-driven resource management.
- PR4 - Adaptation Responsiveness: AI-driven adaptation mechanisms operate within tight reaction times (on the order of milliseconds) to ensure timely mitigation of predicted or detected performance violations.
- PR5 - Scalability: The AI-native networks support scalable operation, enabling multiple concurrent teleoperation sessions and high device densities without compromising per-task performance guarantees.
- PR6 - Computational Efficiency: The network efficiently utilizes edge and cloud computational resources, minimizing redundant processing while ensuring sufficient capacity for real-time AI inference and control coordination.
- PR7 - Security and Trust Latency: Security and trust mechanisms, including authentication and integrity verification, introduce minimal additional latency and remain compatible with sub-millisecond control requirements.

7.14.4 Use case specific figures

The diagram below illustrates the AI-Native Task-Oriented Robotic Teleoperation use case, where the human operator controls the teleoperated robot while receiving real-time feedback. The AI-Native Network coordinates communication, computation, and control, continuously monitoring network conditions and task context to adapt resources, enable predictive control, and enable autonomous robot operation. Edge and cloud nodes provide latency-critical inference and long-term learning, respectively, ensuring resilient, low-latency, and safe teleoperation.

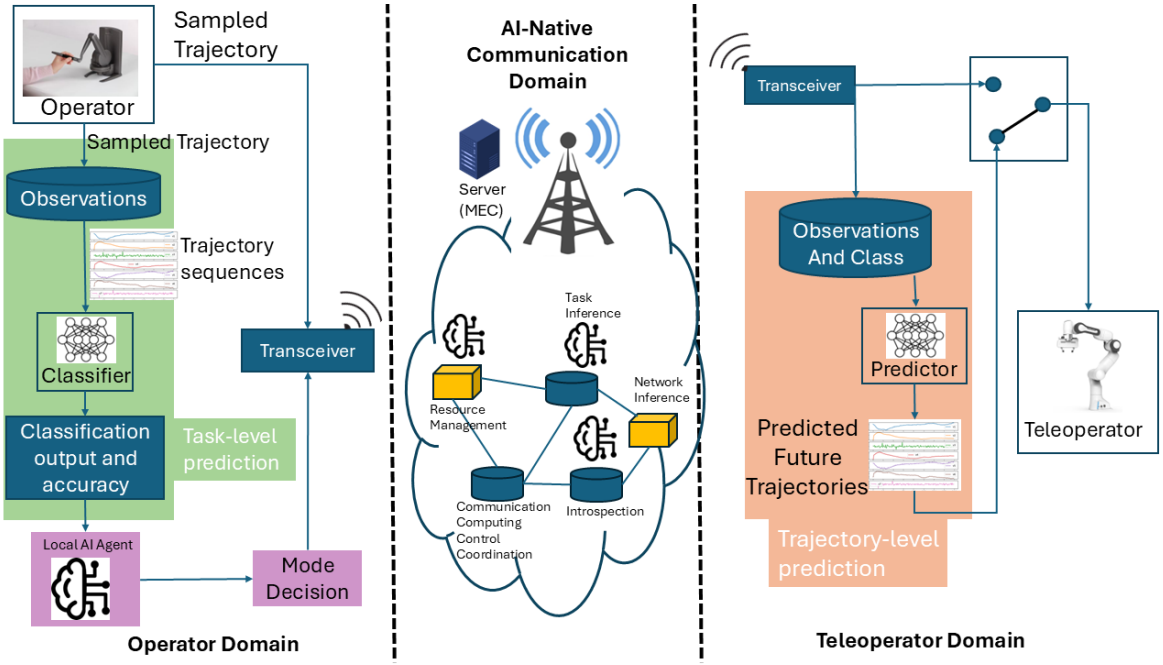


Fig 7.14-1 An overview of AI-Native Task-Oriented Robotic Teleoperation

7.14.5 Bibliography

[1] B. Kizilkaya, G. Zhao, Y. A. Sambo, L. Li and M. A. Imran, "5G-Enabled Education 4.0: Enabling Technologies, Challenges, and Solutions," in IEEE Access, vol. 9, pp. 166962-166969, 2021, doi: 10.1109/ACCESS.2021.3136361.

[2] P. Li, X. Lin and A. Aijaz, "Rethinking Networked Robotics in the 6G Era With Generative AI-in-the-Loop," in IEEE Communications Standards Magazine, doi: 10.1109/MCOMSTD.2025.3620856.

[3] B. Kizilkaya, O. Popoola, G. Zhao, and M. A. Imran, “5G-Based Low-Latency Teleoperation: Two-Way Timeout Approach,” in Towards Autonomous Robotic Systems (TAROS 2023), Springer, 2023, vol. 14136, Lecture Notes in Computer Science, pp. 457–468. doi: 10.1007/978-3-031-43360-3_38.

[4] B. Kizilkaya, C. She, G. Zhao and M. Ali Imran, "Intelligent Mode-switching Framework for Teleoperation," 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 2024, pp. 15692-15698, doi: 10.1109/ICRA57147.2024.10611333.

[5] B. Kizilkaya, C. She, G. Zhao and M. A. Imran, "Task-Oriented Prediction and Communication Co-Design for Haptic Communications," in IEEE Transactions on Vehicular Technology, vol. 72, no. 7, pp. 8987-9001, July 2023, doi: 10.1109/TVT.2023.3247442.

7.15 Use Case 015: Intelligent Transport Systems

National transportation infrastructure is undergoing digital transformation and moving towards the creation, deployment, and operation of digital twins, in a similar manner to their role in networks. This will lead to massive increase of mobile data creation and consumption via the network.

7.15.1 Use Case Summary

Use case id	FG-AINN-usecase-015
Use case name	Sustainable Networks for Intelligent Transport Systems
Created by	Paul Harvey (part of www.transit.ac.uk), Sumiko Miyata, Takamichi Miyata
Creation date	28 August 2025
Use case context	Submitted to FG- AINN

Use case description	In future transportation systems, the amount of data exchanged, as well as the high degree of mobility in the system, will place a significant operational burden on our communication networks. This data exchange with be from both fixed infrastructure assets, as well as mobile assets, and will include a mixture of upload and download traffic. Compared to current IoT traffic or content centric traffic, this presents highly irregular, inconsistent, and asymmetric traffic in different locations in the network. We can consider the following scenario: • The mobile network allocates dynamic slices for ITS traffic (e.g. safety beacons, HD-map updates, OTA, infotainment). • An AI-driven *controller* [4] continuously predicts demand from vehicle density, road events, and historical patterns. • When the controller detects an impending uplink burst (e.g., accident → mass sensor reporting), it pre-emptively scales slice resources, reroutes flows, or throttles non-critical uploads. • Another AI-driven controller, possibly located in a MEC node, classifies ITS data (safety vs. enrichment vs. OTA), applying differentiated traffic treatment. • Operators achieve guaranteed latency for safety services while optimizing backhaul and spectrum use. This can then be composed into the following tasks: Vehicular traffic prediction; Network traffic classification and prioritisation; AI-model drift detection and monitoring; Model retraining; Closed-loop slice assurance and optimisation For each of the above, there are several different AI technologies that can be used in addressing the problem. As different approaches are possible, an AI-native network must be able to accommodate different approaches. Possible AI technologies required for this can include: <u>Vehicular traffic prediction</u> • Time-series forecasting: LSTMs, Temporal Convolutional Networks (TCNs), Transformer-based seq2seq models. • Graph Neural Networks (GNNs): to capture correlations between adjacent cells and road topology. • Hybrid statistical + ML models: for lightweight inference at constrained edge sites. <u>Network traffic classification and prioritisation</u> • Deep packet inspection with ML classifiers (CNNs/RNNs on flow features). • Autoencoders / One-class SVMs for anomaly detection (flagging unseen OTA patterns). • TinyML models optimized for low-power inference on MEC/RAN processors.

	<u>AI-model drift detection and monitoring</u> • Statistical drift detectors (Kolmogorov–Smirnov test, Population Stability Index). • Uncertainty-aware ML (Bayesian Neural Networks, Monte Carlo Dropout). • Online learning algorithms (adaptive random forests, Hoeffding trees) that adapt incrementally. <u>Closed-loop slice assurance and optimisation</u> • Reinforcement Learning (RL) (Deep Q-Networks, Policy Gradient methods) to dynamically allocate slice resources. • Multi-agent RL for coordinating across many cells and slices. • Causal inference models to distinguish correlation vs. causation in SLA violations.
Core concepts / characteristics of AINN (Optional)	- The use of an *AI-based approach* (e.g. stochastic, traditional, agentic) is required to handle the scale, variety, and response required for the content - The use of *customisation*, whereby the AI-based controller in use is customised based on deployment or re-deployment in the network considering: o the hardware resources available (e.g. edge vs core) o The operational context (e.g. passenger vehicles vs commercial) o the regulatory context (e.g. fingerprinting of traffic for entertainment vs regulatory monitoring and compliance) o the performance context (e.g. latency vs power vs hybrid) - The use of *on-demand deployment,* whereby AI-based controllers are not always deployed or are redeployed depending on operational need - The ability to accommodate different AI-models, including their associated metrics, data requirements, and monitoring parameters
Actors and Roles (Optional)	- **Urban Access Network**: Access network nodes serving dense city cells - **Suburban Access Network**: Linear set of access network cells along the suburban highway/on-ramps - **Backhaul Transport Network**: Aggregation and core transport domain that carries slice-isolated ITS flows and enforces low-latency paths for safety traffic.

	- **MEC (Edge Classifier)**: Edge compute platform that terminates ITS traffic locally, runs AI classification, and enforces per-class QoS before sending into the core. - **AI Traffic Classifier @ MEC**: AI function that semantically labels ITS flows (safety, enrichment, OTA) and assigns priority/QoS markings for the underlying slices. - **Network Slice Manager (NSMF)**: Orchestrator that instantiates and reconfigures access network and transport slices - **AI Network Controller:** Instance of an operational controller [y.3061] that uses AI to: o (scenario 1) predicts short-term slice-level demand from KPIs and ITS context and issues network-level resource and routing intents. o (scenario 2 – urban) providing initial predictions and detecting traffic degradation o (scenario 2 – rural) providing low-latency, context-specific predictions for merging traffic - **AI Network Ops**: Instance of an evolution controller [y.3061] that detects model drift, evaluates performance, and triggers retraining or rollout of updated models in AI Network Controller. - **ITS Telematic Feed**: External and application-level data (vehicle density, trajectories, road events) used as contextual input to improve network demand predictions. -
Technical challenges or Gaps (Optional)	- The scale of the proposed challenge in terms of data volume and dynamicity is beyond the capacity of current networks to handle - Current network architectures do not o support on-demand deployment of AI-based controllers o support customisation of AI-based controllers o support the coordination of various AI-based controllers o support the necessary on-demand establishment of data and telemetry sources of AI-based controllers - Integration of control signals or *intent* of intelligent transport systems is not current supported in networks - Integration of third-party data sources that are accessible by the AI-based controllers is not currently supported
Use case category	TBD
Reference	see below

7.15.2 Assumptions

None

7.15.3 Use case requirements

a. Functional requirements

- FR1: AI-Native Vehicular Traffic Prediction: The network shall provide continuous short-term prediction (e.g., 1–300 seconds horizon) of ITS-related network traffic demand per coverage area and per network slice (e.g. safety, HD-map, OTA, infotainment), using inputs such as vehicle density, mobility patterns, road events and historical traffic profiles. Prediction outputs shall be suitable to drive automated slice scaling, routing and scheduling decisions in the underlay network.
- FR2: Slice-Aware ITS Network Traffic Differentiation: The network shall classify ITS flows into at least the following categories: (a) safety-critical (e.g., CAM/DENM, collision warnings, actuator commands), (b) operational enrichment (e.g., HD-map updates, telemetry for digital twins), and (c) non-critical bulk (e.g., OTA updates, infotainment), and bind each category to a corresponding network slice or QoS treatment. Classification shall be possible at different areas of the network (e.g. MEC) based on packet/flow features and auxiliary context.
- FR3: Pre-Emptive Safety Slice Scaling and Protection: When the AI controller predicts an impending safety-related uplink or downlink burst (e.g., accident, hazard zone, emergency braking cluster), the network shall pre-emptively scale the safety slice resources of the network and, if needed, reallocate capacity from non-critical slices while maintaining minimum contracted QoS for those slices. This shall occur before measured KPIs breach configured thresholds.
- FR4: MEC-Assisted Safety Prioritisation: The network shall support MEC nodes that can locally identify safety-critical ITS network traffic and enforce differentiated treatment (e.g., local breakout, queue prioritisation, local aggregation) without requiring every decision to traverse the core network. MEC nodes shall be able to request or adjust local resource allocations of the network in coordination with one or more controllers.
- FR5: Mobility-Aware Controller Redeployment: The network shall support on-demand deployment, redeployment and termination of AI network controllers in different network locations (e.g., core DC, regional DC, MEC site) in response to changes in traffic demand, mobility hotspots, or resource constraints. Migration shall preserve relevant state (e.g., learned policies, slice KPIs, active intents) to avoid operational discontinuity.
- FR6: Model Customisation for Edge Constraints: The network shall support customisation of AI controllers based on the capabilities of the target execution environment, including but not limited to model pruning, compression, quantisation, and feature reduction.
- FR7: Closed-Loop Slice Assurance for ITS Network Services: The network shall implement closed-loop control whereby slice-level KPIs (latency, reliability, throughput, packet error rate) for ITS network services are continuously monitored and fed back to the AI controller. The controller shall adjust slice parameters (e.g., bandwidth, scheduler weights, routing policies) automatically to maintain configured service-level objectives.
- FR8: Online Adaptation and Drift Handling: The network shall support online learning and concept-drift handling for ITS-related AI models (e.g., traffic prediction, classification, anomaly detection), including incremental adaptation techniques that can run under resource constraints across the network. It shall be possible to detect when model behaviour diverges from current network traffic conditions and to trigger retraining or model replacement.
- FR9: Integration of External ITS Context: The network shall be able to ingest external ITS context sources, such as road authority messages, weather alerts, infrastructure sensor feeds, fleet schedules, or event calendars, and use them as additional features for prediction and control decisions in the network. Policies shall allow operators to configure which external sources influence which slices or domains.
- FR10: Intent-Based ITS Network Operation: The network shall allow ITS service providers or operators to express high-level intents (e.g., “maintain sub-10 ms latency for collision warnings in

region X", "complete fleet OTA updates by 03:00 with minimal impact on consumer traffic"), which the AI controllers translate into concrete resource and configuration changes across the access network, transport and MEC.

