6GSNS

Smart Networks and Services
Joint Undertaking (SNS JU)

Intelligent and Reliable Software
Networks Working Group (SNWG)

White Paper

The AI/ML Framework for Smart Networks and Services

DOI: 10.5281/zenodo.20629064
URL: https://doi.org/10.5281/zenodo.20629064

July 2026

ACKNOWLEDGEMENT

Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the SNS JU. Neither the European Union nor or the SNS JU can be held responsible for them.

EXECUTIVE SUMMARY

This white paper presents a comprehensive vision for the evolution of smart network infrastructures and services, positioning Artificial Intelligence (AI) as a foundational enabler of sixth generation networks (6G) and next-generation digital ecosystems. Serving as a sequel of the SNWG white paper[1] on AI and Machine Learning (ML) landscape for smart network and services, this white paper highlights the great AI potential for the 6G networks as well as the key challenges emerge from the adoption of AI in network operations. With inputs from multiple SNS JU projects, AI/ML frameworks are presented towards transforming network operations from traditional rule-based management into autonomous, adaptive, resilient, and intent-driven systems. In the service and resource orchestration domain, those frameworks bring intent-driven solutions and enable zero-touch closed loops; while they facilitate orchestration in multi-domain and federated environments. In the application domain, AI/ML-enabled applications drive innovation in future network services, including concepts, such as programmability, joint communication & sensing, and Network Digital Twins(NDTs).

Regardless of the scope of application and the target network domain where AI/ML is adopted, security, trust and privacy aspects should be considered both for the integral operation of AI/ML components and their lifecycle management. Especially for lifecycle of AI/ML models there is an increased need for pipelines that ensure reproducibility, traceability, continuous learning, and operational consistency.

1 https://smart-networks.europa.eu/wp-content/uploads/2026/01/sns-ju-snwg_ai.ml-landscape_final.pdf

CONTENTS

INTRODUCTION

The increasing complexity and dynamism of telecommunication networks, which demand more than traditional static or semi-automated management approaches, set an urgent need for intelligent, autonomous, and secure network operations and services.

In this context, AI and ML are positioned as foundational technologies to enable adaptive, resilient, and intent-driven operations, supporting business transformation and unlocking new value across the Information and Communication Technologies (ICT) ecosystem. However, many technological challenges emerge due to a) the stochastic dynamics and resource constraints in radio access and edge domains, b) the need for scalability and heterogeneity across distributed, federated network environments, c) the increased security threats to AI models, including data poisoning, adversarial attacks, and model extraction, and d) the demand for efficient lifecycle management of AI/ML models, ensuring reproducibility, traceability, and continuous optimization.

In view of these challenges, SNS JU projects have proposed orchestration frameworks and agentic AI approaches for multi-domain, cloud-native service deployment and adaptation. Such frameworks enable a) Intent-driven orchestration, where stakeholders specify desired outcomes and AI-enabled systems autonomously determine optimal actions; b) Distributed intelligence layers and marketplaces, towards AI-as-a-Service (AIaaS) and collaborative model sharing; c) NDTs for predictive reasoning, policy validation, and conflict detection, enhancing operational assurance; d) Robust security and trust frameworks leveraging Federated Learning (FL), Trusted Execution Environments (TEEs), and privacy-preserving mechanisms; and e) Machine Learning Operations (MLOps) pipelines and federated frameworks to automate and scale AI/ML lifecycle management across diverse domains.

1 AI/ML IN THE NETWORK DOMAIN

As the complexity of digital infrastructures increases, traditional static or semi-automated management approaches are no longer sufficient to guarantee the levels of performance, adaptability, and resilience expected in 6G networks. Irrefutably, AI is becoming a foundational component of next-generation mobile networks. As mentioned in the recent European Telecommunications Standards Institute (ETSI) white paper on Autonomous networks (AN) [1], AI will play a key role in the evolution of AN to achieve AN level 4 (where network analyses and makes decisions with no human intervention) and to open the gateway to the Digital Transformation, with key business advantages for the whole ICT ecosystem and Enterprises. At the core of this process lies the use of AI models in network operations [2], having, amongst others, the two following challenges: a) trust and privacy issues when network data are used to train AI models, and b) stochastic dynamics of the Radio Access Network (RAN) and the computing constraints at the edge domain [3].