- FR11: End-to-End Network Coordination: The network shall support coordination of control actions across multiple domains in the network so that ITS-related decisions (e.g., scaling a safety slice) are consistently applied end-to-end, and conflicting actions between domains are avoided or resolved.
- FR12: Safety-Focused Fallback and Graceful Degradation: In cases where AI components or data sources are unavailable, inconsistent, or under suspicion (e.g., security anomaly, model drift beyond thresholds), the network shall provide deterministic fallback behaviours which ensure that safety-related ITS traffic remains protected with conservative resource allocations and routing policies.
- FR13: Explainability and Traceability for ITS Decisions: The network shall provide traceability for key AI-driven control actions affecting ITS network services (e.g., why safety slice capacity was increased or why OTA was throttled), including logging of input features, decision outputs and applied policies, to support troubleshooting, auditing, and regulatory compliance.
- FR14: Support for Multi-Operator / Multi-Tenant ITS: The network shall support multi-tenant ITS environments where different transport operators or service providers share network infrastructure but require logically isolated slices and potentially different AI policies, while still allowing coordinated actions for safety-related use cases when mandated by regulation or mutual agreements.

a. Performance requirements

- PR1: Safety Latency for V2X Messages: For ITS safety-critical messages (e.g., collision warnings, cooperative manoeuvre coordination) within a defined geographic area, the network shall support end-to-end latency not exceeding 10 ms for at least 99.999% of successfully delivered messages under normal operating conditions.
- PR2: Safety Reliability under Burst Conditions: Under predicted or detected safety-related uplink bursts (e.g., mass sensor reporting after an accident), the network shall maintain a packet delivery success rate $\geq$ 99.999% for safety-critical ITS network traffic within the impacted areas of the network, while non-safety slices may experience controlled degradation as per configured policies.
- PR3: Predictive Horizon and Accuracy: The AI network controller shall provide traffic demand forecasts for each relevant slice and coverage area with a minimum horizon of 30 seconds and an update interval of $\leq$ 5 seconds, achieving a mean absolute percentage error (MAPE) of $\leq$ 15% for safety and enrichment slices under typical operating conditions.
- PR4: Reaction Time for Pre-Emptive Scaling: From the moment the controller identifies that a safety slice will violate its configured latency or load thresholds, the network shall be capable of applying pre-emptive scaling or reconfiguration actions (e.g., transport reservation) within $\leq$ 1 second in the affected region.
- PR5: MEC Classification Latency: The end-to-end processing delay introduced by MEC-based traffic classification and prioritisation for ITS network flows (from packet arrival at MEC ingress to decision enforcement at the local scheduler) shall not exceed 10 ms for safety network traffic under nominal MEC load.
- PR6: Controller Migration Time: For planned redeployment of an AI network controller instance (e.g., from core to edge or between regional DCs) while maintaining service continuity, the network shall achieve a switchover time (including state transfer and activation) of $\leq$ 60 seconds for regional moves, with no violation of active safety SLAs.
- PR7: Edge Model Footprint: Pruned/quantised AI controller instances designed for MEC or access network deployment shall preserve at least 90% of the control performance metrics (e.g., prediction accuracy, policy quality) relevant to safety and enrichment slices.
- PR8: Online Adaptation Speed (Hoeffding / Incremental Models)

- For incremental models used at the edge (e.g., Hoeffding-based or similar), the network shall adapt to a sustained change in network mobility or traffic patterns such that the prediction error is reduced by at least 50% from its drifted value within ≤ 30 minutes of continuous new data.
- PR9: Slice Reconfiguration Overhead: The application of a new slice configuration (e.g., adjustment of scheduler weights, QoS profiles) driven by the AI controller shall not introduce additional one-way delay greater than 10 ms for ongoing ITS network safety flows during the reconfiguration window.
- PR10: OTA Traffic Impact Containment: During large-scale OTA update campaigns for vehicles, network actions taken by the AI controller shall ensure that additional latency introduced on safety-critical traffic does not exceed 2 ms compared to baseline, and that average throughput for OTA traffic still allows ≥ 90% of targeted vehicles to complete updates within the configured maintenance window.
- PR11: Monitoring and Telemetry Granularity: The network shall support collection and processing of relevant KPIs (per slice, per MEC node) for AI decision-making with a granularity of ≤ 1 second for safety-related metrics and ≤ 10 seconds for non-safety ITS metrics.
- PR12: Robustness to Data Quality Issues: In scenarios where up to 10% of input telemetry sources (e.g., faulty sensors, missing counters, delayed external feeds) are unavailable or inconsistent, the AI controllers shall still maintain safety slice latency and reliability within 10% of their target values, using fallbacks and redundancy.
- PR13: Multi-Tenant Isolation Efficiency: In multi-tenant ITS scenarios, actions taken to protect the safety slice of one tenant shall not degrade safety KPIs of another tenant by more than 5% relative to their respective configured targets, unless explicitly overridden by cross-tenant safety policies.
- PR14: Logging and Traceability Latency: Logs and decision traces for AI-driven control actions that affect safety-critical ITS slices shall be available to authorised operations and audit systems within ≤ 60 seconds of the action being executed.

7.15.4 Use case specific figures

We consider the following scenarios of this use case.

Scenario 1: Network Centric AI Control During an Urban Accident

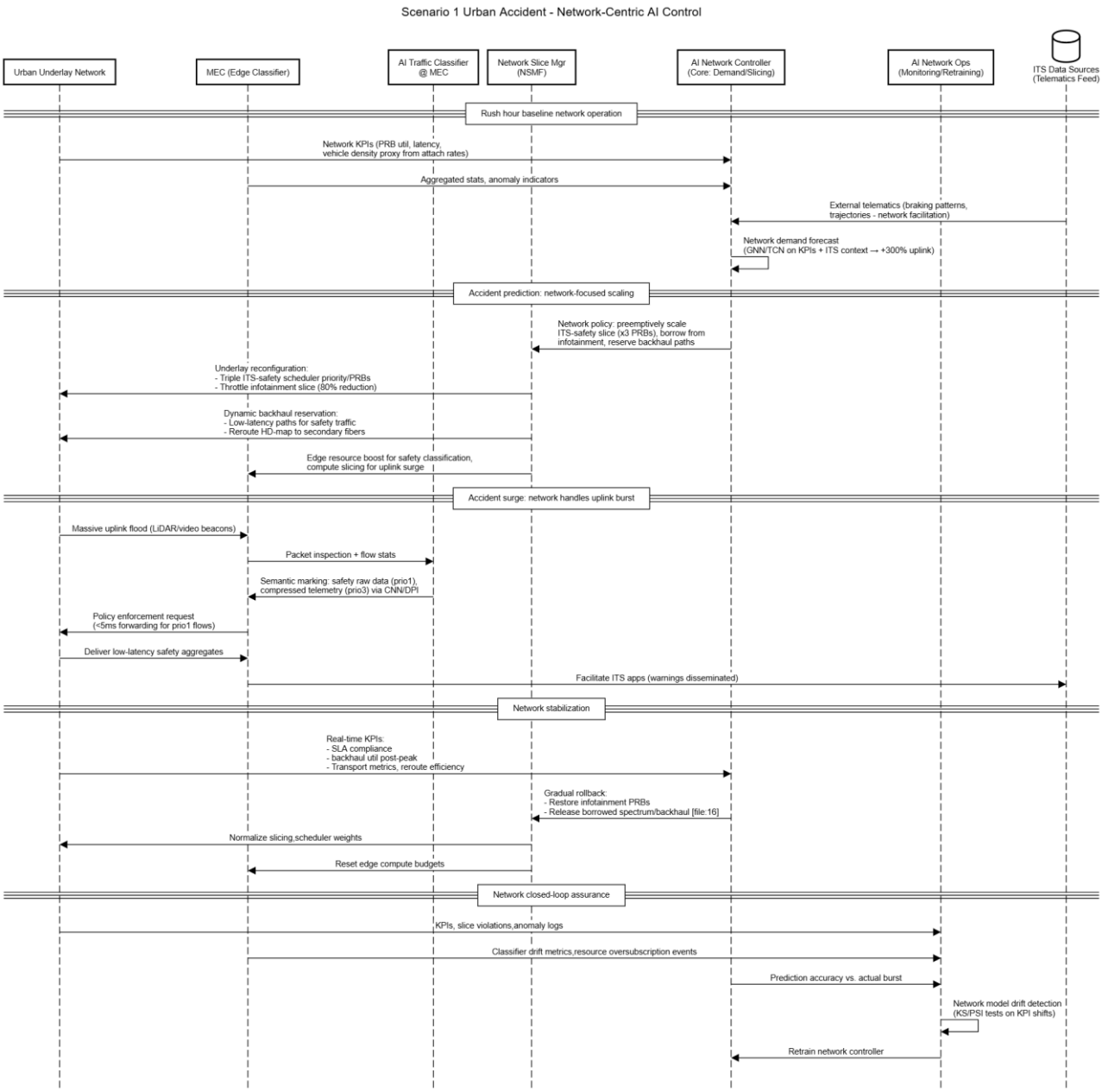


Figure 5: Scenario 1

During rush hour in a dense city center, the AI Network Controller monitors network KPIs (e.g. PRB utilization, attach rates as vehicle density proxy) alongside facilitated ITS telematics feeds showing elevated braking patterns and erratic trajectories. It predicts an imminent multi-vehicle collision 30-60 seconds ahead, forecasting a +300% uplink burst in safety beacons across affected network areas. The controller issues network requests to the NSMF: triple ITS-safety slice capacity by pre-emptively boosting network configuration traffic scheduler priority/PRBs, borrowing capacity from infotainment slices (80% throttle), reserving low-latency backhaul paths. The network prioritise safety bursts using techniques, such as robust FEC (self-correcting parity) and short TTI mini-slots (0.125ms) despite congestion, ensuring <5ms E2E propagation. MEC handles edge classification (DPI+ML on raw LiDAR/video vs compressed telemetry), marking prio1 flows for fast-path forwarding while queuing non-critical uploads. Post-incident, network KPIs and classifier drift metrics trigger rapid retraining of the AI controller, improved future urban burst prediction accuracy using the fresh accident dataset.

Scenario 2: Network AI Controller Redeployment, Customisation, and Retraining

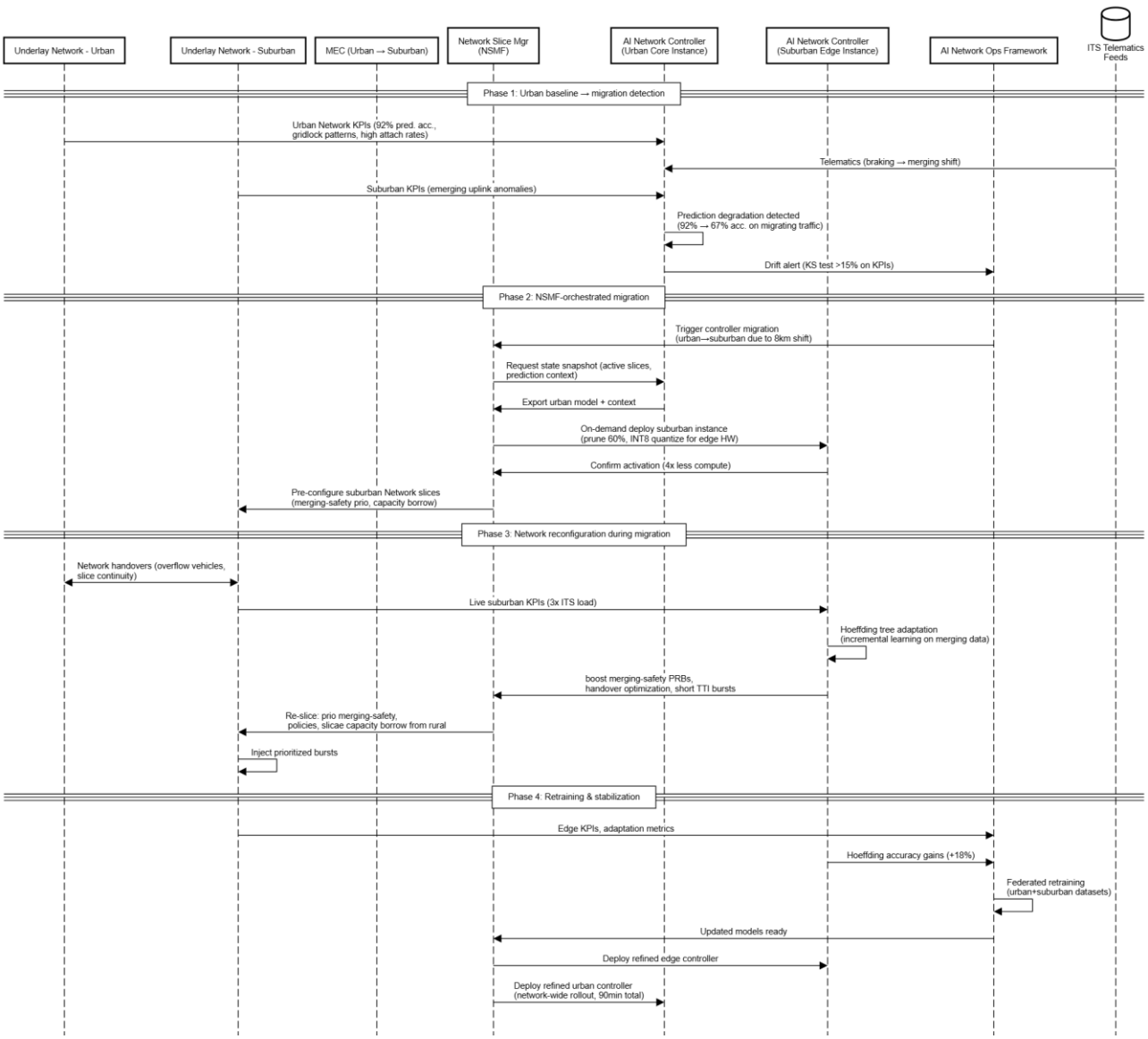


Figure 6: Scenario 2

Morning rush-hour congestion with 2,000+ connected vehicles migrates 8km from city centre to suburban highway over 45 minutes, causing the urban-optimized AI Network Controller's prediction accuracy of short-term resource demand to drop as patterns shift from gridlock to highway merging. The controller is migrated to the suburban transport edge reducing the latency of responses while pruning the core model (e.g. 60%) and quantizing (e.g. to INT8) for resource-constrained suburban network/MEC hardware. The suburban controller re-slices the suburban network for merging-safety prioritization, borrows slice capacity, and optimizes network handovers, while 45-minute mobility data triggers edge adaptation (e.g. Hoeffding tree adaptation) plus core federated retraining. Updated controllers redeploy network-wide, boosting accuracy and reducing SLA violations across migrating cells.

7.15.5 Bibliography

[1] Operational Behavior of a High Definition Map Application White Paper, https://aecc.org/wp-content/uploads/2020/07/Operational_Behavior_of_a_High_Definition_Map_Application.pdf
[2]https://www.racfoundation.org/motoring-faqs/mobility#:~:text=A1)%20At%20the%20end%20of,of%20the%20Second%20World%20War.
[3] B. Varghese et al., "Revisiting the Arguments for Edge Computing Research," in IEEE Internet Computing, vol. 25, no. 5, pp. 36-42, 1 Sept.-Oct. 2021, doi: 10.1109/MIC.2021.3093924.
[4] Y.3061 Autonomous networks - Architecture framework

7.16 Use Case 016: AI-Agent-Driven Intelligent Network

The convergence of communication and computing in network environments necessitates intelligent automation. Current closed-loop automatic network management relies heavily on static configurations and human intervention, limiting adaptability and efficiency. Unlike traditional AI models constrained by rule-based workflows or problem-specific data mining strategies (e.g., reinforcement learning agents), AI agents, leveraging multimodal foundation models, can dynamically monitor, analyze, and optimize network performance. By fully harnessing the logical reasoning and generative capabilities of large foundation models within the AI agent architecture, as well as the problem-solving capabilities of specialized domain knowledge bases, tool libraries, and specialized small AI models, AI agents can achieve greater generalization, flexibility, and expandability compared to traditional AI models which are designed or trained based on specific problems, scenarios, and datasets. Consequently, AI agents demonstrate more generalized intelligence.

Leveraging these capabilities, in the network, various tasks such as resource allocation, task scheduling, fault diagnosis, metric maintenance, and service assurance, spanning from the management and orchestration layer to the infrastructure layer, will gradually require less human intervention. Moreover, it can avoid the clumsy and inflexible AI application paradigm of one - problem - one - development paradigm.

This use case explores AI-agent-driven solutions for network operation and maintenance, joint communication-computation orchestration, and infrastructure management. Implementing AI agents in network management significantly enhances efficiency, autonomy, and scalability, ensuring seamless operation in evolving network scenarios, such as 6G and beyond.

7.16.1 Use Case Summary

Use case id	FG-AINN-usecase- 016
Use case name	AI-Agent-Driven Intelligent Network
Created by	Yixuan Huang, Lehan Wang, Ziqi Chen and Qi Sun from China Mobile
Creation date	04.03.2025
Use case context	Submitted to FG-AINN, and approved in the 36th WG2 meeting (14th May 2026) FG-AINN meeting.
Use case description	The motivation of this use case is to enable intelligent network management through AI agents. Traditional network operation and maintenance require extensive human oversight and are constrained by predefined policies, limiting real-time adaptability. AI agents enhance network intelligence and automation by integrating large network foundation models, specialized small AI models, and feedback loops mechanisms. Large network foundation models can provide basic logical reasoning and generalized analytical decision-making capabilities. While specialized small AI models can provide efficient problem-solving capabilities in specialized areas. The synergy between the two allows the AI agent to combine both multitasking generalization capabilities and domain expertise, thus minimizing the reliance on human intervention in the network management process. Specifically, an possible use case is addressing hierarchical differentiated experience assurance in wireless networks, where traditional solutions face slow package launch and difficult VIP guarantee. AI agents enable intent-driven workflow orchestration for on-demand package delivery and end-to-end assurance, translating user intent to network intent, automating cross-system interfaces, and encapsulating atomic capabilities (e.g., carrier selection) to achieve low-cost, fine-grained optimization. Furthermore, the AI agents can use feedback loops for self-optimization, leading to more experience and stronger problem-solving

	skills during the practice of interacting with the environment. This enhancement can lead to a significant boost in network optimization, maintenance, operation, management and upgrade efficiency, strengthens the network's self-repair ability, and cuts down system downtime. One example of AI agents' applications in network management and orchestration is automated deployment of network functions such as automated instantiated deployment of base stations and automated scaling up and down based on predicted load metrics. An example is infrastructure management troubleshooting and fault diagnosis for cloud platforms.
Core concepts / characteristics of AINN (Optional)	Refers to core concepts or characteristics of AINN applicable to the use case. The terminology will be updated to reflect the work done in WG1.
Actors and Roles (Optional)	AI Agent: Acts as the central orchestrator that interacts with different components. Responsible for executing tasks, making decisions, and responding to requests. Calls tools and models to assist in various network management tasks. Executor: Interacts with network and infrastructure management components. Ensures that decisions made by AI agents are applied effectively. Model Repository: Provides Large Network Foundation Models for general-purpose reasoning. Offers Specialized Small AI Models for task-specific optimizations. Supports functions such as task decomposition, logical reasoning, data analysis, inference & prediction, and decision-making. Memory: Manages various types of data generated in the network. Provides historical and real-time data to AI agent for decision-making. Enables long-term learning and knowledge retention. Tool-box: Contains tools that AI agent can invoke for specific tasks. Validation Module: Ensures that actions taken by AI agent comply with authorization and modification rules. Incorporates authorization (access control), modification (updates and changes), and reflection (feedback mechanisms). Prevents unauthorized or incorrect executions. Network Management & Infrastructure Management: Receive execution outputs from AI agent through the executor. Represent the real-world network and infrastructure where actions are applied. Ensures stability, efficiency, and performance of network operations.
Technical challenges or Gaps (Optional)	Compared to existing automation approaches such as rule-based systems, closed-loop control, and reinforcement learning-based optimization, AI agents introduce several unique capabilities: 1. Generalized reasoning: Unlike task-specific models, AI agents leverage foundation models to handle previously unseen scenarios. 2. Cross-domain orchestration: AI agents can coordinate multiple tools, models, and network functions across layers, breaking the "one-task-one-model" paradigm. 3. Intent-driven operation: AI agents can translate high-level human or service intents into executable workflows, reducing the need for manual configuration. Despite the promising potential of AI agents in network management,

	several challenges and gaps must be addressed to ensure practical deployment and scalability: Task Decomposition: Designing robust environment perception for intent parsing and balanced task planning in wireless domains. Lack of Unified Interaction Interface: AI agents require seamless integration with various tools, data sources, and other AI agents. The absence of a standardized interaction interface complicates interoperability and limits scalability. Task Collaboration: Resolving goal/decision conflicts in distributed multi-agent systems. Data Access and Training Challenges: The training of AI agents depends on high-quality, diverse datasets. Obtaining relevant data for AI model training remains a challenge, especially in the wireless domain where access to proprietary datasets may be restricted. Compute Resource Constraints: Communication networks infrastructure may not be well-suited to support the high computational demands of large-scale AI models. Real-Time Processing Limitations: AI agent-based inference and decision-making models may lack the real-time processing capabilities needed for wireless communication scenarios. Ensuring low-latency, high-reliability AI-driven optimizations is crucial for meeting stringent network performance requirements. Scalability of AI-Agent-Driven Networks: As networks become increasingly complex, managing AI agents at scale may introduces orchestration and resource allocation challenges. Efficient AI agents coordination and workload distribution strategies are necessary to avoid bottlenecks.
Use case category	AI-agent in AI native networks
Reference	None

7.16.2 Assumptions

None

7.16.3 Use case requirements

a. Functional requirements

- AI agents should dynamically monitor and analyze real-time network conditions to optimize performance.
- AI agents should integrate with large foundation models and specialized small models for context-aware decision-making.
- The AI-native network should enable autonomous execution of network optimization actions through coordinated interaction among AI agents, foundation models, specialized AI models, policy management functions, and network control entities under policy and authorization constraints.
- AI agents should facilitate cross-layer collaboration, enabling seamless communication between network elements.

b. Performance requirements

i. Reduction of human intervention in network operation and maintenance by at least 60%.

ii. Improvement of network resource utilization efficiency by 30% through AI-driven dynamic allocation.

iii. Reduction of network downtime and fault resolution time by 30% via AI-based predictive maintenance.

iv. Real-time inference and optimization capabilities with a latency threshold of less than 10 ms for critical decision-making.

v. Adaptively generate, select, and apply complete solutions and tools tailored to different services, scenarios. Each AI agent should be able to solve at least three different problem types.

7.16.4 Use case specific figures

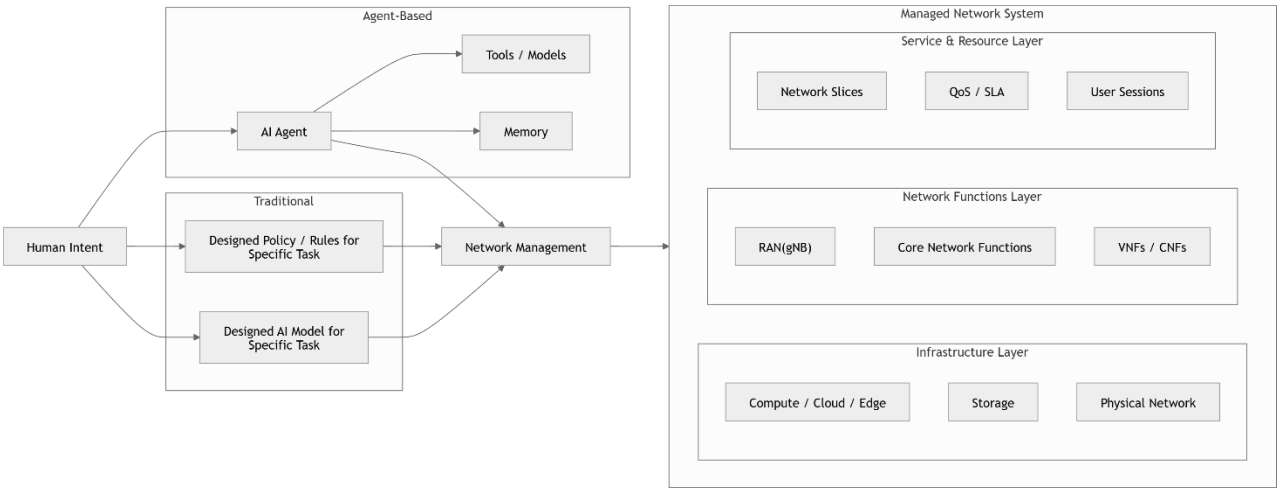


Figure 7.16-1: AI-Agent-Driven Intelligent Network related figure

7.17 Use Case 017: Proactive AN optimization for embodied intelligence

Embodied intelligence refers broadly to intelligent systems that integrate sensing, computation, and execution in a physical form and dynamically interact with the environment, which underpins advanced robotics, self-driving cars, and drones, etc. Unlike traditional AI systems that operate solely on abstract data, embodied intelligence learns and adapts through direct sensorimotor experiences with multi-modal data fusion (e.g., signal from cameras, LiDAR, microphones, force/torque sensors), and thus imposing stringent demands on wireless communication, particularly on AN, due to its reliance on high-throughput and low-latency data transmission, real-time decision-making, and high-reliability connectivity. However, these requirements cannot be fulfilled in current networks with passive scheduling policies, in particular for mobile and edge devices. That is because existing AN mechanisms rely on reactive decisions based on received feedback information, which become outdated due to processing and feedback delays, without the capability to predict future traffic, task, or channel dynamics, thus inherently preventing proactive optimization. In addition, existing data transmission mechanisms designed for low-volume signaling and high-volume, delay-tolerant data are inadequate for multi-modal data with diverse QoS requirements. Moreover, reliable data exchange between devices is critical for distributed or mobile devices to ensure seamless operations (e.g., coordinated control, real-time monitoring, and adaptive task execution), which is also not supported by current networks. To solve these problems, a proactive AN data transmission optimization for embodied intelligence is proposed in this use case. By introducing AI-driven prediction, the network is enabled to infer future traffic patterns, task requirements, and channel conditions, thereby supporting proactive resource allocation and schedulin. It focuses on enabling high-throughput, low-latency multi-modal data transmission and information sharing among embodied intelligent devices to facilitate the seamless operation with the assist of new data bearer introduction and AI prediction.

7.17.1 Use case summary

Use case id	FG-AINN-usecase-017
Use case name	Proactive AN optimization for embodied intelligence
Created by	Yang Wang, Kai Zhang, Lehan Wang, and Jingwen Xu from China Mobile

Creation date	04.03.2025
Use case context	Approved in the 38th WG2 meeting (28th May 2026)
Use case description	The motivation of this use case is to enable AI-driven proactive AN optimization and introduce a new data bearer for embodied intelligence applications. Specifically, a dedicated **AI service data bearer** is introduced to facilitate efficient multi-modal data transmission and delivery in end-to-end or edge-collaborative scenarios, enabling AI service data transmission, processing, and routing operations to be completed within the AN. This AI service data bearer is designed to support high-reliability, high-throughput, and low-latency data transmission and sharing. To distinguish sessions utilizing this new bearer from traditional data bearers, a **Service Type ID** is introduced to indicate whether the session requires the AI service data bearer or a traditional data bearer. A description of the overall procedure is given as follows. First, embodied intelligence UE sends service requests to the AN, including service requirements and a Service Type ID. The AN identifies the Service Type ID and generates resource configuration policies for the AI service data bearer based on the service requirements. The AN continuously monitors the network states, service requirements and device behaviors (e.g., mobility, period of activity), makes predictions, and proactively updates the AI data bearer configurations for embodied intelligence service assurance.
Core concepts / characteristics of AINN (Optional)	AINN meets the stringent requirements of embodied intelligence for reliable, high-throughput, low-latency multi-modal data transmission and delivery in end-to-end or edge-collaborative scenarios by introducing new data bearers, and proactively optimizing AN configurations with predictive analytics from continuous network or service monitoring.
Actors and Roles (Optional)	This describes roles (i.e., functions) involved in the use case and associated actors.
Technical challenges or Gaps (Optional)	In current network systems, data transmission mechanisms designed for low-volume signaling and high-volume, delay-tolerant data are incompetent in supporting high-throughput, low-latency multi-modal data transmission in embodied systems. In detail, DRB-based transmissions via UPF suffer from extra latency caused by the interaction with core network, while SRB-based transmission, despite low-latency, are constrained by capacity limitations for high-throughput data. In addition, low latency and reliable data exchanging among devices is essential for seamless collaboration in distributed or mobile deployments of embodied intelligence systems. While D2D communications fail to meet the stringent reliability and security requirements for mission-critical operations, AN-assisted data exchange emerges as a preferred alternative. However, current AN architectures lack native support for such paradigms. Moreover, the throughput of edge devices cannot be guaranteed in current networks.
Use case category	AI-native network optimization
Reference	None.

7.17.2 Assumptions

None

7.17.3 Use case requirements

a. Functional requirements

- New data bearers that support closed-loop operation of data within the AN should be introduced to transport AI service data transmission, processing and sharing.
- A Service Type ID should be included to indicate whether the session requires the AI service data bearer or not.
- AN should have the ability to identify service type and service requirements, and generate resource configurations for the new data bearer.
- AN should have the ability to predict network states, device behavior (e.g., UE mobility, UE load) and service requirements based on collected historical information.
- AN should have the ability to perform proactive AN optimization policies based on predictions, such as communication resource optimizations, data routing optimization.
- Data delivery among devices should be supported within AN, and enhanced data routing and forwarding efficiency to facilitate the collaboration between embodied intelligence devices and that between device and AN.

b. Performance requirements

- Low-latency and high-throughput transmission for embodied intelligence, including edge devices.
- Real-time inference and optimization capabilities with a latency threshold for critical decision-making.

7.17.4 Use case specific figures

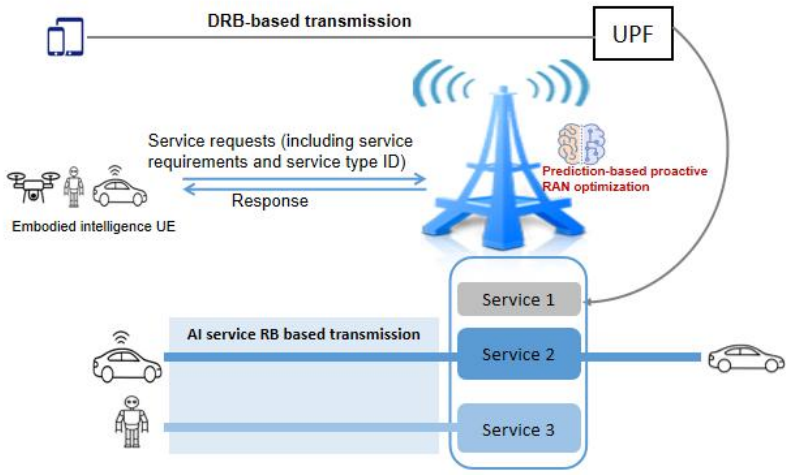


Figure 7.17-1: Proactive AN optimization for embodied intelligence related figure

7.18 Use Case 018: Network-native Personal AI Assistant service

The motivation of the use case lies in addressing the growing demand for intelligent, real-time, and context-aware Personal AI Assistant services across network consumer.

7.18.1 Use case summary

Use case id	FG-AINN-usecase-018
Use case name	Network-native Personal AI Assistant service

Created by	Ziqi Chen, Yi Ren, Lehan Wang, Qi Sun and Hang Ruan from China Mobile
Creation date	04.03.2025
Use case context	Approved in the 38th WG2 meeting (28th May 2026)
Use case description	Proposed use case herein is AI-native service provision of Personal AI Assistant (PAA). The convergence of multi-modal large language models (LLMs) and AI agent technologies is redefining personal AI assistants (PAAs), enabling context-aware, cross-modal interactions (text, voice, vision) on mobile devices. Modern multi-modal LLMs process and synthesize heterogeneous data streams across modalities, demanding ultra-low latency to deliver real-time user experiences. Network-native PAA services leverage AI-native networks, transforming traditional communication infrastructures into essential enabler of mobile intelligence by embedding multi-modal LLMs into the network multi-layered computing platforms. This use case allows for flexible orchestration of LLM deployment to meet stringent user experience, optimizes multi-modal data flows between network and users, facilitates compliance with privacy regulations, and delivers robust AI services across industrial and consumer applications. This integration not only reduces reliance on centralized cloud resources but also paves the way for networks that natively support next-generation AI-driven experiences. In the context of this use case, AI-native telecommunication networks are designed to support large-scale, real-time LLM inference workload by embedding intelligent resource orchestration, AI data internal dynamic routing, distributed multi-modal data processing, and privacy-preserving mechanisms. Specifically, AI-native support for the Personal AI Assistant (PAA) service implies that: • The network itself functions as an intelligent computing substrate, dynamically allocating and scaling LLM inference tasks across cloud and edge nodes in response to latency, user context, and service density; • Data flows are optimized natively to support synchronous multi-modal transmission (e.g., audio-visual-text) with minimal delay and jitter, even under network congestion; • Privacy and model collaboration are managed intrinsically through local processing as needed, ensuring that user private data is handled securely and autonomously. This “native” capability means that the PAA service is not merely hosted over the network but is co-designed with and embedded within the network’s control, computing, and data planes.
Core concepts / characteristics of AINN (Optional)	Transforming traditional network infrastructure into distributed multi-modal LLM platform that caters to end user PAA service requirements.
Actors and Roles (Optional)	An API should connect user(s) to PAA service provider’s system to allow for user prompt, sensed data and response communication. Associated actors are user(s) that demands PAA service, PAA service provider company that provides multi-modal LLM inference service, Service orchestrator that route the user data to running LLM inference servers or initiates new instance of LMM inference engine when

	needed, and LLM Model provider that provides LLM model needed to process user data.
Technical challenges or Gaps (Optional)	Currently, the inference of multi-modal LLMs is mainly performed in centralized cloud service clusters owned or leased by LLM designer. However, with the proliferation of highly interactive, real-time Personal AI Assistants, the growing demand for low latency and increasing service density necessitates the migration of LLM inference to network edges, such as Multi-access Edge Computing (MEC). This transition faces three major technical challenges or gaps: 1) How to effectively orchestrate LLMs deployments across centralized and edge computing resources to mitigate inefficient hardware utilization caused by uneven decentralized user demand and also meets dynamic user experience requirements. 2) How to ensure seamless, synchronized delivery of multi-modal data (e.g., audio, video) even during network congestion triggered by extreme user density. 3) How to guarantee the security of user privacy data throughout its lifecycle—during transmission, caching, and processing—within distributed network architectures. Compared to traditional network functions or conventional machine learning (ML) inference tasks, orchestrating large language model (LLM) services presents distinct challenges. LLMs—especially multi-modal ones—are resource-intensive, requiring significant GPU/TPU computing power, memory, and bandwidth. Unlike typical ML models that are statically deployed, LLM workloads are highly dynamic, with varying model sizes, prompt lengths, and interaction patterns. Moreover, LLM inference often operates in an interactive, streaming, or session-based manner, requiring consistent model states and minimal latency. This imposes stringent orchestration requirements to ensure fine-grained allocation of inference pipelines close to users, based on user context, session persistence, and local demand, real-time horizontal scaling of LLM instances in response to unpredictable traffic bursts and model-aware routing that considers not just network conditions, but also the compute and memory profiles of different LLMs. These requirements go beyond what typical Network Function Virtualization currently support, and call for native orchestration capabilities within the network infrastructure tailored for LLM-based services.
Use case category	LLM-based
Reference	[ITU-T G.114]

7.18.2 Assumptions

None

7.18.3 Use case requirements

a. Functional requirements

- It is of critical value that AI-native telecommunication networks dynamically allocate LLM inference tasks across centralized cloud and edge nodes (e.g., MEC) and other computing platforms based on real-time latency and user density.
- It is expected that AI-native telecommunication networks provide standardized APIs to enable communication between users, PAA service providers, and LLM providers. The API is expected to support real-time bidirectional data exchange for prompts, sensor inputs, and AI-generated responses, as well as maintain synchronization across modalities.

- It is expected that AI-native telecommunication networks provide collaborative LLM services by integrating with local models, thereby preserving user privacy while leveraging interaction history for personalized experiences.
- It is expected that AI-native telecommunication networks scale LLM instances to meet fluctuating demand.

b. Performance requirements

- It is of critical value that the end-to-end response time for user queries should not exceed human-perceivable limits for multi-modal interactions, e.g., few hundreds of ms of ask-response time [ITU-T G.114].
- It is expected that AI-native telecommunication networks handle concurrent requests covering service area without degradation in service quality.
- It is expected that multi-modal data streams (e.g., audio-video pairs) maintain synchronization with a maximum skew of 15 ms under network congestion [3GPP TR 22.874].

7.18.4 Use case specific figures

This clause includes the figures used to describe the use case and clarifies the interaction logic of key components, as well as the support role of the AI-native network for PAA services.

- **End user device**: The user-facing interactive terminal that initiates multi-modal input requests and receives, presents responses from the Personal AI Assistant service.
- **PAA service provider**: The service operator that delivers Personal AI Assistant services in AI-native Telecommunication Network, manages user interactions, and coordinates data flows as well as distributed computing resources.
- **LLM model provider**: The entity that supplies multi-modal large language models, supports model subscription, inference, and offloading for the PAA service.

The figure defines standardized APIs for interaction among the three core participants:

- The API between end user device and PAA service provider carries real-time bidirectional transmission of user prompts, multi-modal sensed data and AI response data, serving as the interaction entry for PAA services. The data used for PAA is the needed data for multi-modal LLM inference, including user text prompt inputs, user speaking voices or surrounding sounds, user filmed camera real-time videos, etc.
- The API between PAA service provider and LLM model provider completes LLM model subscription, model offloading and usage status feedback, building a standardized collaboration channel between service delivery and model supply.

The **PAA Service Provider** is a core functional component embedded within the AI-native Telecommunication Network, serving as the intelligent intermediary and orchestration hub for PAA services. Hosting the network-side Orchestrator, it coordinates with distributed computing resources including the Data Centre and Edge Cloud, to receive multi-modal user prompts and sensed data from end user devices, route requests to appropriate LLM inference nodes, and handle LLM offloading, model subscription, and usage status feedback with the LLM Provider. As an integral part of the AI-native network, it leverages the network's inherent capabilities to manage cross-domain resource scheduling, optimize multi-modal data transmission paths, and ensure low-latency, high-reliability delivery of AI responses, acting as the bridge that connects end users, LLM models, and the underlying telecommunication infrastructure to enable native, carrier-grade PAA service support.

The **Orchestrators** deployed on the end user device side and the PAA Service Provider form a distributed, collaborative scheduling framework that underpins the end-to-end delivery of PAA services. The end-user device-side Orchestrator acts as the local frontline control unit: it receives multi-modal prompts and sensed data directly from the user, invokes the local LLM for low-latency, privacy-preserving inference tasks when possible, and evaluates whether the task requires offloading to the network based on local computing capacity, task complexity, and latency requirements. When

offloading is needed, it forwards the standardized request to the network-side Orchestrator via the unified API interface.

The network-side Orchestrator, as the core scheduling hub of the AI-native Telecommunication Network, receives the forwarded requests, perceives real-time network conditions, and computing resource loads across the Data Centre and Edge Cloud. It then dynamically routes requests to the optimal LLM inference node, coordinates LLM offloading and usage status synchronization with the LLM Provider, and orchestrates cloud-edge collaborative resource allocation.

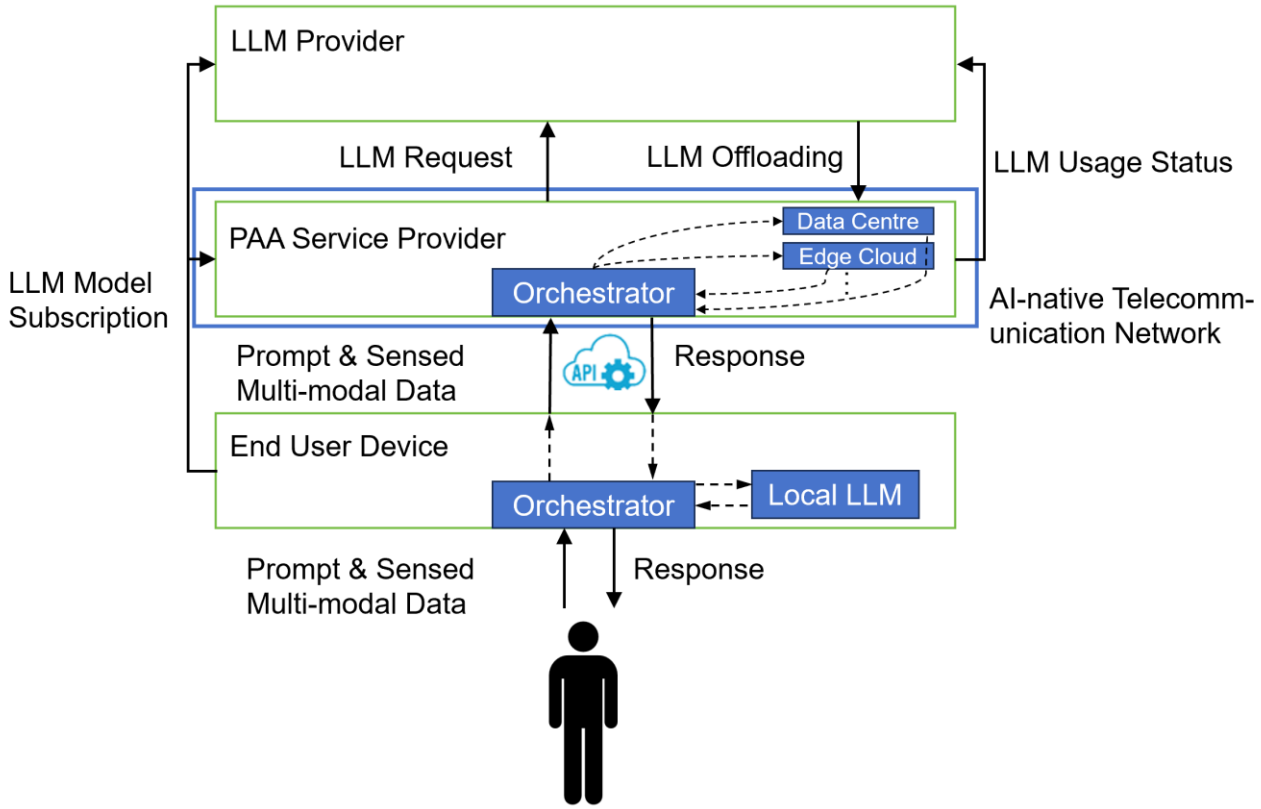


Figure 7.18-1: Network-native Personal AI Assistant service

7.18.5 Bibliography

[ITU-T G.114] ITU-T Recommendation G.114, “One-way transmission time,” International Telecommunication Union (ITU), 2003.

[3GPP TR 22.874] Study on Supporting Tactile and Multi-Modality, document TR 22.847 R18, 2022.

7.19 Use Case 019: Semantic Communications Services for Agents

AI agents are poised to become ubiquitous soon, evolving from simple request-response services for human users to autonomous entities capable of complex interactions. This evolution will drive extensive Machine-to-Machine (M2M) communications based on embedding representations, potentially creating significant network capacity challenges.

Mobile Network Operators (MNOs) could offer differentiated communication services tailored to semantic representations used by AI applications. This would involve negotiating Semantic Service Level Agreements (SSLAs) with applications through dedicated interfaces that exchange key parameters such as relevant metrics, embedding models, and application requirements. Such agreements would enable optimized encoding and transport mechanisms, including intelligent compression and packet handling, to support low-latency AI agent services. To meet strict latency requirements, these services would rely on geographically distributed data centers, including edge resources within operator networks. While this creates new monetization opportunities, such services may also be offered as value-added features to improve network efficiency, reduce operational costs, and enhance user

experience. However, ensuring service continuity for roaming users requires efficient transfer of communication context across distributed nodes.
At the same time, MNOs face increased responsibility for data privacy and security as AI agents and third-party services become more integrated into their networks. Beyond transporting data, operators must ensure end-to-end protection, monitor and audit data access by AI agents, and provide verifiable compliance with data-handling policies. The autonomous and dynamic behavior of AI agents challenges traditional security mechanisms, requiring new, AI-adapted security frameworks. In parallel, AI agents also offer major benefits for network operations, enabling zero-touch management, predictive maintenance, anomaly detection, energy optimization, and customized network slicing. By controlling both infrastructure and data flows, MNOs are uniquely positioned to optimize communication protocols for agent-based, embedding-driven interactions.

7.19.1 Use case summary

Use case id	FG-AINN-usecase-019
Use case name	Semantic Communications Services for Agents
Created by	Quentin Lampin, Louis-Adrien Dufrène (Orange) quentin.lampin@orange.com, louisadrien.dufrene@orange.com
Creation date	23/02/2026
Use case context	Approved in the WG2 meeting on 25 June 2026.
Use case description	Semantic Communications Services for Agents enables mobile networks to offer specialized, low-latency communication services tailored to AI agents by exchanging meaning-rich representations. Through negotiated semantic service agreements, the network adapts encoding, transport, and resource allocation to application needs, supporting distributed AI services across edge and cloud while maintaining service continuity for roaming users. The use case also addresses the need for secure, privacy-preserving operation of autonomous agents, with monitoring and compliance mechanisms integrated into the network. Beyond user-facing services, these capabilities empower operators to deploy AI agents for network automation—supporting zero-touch operations, anomaly detection, energy optimization, and customized network slicing—making semantic communication a foundation for both agent services and autonomous network management.
Core concepts / characteristics of AINN (Optional)	This use case reflects key AINN characteristics, including AI-native service exposure, semantic-aware communication, and closed-loop intelligence across network layers. AI models and semantic processing are embedded within the network to adapt encoding, transport, and resource allocation to application intent and context. The network supports distributed AI inference across edge and cloud, enabling low-latency agent interactions and service continuity under mobility. The use case also aligns with AINN principles of autonomous operation, intent-based management, and trustworthy AI, integrating privacy, security, and compliance mechanisms into AI-driven network services.
Actors and Roles (Optional)	The Mobile Network Operator (MNO) provides AI-native semantic communication services, including resource orchestration, semantic-aware transport, and security enforcement. AI agent service providers deploy and manage autonomous agents and embedding models that rely on network-exposed AI capabilities. Application and service developers define semantic requirements and negotiate service parameters with the network. End users and vertical stakeholders

	consume agent-based services. Edge and cloud infrastructure providers host distributed AI inference and model synchronization functions. Regulators and compliance authorities define privacy and security constraints that guide trustworthy AI operation within the network.
Technical challenges or Gaps (Optional)	This use case exposes several technical gaps in current AI and network systems. Existing networks lack native support for intent-based communication, semantic SLAs, and adaptive encoding of embeddings. Current AI deployments are largely siloed at the application or cloud layer, with limited integration of AI lifecycle management into network functions, especially across edge–cloud continua. There is no standardized framework for exposing network AI capabilities via APIs, nor for coordinating distributed inference and maintaining semantic context under mobility and roaming. Security and privacy mechanisms are not designed for autonomous, continuously learning agents, creating gaps in monitoring, compliance verification, and trust management. Finally, existing orchestration systems lack AI-native closed-loop control to jointly optimize communication, compute placement, and energy efficiency for agent-based services in real time.
Use case category	AI-agent in AI native networks
Reference	None.

7.19.2 Assumptions

The use case assumes the availability of AI-native network capabilities, including semantic-aware communication functions and exposure of AI/ML resources through standardized APIs. It assumes the presence of distributed edge and cloud infrastructure within the operator network to support low-latency inference and model synchronization. Basic support for intent-based networking, network slicing, and AI-driven orchestration is assumed to be available in future network systems. It is also assumed that privacy, security, and compliance mechanisms are integrated into the network architecture and that participating stakeholders (operators, service providers, and application developers) agree on semantic service parameters and data-handling policies.

7.19.3 Use case requirements

a. Functional requirements

- The network shall support real-time interactions between human users and agents, as well as between agents, enabling continuous exchange of semantic information.
- The network shall provide a distributed computing framework that enables execution of agent functions across edge and cloud nodes, including dynamic workload placement and coordination.
- The network shall provide authentication, authorization, and access control mechanisms to contain agent operations within defined permissions and security boundaries.
- The network shall provide secure execution environments for agent operations to prevent unauthorized actions.
- The network shall support data governance functions, including data access control, integrity verification, and enforcement of data-handling policies for user and operator data.
- The network shall support semantic communication capabilities, including:
  - embedding-based data exchange,
  - semantic-aware encoding and compression,
  - protocol support for semantic transmission patterns.
- The network shall support interoperability between agents, including:
  - common or translatable embedding representations,

- open interfaces for inter-agent communication.

- The network shall expose developer-facing APIs that allow third parties to integrate applications and AI functions with network capabilities.
- The network shall support agent state management, including state preservation and restoration across network events and failures.

b. **Performance requirements**

- The network shall optimize end-to-end latency for both human–agent and agent–agent interactions.
- The network shall maintain real-time response capabilities for interactive services under varying network conditions and load levels.
- The network shall monitor and optimize performance using KPIs reflecting user experience, not only low-level technical metrics.
- The network shall ensure efficient workload distribution across distributed computing nodes to sustain high-performance agent operations.
- The network shall maintain Quality of Service (QoS) guarantees for semantic communications, accounting for the specific characteristics of embedding-based data exchange.

7.20 Use Case 020: Federated Cross-Domain Energy-Efficient Network Optimization

Driven by the densification of urban areas, the information and communications technology sector contributes to the globally growing energy consumption [3]. In particular, the Access Network are responsible for approximately 80% of the total network's energy consumption. However, since the load on different base stations changes dynamically with time depending on traffic arrivals and user mobility, there is a high potential to improve the overall network energy efficiency with smart algorithms that dynamically change the network configuration. Further energy savings can be achieved if the consumption of the computing platforms deployed at the edge/cloud is considered [1], [2].

This use case describes an AI-native approach to end-to-end (E2E) energy efficiency in future communication networks using Digital Twins (DTs) operating in a federated, cross-domain environment. The use case focuses on how AI-native capabilities enable predictive and closed-loop optimization across networking and computing domains while respecting administrative and operational boundaries.

Rather than relying on tightly coupled, domain-specific optimization, the proposed approach leverages abstracted network state information, predictive functional models, and federated coordination mechanisms. This allows network and computing domains to independently optimize their resources while contributing to system-level energy efficiency goals and maintaining Quality of Service (QoS) guarantees.

As a synchronized digital representation of the network and computing domains, the federated DT approach enables what-if analysis, predictive traffic estimation, and proactive resource orchestration. From the network perspective, the AI-native functional models embedded in the Network DT (NDT) can anticipate traffic fluctuations and optimize multi-band resource allocation. From the computing domain, the computing DT proactively creates controllers to scale in or out the computing replicas, depending on the traffic estimations made on the network counterpart. The federated approach aligns network and edge computing decisions in a closed-loop manner. This predictive, cross-domain, and data-driven control is required to achieve substantial energy reduction while guaranteeing SLA compliance, which cannot be reliably realized with conventional non-AI-native architectures.

In this use case, a Network DT (NDT) refers specifically to a digital twin of communication networks in their traditional sense, including the access, core, and transport domains. A computing DT, by contrast, refers to a digital twin of computing infrastructure, which is not considered an integral part of the network under this traditional definition. Unless stated otherwise, the term DT is used to cover both NDTs and computing DTs.

7.20.1 Use case summary

Use case id	FG-AINN-usecase-020

Use case name	Federated Cross-Domain Energy-Efficient Network Optimization
Created by	6G-TWIN Consortium from SNS-JU
Creation date	30/04/2026
Use case context	This use case was approved in 45th WG2 meeting.
Use case description	The use case considers a multi-domain environment where networking and computing infrastructures are operated by different administrative entities. Direct joint control or centralized orchestration across these domains is therefore not assumed. While the networking environment can apply specific optimizations to optimize energy consumption (e.g., user association to specific radio bands, coordinating sleep modes and power control), the computing environment should proactively adjust to such changes. For doing that, each domain creates their own controllers to optimize their resources, e.g., the computing domain can create a controller to scale in or out the computing resources accordingly to the expected workload for the networking domain to optimize a given KPI. Following that concept, both domains maintain its own DT, continuously synchronized with real-time and historical telemetry. These DTs expose abstracted state information and predictive insights, enabling coordination without revealing internal implementation details. To create those controllers, the AN orchestrator will create different experiments where the DTs are used as AN Sandboxes allowing the controllers to be deployed and jointly explore the optimization space. The interaction of the federated optimization through the domain-specific controllers happens not at the level of the overlay network, but at the digital level, making the applied changes be safely evaluated before deployment. To allow a faster search in the optimization space, the domain specific DTs can be based on data-driven functional models such as, but not limited to: • Prediction of traffic and service demand evolution • Estimation of energy consumption under alternative configurations • Evaluation of QoS risks associated with energy-saving actions The decisions of the domain-specific controllers are validated within the digital environment before being applied to the operational network, ensuring safety, and policy compliance. The following functional models (domain-specific models of the underlay) are provided: • **Access Domain Adaptation Models** to support predictive refinement of transmission configurations and planning of reconfigurable network elements, operating on abstracted access domain indicators exposed by the NDT. • **Network Energy Consumption Models** to estimate energy usage of network infrastructure under different traffic distributions, user associations, and cell activation states. These models evaluate the energy impact of configuration alternatives and support proactive, energy aware network operation while respecting service constraints.

	• **Access Configuration Selection Models** to recommend suitable connectivity and transmission options for active users when capacity-oriented resources are enabled, balancing service performance targets with energy efficiency objectives. • **Cross Domain Latency Prediction Models** to forecast end to end service latency metrics (e.g., p95/p99) as a function of traffic load and resource allocation across domains, enabling early detection of potential SLA violations. • **Computing Domain Energy Models** to estimate energy consumption at node and cluster level in edge and cloud environments, considering resource utilization, scaling policies, and heterogeneous hardware characteristics. These models enable evaluation and comparison of orchestration strategies, including workload consolidation and elastic scaling, from an energy efficiency perspective. • **Joint Energy-Performance Assessment Models** to analyze tradeoffs between service quality and energy consumption across networking and computing domains, supporting what if analysis and policy decisions aligned with sustainability objectives. All models are managed through AI-native pipelines and operate within the federated DT framework, supporting shadow mode validation and autonomous execution in federated, multi domain environments.
Core concepts / characteristics of AINN (Optional)	• The use of DTs as AN Sandboxes is required for safely exploration of the optimization space, so the introduced changes during this exploration, do not reflect in production networks. • The use of AI-based models as domain specific models in the DT is required to scale and speed the approximation of future states. • The use of on-demand deployment of scaling policies that better suit the current operational states is required. • The use of MLOps is required for the creation and optimization of AI-based controllers.
Actors and Roles (Optional)	• **NDT** (e.g., Access Network): Digital twin of the underlay network responsible for the generation of current and hypothetical scenarios of the network. • **Computing DT**: Digital Twin of the computing infrastructure, responsible for the generation of current and hypothetical scenarios of the computing domain. • **Knowledge Base**: Database responsible for the storage and management of the generated assets, including scenarios and controllers, and their corresponding metadata. • **Model Optimization Pipeline**: Instance of a MLOps framework, responsible for the training and optimization of AI-based models. • **E2E Network Orchestrator**: Orchestrator that supervises the E2E operation of the federation.

	• **Underlay Network**: Instance of the real, physical network, responsible for the generation of data to feed the (N)DTs and receiving the actions from the controllers. This can represent the networking or the computing domain.
Technical challenges or Gaps (Optional)	• Definition of standardized abstraction and federation interfaces for the DTs • Ensuring prediction accuracy under partial observability • Maintaining QoS and SLA guarantees during energy optimization • Interoperability with non-AI-native and legacy systems • Governance and accountability for AI-driven cross-domain decisions
Use case category	Digital twin
Reference	https://6g-twin.eu/demonstrations/

7.20.2 Assumptions

The following assumptions apply to this use case:

- **High-fidelity Digital Twins are available**, capable of mirroring network and computing states, performing what-if analyses, and enabling predictive and closed-loop optimization.
- The DTs have access to **real-time and historical telemetry** from the network and, where applicable, from the edge/cloud computing infrastructure, including load, traffic patterns, radio conditions, and energy consumption metrics.
- **Actuation interfaces are exposed by the network infrastructure**, allowing the DT decisions to influence network and computing configurations.
- The **network and computing domains are assumed to be administratively separate**, preventing direct joint orchestration and requiring coordination through federated DTs rather than tight integration.
- The evaluation assumes that **QoS and SLA requirements must not be violated** while applying energy-saving strategies; energy optimization is therefore always constrained by service performance guarantees.
- Alternative energy-saving mechanisms might apply but are considered **out of scope** for this use case.

7.20.3 Use case requirements

a. Functional requirements[PS1]

- **Realistic network environment and service modelling capabilities:** It is of critical value for the system to support modelling of network and computing environments, traffic dynamics, and service demands to enable predictive analysis of performance and energy impact under varying operational conditions.
- **Telemetry, Observability, and Data Handling:** It is of critical value for the system to provide continuous observability of performance, resource utilization, and energy metrics across networking and computing domains to support KPI computation and AI-decision validation.
- **AI Pipeline and Lifecycle Support:** It is of critical value for the system to support end-to-end AI lifecycle management, including data ingestion, model training, validation, deployment, inference, and re-training, and shall provide the necessary networking and computing resources for AI operation.
- **Distributed and Federated AI Coordination:** It is of critical value for the system to enable distributed and hierarchical AI models with federated coordination across administratively separate domains, supporting predictive and what-if analysis through Digital Twins.

- **Actuation and Control interfaces:** It is expected for the system to support abstracted interfaces for network and computing configuration and actuation via the DT.
- **Visualization and What-If Evaluation:** It is expected for the system to provide visualization and what-if analysis capabilities to assess system state, compare optimization strategies, and evaluate outcomes against baseline configurations.

b. Performance requirements

- **End-to-End Energy Efficiency Improvement:** It is of critical value for the system to achieve a degree of improvement in end-to-end energy savings across networking and computing domains when compared to baseline or expert-driven operation.
- **Computational Efficiency:** It is expected for the system to reduce execution time and computing resource consumption associated with AI-driven optimization and Digital Twin operation, relative to baseline approaches.
- **Resource Utilization Optimization:** It is of critical value for the system to improve utilization efficiency of networking and computing resources, minimizing redundant processing and unnecessary data exchange
- **QoS-Constrained Optimization:** It is of critical value for the system to maintain service performance and SLA compliance for the majority of users while applying energy-saving actions, ensuring that optimization does not negatively impact QoS.
- **AI Model Accuracy and Effectiveness:** It is expected for the AI-based functional models to achieve high prediction accuracy, sufficient to support reliable decision-making and proactive optimization.
- **Service Availability and Stability:** It is of critical value for the system to improve operational stability, reducing service interruptions, failures, and unnecessary restarts compared to baseline operation.
- **Efficient AI Lifecycle Operation:** It is expected for the AI inference and adaptation cycles to be executed with limited latency and without causing service disruptions or SLA violations.

7.20.4 Use case specific figures

This use case addresses the optimization of network and computing infrastructure to achieve end-to-end energy efficiency. For instance, one of the most effective strategies for reducing energy consumption in the access network is the use of advanced sleep modes in their base stations [4], [5].

From the computing perspective, activating or deactivating base stations introduces fluctuations in aggregated traffic that must be processed. Accurately identifying these changes is essential to maintain Quality of Service (QoS) and Quality of Experience (QoE). Furthermore, power-aware orchestration of computing applications can be optimized to enhance energy efficiency without compromising service performance.

To be able to collaborate, the two domains require a federation mechanism to achieve holistic energy efficiency. To proactively adapt to traffic variations, the computing domain can request traffic traces from the NDT as part of its internal optimization loop, enabling “what-if” analyses. For instance, the NDT may provide traces for bursty traffic, on-off traffic patterns, or scenarios involving the addition (e.g., capacity planning) or removal (e.g., resilience to failures) of base stations. The computing platform also collects infrastructure-level data to model the impact of traffic patterns on QoS metrics such as latency and energy consumption. By combining these insights, a Computing DT can be constructed to evaluate management strategies, including scaling policies that incorporate energy efficiency objectives [6], [7]. Using these models and traffic traces, the system can simulate multiple scenarios and assess alternative scaling strategies. Ultimately, the Computing DT recommends the most suitable approach for each traffic pattern, ensuring optimal performance during real-world operation.

The use case is divided in three phases. **Figure 7.20-1: Phase 1 of UC, Federated Operation**

depicts the first phase, where the Computing DT requests metrics from the NDT (e.g., uplink/downlink throughput, number of connected users) for current and hypothetical scenarios, including changes in user density, addition/removal of gNBs, and traffic pattern variations. Interactions between these independent domains occur via the E2E Network Orchestrator [8]. If the Knowledge base contains the requested data, it shares them directly; otherwise, the NDT generates hypothetical scenarios through simulation/emulation. Once generated, outcomes are transferred to the Computing DT via the E2E Network Orchestrator. Data transformations may be required to ensure compatibility, handled by the Knowledge Base block applying DataOps.

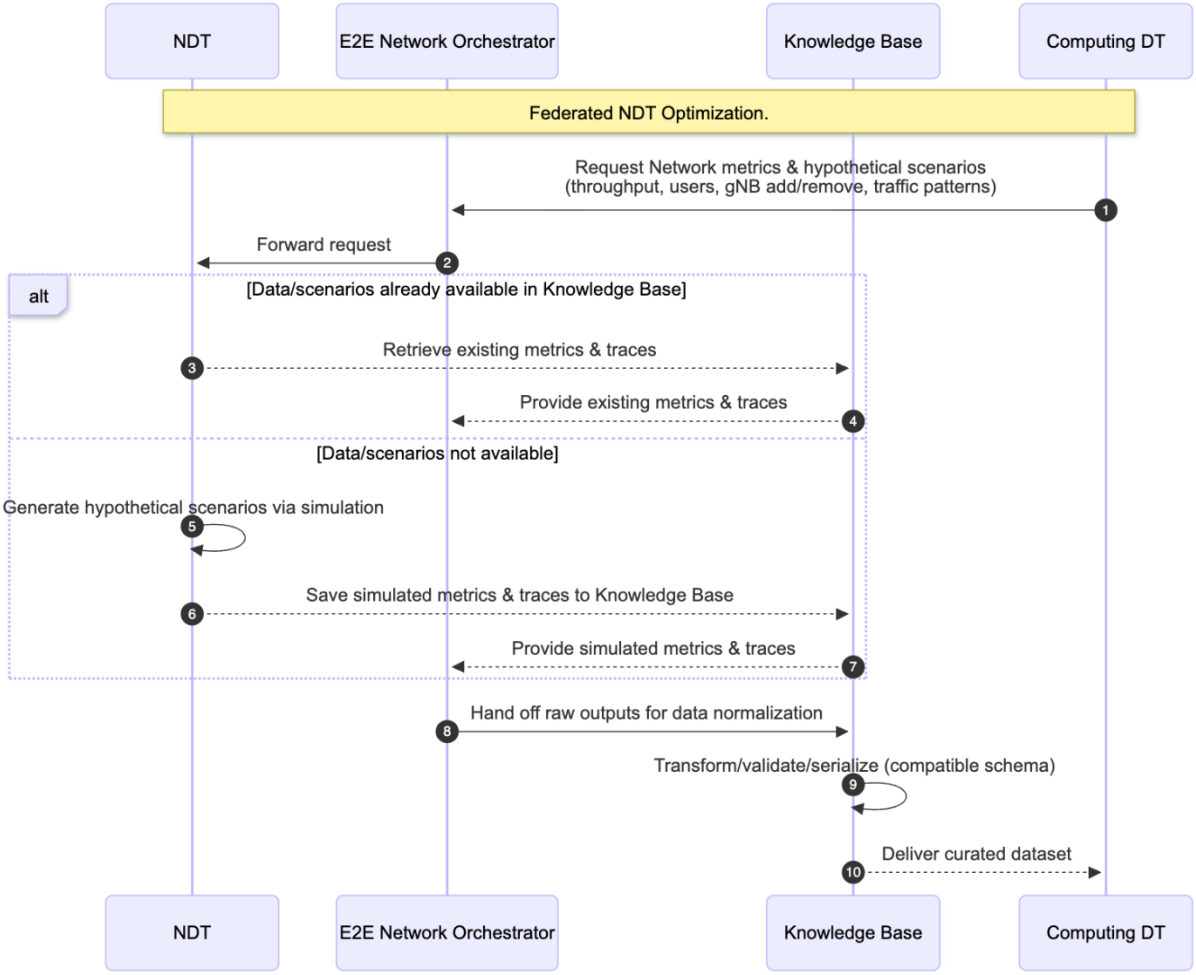


Figure 7.20-1: Phase 1 of UC, Federated Operation

In the second phase of this use case, the Computing DT then uses generated data/scenarios to create and optimize potential experimental controllers responsible for generating scaling policies. These policies determine the computing resources required to meet SLA requirements under varying RAN conditions while minimizing energy consumption. The workflow of this phase is shown in **Error! Reference source not found.**.

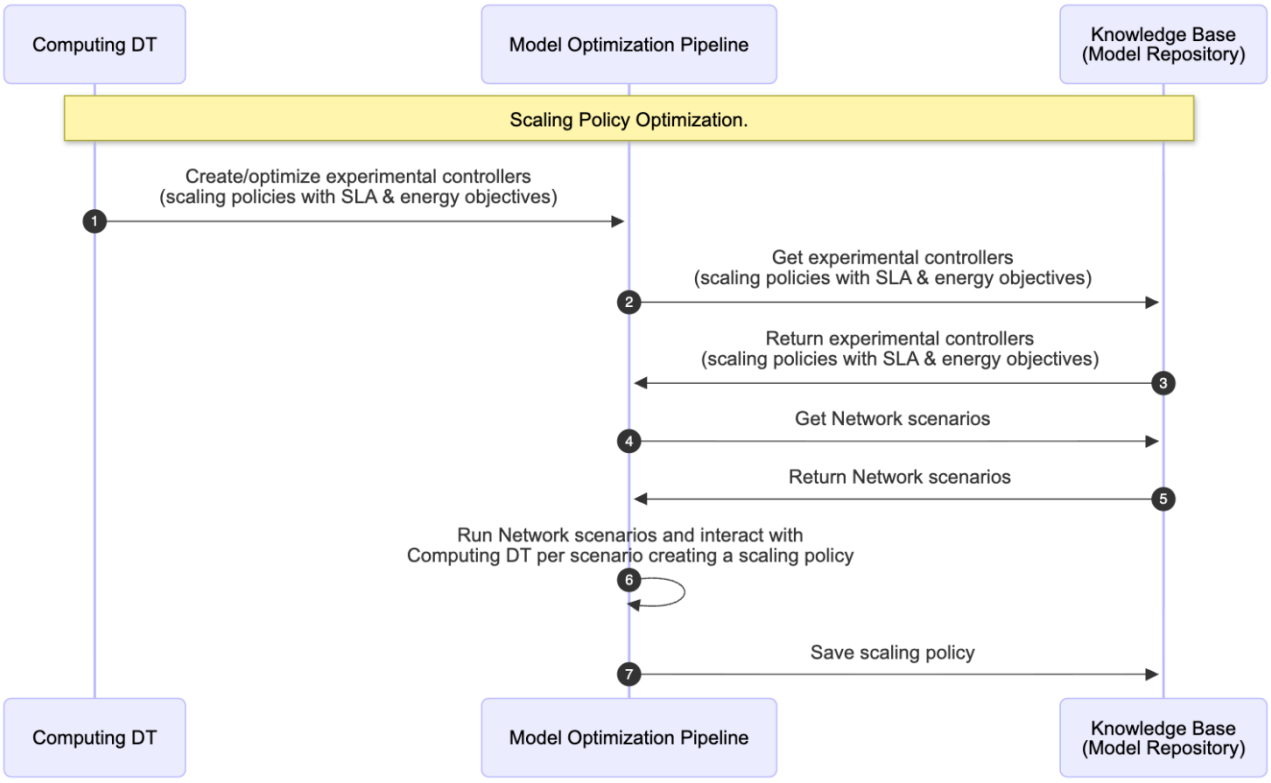


Figure 7.20-2: Phase 2 of UC, Scaling policy optimization

Lastly, the third phase carries the evaluation of the experimental controllers i.e., scaling policies. Multiple scenarios yield multiple scaling policies, stored in a model repository with associated metadata. From the multiple generated controllers, the Computing DT evaluates the behaviour of such controller in the current state of the underlay network, as seen by the NDT. The best performing controller can then be deployed in the underlay network (physical twin).

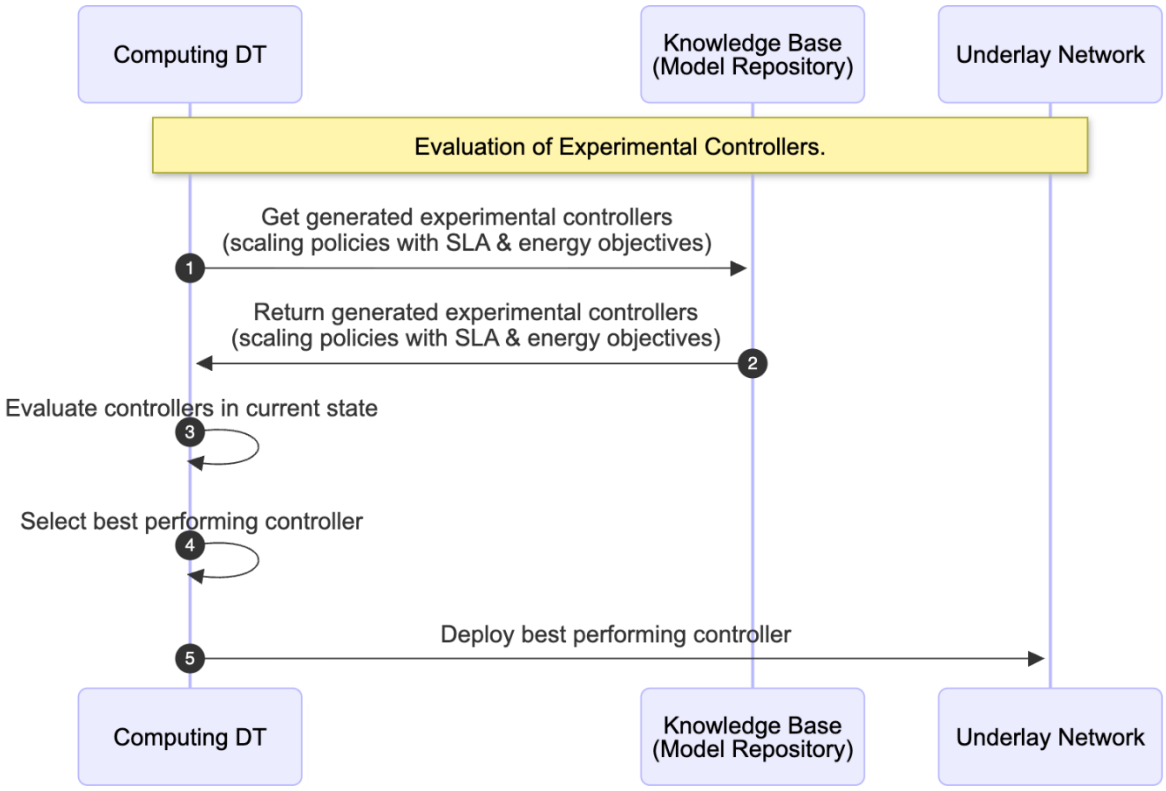


Figure 7.20-3: Phase 3 of UC, Evaluation of Experimental Controllers

7.20.5 Bibliography

[1] ITU-T, “Digital Twin Network - Requirements and Architecture,” Recommendation Y.3090, 2022.
[2] A. Zaki-Hindi, P. Soto, G. Castellanos, T. H. Pritom, G. Volpe, I. Zellagui and others, “A Reference Functional Architecture for Network Digital Twins in 6G Systems,” *IEEE Open Journal of the Communications Society,* vol. 7, pp. 2068-2101, 2026.
[3] J. Huttunen, M. Pärssinen, T. Heikkilä, O. Salmela, J. Manner and E. Pongracz, "Base Station Energy Use in Dense Urban and Suburban Areas," in IEEE Access, vol. 11, pp. 2863-2874, 2023, doi: 10.1109/ACCESS.2023.3234192.
[4] O-RAN Alliance, “Security Requirements and Controls for O-RAN Defined Interfaces and Network Functions,” O-RAN Alliance, Technical Specification O-RAN.WG4.CUS.0-R003-v13.00, 2023”.
[5] O-RAN Alliance, “Potential Energy Savings Features in O-RAN By the Sustainability Focus Group (SUFG),” 1 2025
[6] R. Brondolin and M. D. Santambrogio, “PRESTO: a latency-aware power-capping orchestrator for cloud-native microservices,” in *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*, 2020.
[7] V. Gudepu, R. R. Tella, C. Centofanti, J. Santos, A. Marotta and K. Kondepu, “Demonstrating the Energy Consumption of Radio Access Networks in Container Clouds,” in *NOMS 2024-2024 IEEE Network Operations and Management Symposium*, 2024.
[8] ITU-T Focus Group on Autonomous Networks, “Architecture framework for Autonomous Networks”, Technical Specification, 2022.

Use Cases In Progress

7.21 Use case No. A1: LLM agents-powered intent-based network slicing for industrial automation

Proposed use case herein is intent-based network management with network slicing for industrial automation, however, it may be extended to any other use case relevant to network slicing. Intent-based network The network slicing is assisted by AI agents (LLM agents) introduced in [ETSI GR ENI 051] in network level in which high-level business intents are received through APIs from customer(s), then translated into actionable workflows in relevant network domains.

7.21.1 Use case summary

Use case id	FG-AINN-usecase-X
Use case name	LLM agents-powered intent-based network slicing for industrial automation
Created by	Ibrahim YAZICI
Creation date	27.01.2025
Use case context	This includes the reference of the input documents and/or related meetings. To be included at a later stage in the final report.
Use case description	The motivation of the use case is to enable industrial automation with enhanced network slicing and NAMO capabilities assisted by AI technology. Entailing intense human-level intervention for network slicing and network administration, the proposed use case cannot be realized currently with a high level of autonomy for the use case. Since intent-based network slicing is not using an intelligent agent in current situation, human operators need to define slice requirements working with customers for their needs (e.g.,

	reliability, latency, bandwidth), configure slices by allocating resources and defining SLAs, and administrate virtualized (VNFs) or cloud-native (C-NFS) for the slices (e.g., mMTC for smart city, URLLC for industrial IoT). Moreover, they may need to intervene when automation fails or customization is needed for deployment and orchestration processes along with monitoring and performance management processes along with dynamic optimization and scaling processes. These requirements entail intense human intervention in current network slicing operations. The key points of this use case are as follows [LLM whitepaper]: • Agentic solution providing for intent-based management through autonomy, • More autonomy-based network configuration for complex tasks and reduce human intervention in "network slice design", • Autonomous design and complex task handling resulting in latency reduction for the use case realization. The diversity of network services, resources and capabilities are expected to thrive, then standardized processes based on scenario-specific expert knowledge will in turn fall short for efficient operation of the network. This is caused by the diversity necessitated along with increased flexibility that will be vital for service requirements and network functionalities for future networks. As a result, unprecedented challenges will be brought to the network architecture design. In particular, these complexities in future network systems will further complicate network slicing in terms of deployment, configuration and management of new type of slice, thus flexible, autonomous and generalizable network slicing will be paramount. This complexity will also increase by incorporation of intent-based management with network slicing. This situation will make conventional methods insufficient, necessitating deep integration of AI/ML technology into the operations of future networks. In this sense, AI agents will offer a promising solution with this unprecedented challenging environment. They are expected to help decision-making and taking actions intelligently without having been specified in advance or without human intervention. The agents equipped with data-driven, advanced reasoning and inference capability, adaptability to real-time changes in the environment present more generalized, adaptable and extensible design for intent-based network slicing powered by LLM agents for intent-based network slicing industrial automation use case.
Core concepts / characteristics of AINN (Optional)	In the context of LLM agents-powered intent-based network slicing use case, LLM agents and network slicing will be enabling technologies while intent-based management is a framework supported by these enabling technologies. LLM agents are the enabling technology for intent-based

	management in the sense of enabling high level business intent translation/interpretation and decision-making introducing intelligence to domains where these operations are handled manually in most of the time, and network slicing enables resource virtualization and allocation based on the translated intents into actionable workflows by the LLM agents. On the other hand, intent-based management is the framework that benefits from these enabling technologies to achieve the goals defined for the use case.
Actors and Roles (Optional)	An API should connect customer(s) to communication service provider's system to demand network slicing by an intent prompt to AI-agent powered system. API(s) should make connection between AI-agent in the E2E management system and domain-specific AI-agents for intent translation, workflow instruction conveying for slicing, intent fulfillment status report. Fulfillment status of requested demand by intent(s) should be conveyed to the customer(s) by the network. LLM agents and network functions should have communication capability to exchange the fulfillment status and instruction(s). Associated actors are customer(s) that demands communication service, communication service provider company that provides communication service, and AI models that help translate high-level abstractions into actionable workflows for the network. The lifecycle of intent-based network slicing powered by AI-agent incorporates intent submission and translation, slice design and planning as per defined policies by the intent, slice instantiation and deployment, slice activation, slice assurance, and slice decommissioning at the end of the slicing process.
Technical challenges or Gaps (Optional)	In current network systems, network slicing heavily depends on human-level intervention to fulfill the network slicing task(s) if resource availability allows for it. In addition, intent-based management currently does not utilize LLMs to introduce more autonomy in network administration, management, and orchestration level. Hence, for industrial automation use case, specific requirements of the use case will be conveyed with little human intervention to network system, and required network slicing and intent-based management will be realized as per the network resources allow. This whole process will make use of AI natively with LLMs in network level that is not currently in action.
Use case category	To be included at a later stage when the families of use cases are defined.
Reference	

7.21.2 Assumptions

- The agents are domain-specific ones, and they carry out tasks defined within their respective domains. They interact with each other enabling cross-domain task execution as well,
- The proposed system does not rely on vendor-specific infrastructure. API-based communication enables smooth and seamless integration within the network,
- Anonymization or encryption for sensitive data can be applied in demand before LLM-agent processing.

7.21.3 Use case requirements

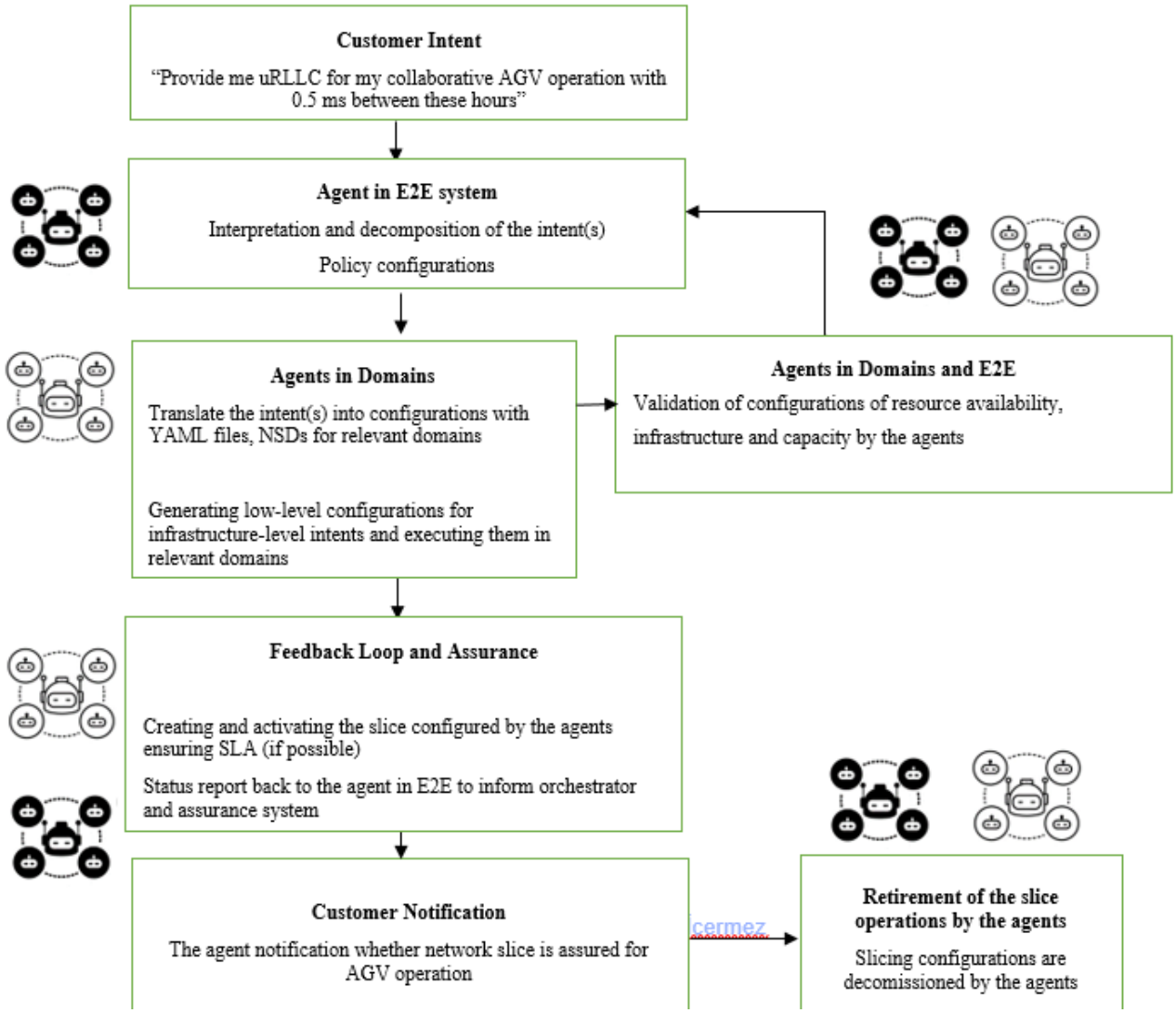
a. Functional requirements

- The network system has capability to receive (high-level) business intent(s) from the customer(s) through the API system,
- In the network system, LLMs have capability to translate the high-level intent(s) into actionable workflows in the network configurations for network domains,
- LLMs have capability to be managed by LLMOps to get updated for changing environment conditions,
- Network functions and LLMs have capability to communicate to exchange information such as the fulfilment status of the customer request, and instructions for the intent(s).

b. Performance requirements

- The key performance requirements for industrial automation, those are low-latency and high-bandwidth assurance by the network,
- Human-level intervention and human-level error decline in network management and network slicing,
- Level of autonomy increase in network administration, management, and orchestration,
- User experience increase for network slicing requests, and automated service providing,
- AI-based technology error level such as hallucination problem of LLMs,
- Protection of data privacy and be complaint with regulations on data.

7.21.4 Use case specific figures



[FGAINN intro] https://www.itu.int/en/ITU-T/focusgroups/ainn/Pages/default.aspx

8 Summary

This Technical Specification summarizes and organizes 20 AI-native telecommunication network related use cases, including but not limited to the following categories:

.

Categories	Collection of Use Cases
LLM-based	004, 018
AI-native networks applied in vertical industries	001, 005, 013, 014, 015
AI-agent in AI native networks	006, 016, 019
AI training/inference as a service	002, 003
AI-native digital twins	009, 015, 020
AI-native network optimization	007, 012, 017

Data support for AI-native networks	008
Native API of AI native networks	010
Security and trust	011

Based on the analysis of all use cases, use case level requirements in terms of functionality, performance and other aspects have been derived for AI-native network. Each category of use cases has certain similarities considering the requirements.

Potential common requirements of use cases:

- a. Functional requirements
  - Support the complete ML model lifecycle, including joint optimization of connection, data collection and processing, algorithm, and models;
  - Support the AI training or AI inference as service provisioning for both Network for AI and AI for Network;
  - Support for distributed and hierarchical AI models operating at different levels (e.g. device, edge, network and cloud);
  - Support data lifecycle management of the entire data process for AI training and inference, including data creation, collection, transmission, processing, storage and consumption;
  - Support the coordination of multiple AI agents through AI-native models.
  - Support autonomous execution of network optimization through coordinated interaction among AI agents and network functions (entities).
  - Support the authentication, authorization, and access control mechanisms to contain agent operations within defined permissions and security boundaries.
  - Support providing intent recognition and interpretation based on AI-native method;
  - Support dynamic QoS enforcement, traffic prioritization, resource orchestration and allocation for the AI-native network.
  - Support exposure of AI related network capabilities that allows third-party applications to integrate applications and AI functions.
  - Support dynamic monitoring and analysis of real-time network conditions to optimize performance based on closed-loop feedback in AI-native network.
  - Support a centralized or distributed repository for AI models, enabling storage, update and deployment of AI models.
- b. Performance requirements
  - Ensure effective service provisioning with limited latency and required accuracy of the AI training and inference;
  - Increase computing resource utilization in different levels (e.g. device, edge, network and cloud);
  - Minimize network overhead for model or data lifecycle management by enabling reuse of data exposed to AI and avoid duplicate data collection;
  - Balance AI task completion with required performance and resource usage;
  - Ensure quality of AI related data across multiple domains, including performance of ML models, statistical richness, compactness of data, timeliness and freshness;

Optimize end-to-end latency for both human user–agent and agent–agent interactions.The use case categorization and corresponding requirements can serve as the foundation for architecture design consideration in WG3.

Bibliography

[FG-AINN-WG1-04]	FG-AINN-WG1-04, Definition of AI-native telecommunication networks
[FG-AINN-I-WG1-AI Agent]	FG-AINN-I-WG1-AI Agent, Definition of AI agent in AI-native telecommunication network
[ADG2023]	F. Ayed, A. De Domenico, A. Garcia-Rodriguez, and D. López-Pérez, Accordion: A communication-aware machine learning framework for next generation networks, IEEE Commun. Mag., vol. 61, no. 6, pp. 104–110, Jun. 2023
[MCP]	Anthropic, “Model Context Protocol: A protocol for seamless integration between LLM applications and external data,” sourceshttps://github.com/modelcontextprotocol
[3GPP TR 22.850]	3GPP TR 22.850; Technical Specification Group Services and System Aspects; Study on 3GPP AI/ML Consistency Alignment (Release 19)
[ETSI GR ENI 004]	ETSI GR ENI 004 V3.1.1 (2023), Experiential Networked Intelligence (ENI); Terminology.
[ETSI GR ENI 051]	ETSI GR ENI 051 V4.1.1 (2025), Experiential Networked Intelligence (ENI); Study on AI Agents based Next-generation Network Slicing.
[3GPP TS 28.105]	3GPP TS 28.105; Technical Specification Group Services and System Aspects Management and orchestration; Artificial Intelligence/ Machine Learning (AI/ML) management (Release 19).
[b-ISO/IEC 2382-28]	ISO/IEC 2382-28 (2015), Information technology – Vocabulary
[b-Yao2019]	Yao, Haipeng, et al. "AI routers & network mind: A hybrid machine learning paradigm for packet routing." IEEE Computational Intelligence Magazine 14.4 (2019): 21-30.
[b-Wu2022]	Wu, Wen, et al. "AI-native network slicing for 6G networks." IEEE Wireless Communications 29.1 (2022): 96-103.
[b-Yang2022]	Yang, Tingting, et al. "Liquid software-based edge intelligence for future 6G networks." IEEE Network 36.1 (2022): 69-75”.
[b-Wang2022]	Wang, Weili, et al. "VHetNets for AI and AI for VHetNets: An Anomaly Detection Case Study for Ubiquitous IoT." arXiv preprint arXiv:2210.08132 (2022).
[b-Duan2022]	Duan, Sijing, et al. "Distributed Artificial Intelligence Empowered by End-Edge-Cloud Computing: A Survey." IEEE Communications Surveys & Tutorials (2022).
[b-Blanco2023]	L. Blanco, S. Kuklinski, E. Zeydan, F. Rezazadeh, A. Chawla, L. Zanzi, F. Devoti, R. Kołakowski, V. Vlahodimitropoulou, I. Chochliouros, A. M. Bosneag, S. Cherrared, L. A. Garrido, S. Barrachina, J. Mangues, “ AI-driven Framework for Scalable Management of Network Slices” in IEEE Communications Magazine , vol. 61 , issue 11, pp. 216 - 222, November 2023.
[b-Barrachina2024]	S. Barrachina-Munoz, F. Rezazadeh, L. Blanco, S. Kuklinski, E. Zeydan , A. Chawla, L. Zanzi, F. Devoti, V. Vlahodimitropoulou, I. Chochliouros, A. M. Bosneag, S. Cherrared, L. Vettori, J. Mangues-Bafalluy “ Empowering Beyond 5G Networks: An Experimental Assessment of Zero-Touch Management and Orchestration ” in IEEE Access, vol. 12, pp. 182752 - 182762, December 2024.
[FG-AINN-I-WG2-016]	Methodology USED CASE Hybrid AI-Native Networks: Architecture for Decentralized Services Security and Interoperability