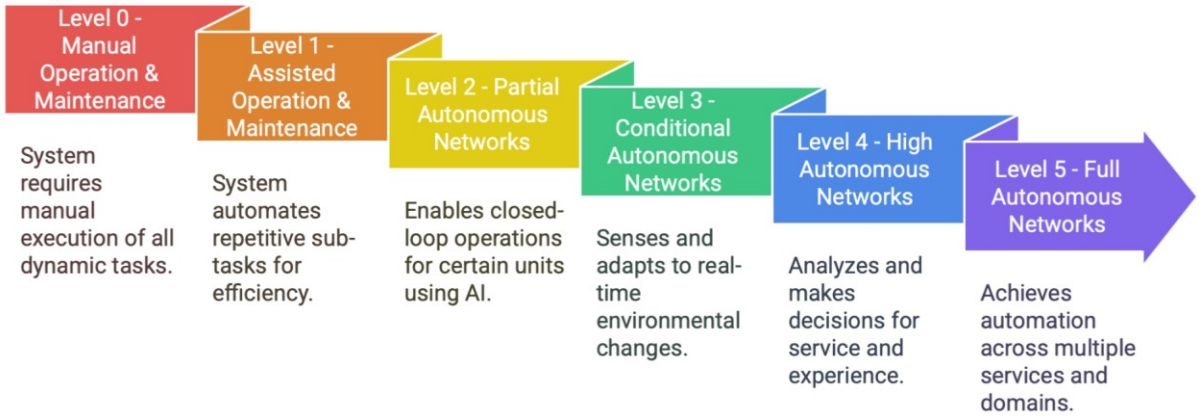


Figure 1: Simplified view of Autonomous Network levels

1.1 AI as integral part of the network

Future infrastructures will need to integrate pervasive network intelligence, capable of learning from dynamic contexts, predicting operational conditions, and autonomously reconfiguring themselves to meet the runtime needs of mobile users, services, and applications. This transformation is enabled by several converging factors: the explosion of data generated by heterogeneous devices, sensors, and network functions spanning terrestrial and non-terrestrial domains, the deep integration of connectivity, computing, and device resources across the infrastructure continuum, and the emergence of services requiring extreme performance combined with trustworthiness and flexibility. Immersive communications, industrial automation, and safety-critical operations are just a few representative examples.

In this evolution, the concept of the "Network of Networks" (NoN) is becoming very relevant, since it enables dynamic and flexible composition and seamless collaboration among infrastructures, resources, and intelligence from multiple administrative and technological domains [4]. However, this highly distributed system introduces several challenges. Ensuring optimal resource utilization, maintaining Service Level Agreement (SLA) guarantees, and guaranteeing consistent levels of trust and security across federated environments require automation mechanisms that go far beyond today's closed-loop systems.

Moreover, as AI becomes natively embedded into every infrastructure layer, networks themselves must learn how to manage AI in a sustainable and trustworthy manner from an end-to-end (E2E) perspective, by optimizing placement and scheduling of AI tasks, orchestrating models, validating their outcomes, and reconciling conflicts among several concurrent decision loops. This evolution towards Native AI architectures considers AI functions as intrinsic components of the network, requiring mechanisms for transparency, explainability, and reliability.

A potential framework in this direction is illustrated in **Figure 2** where network intelligence and AI/ Machine Learning (ML) mechanisms are applied in a pervasive manner, tightly integrating them across **business, service, and resource layers** to enable E2Eand intent-driven automation, collaboration, and assurance. Three main pillars are identified, as described below.

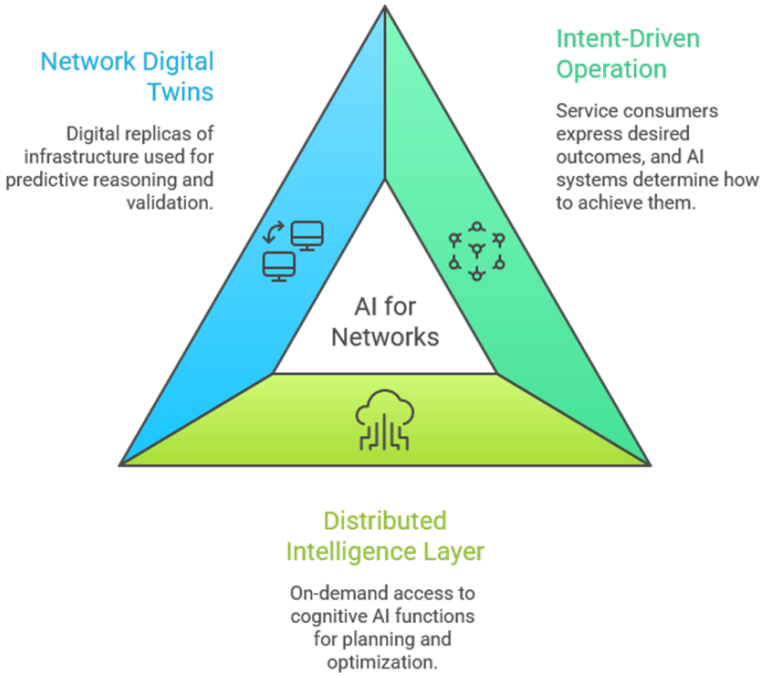


Figure 2: The three main pillars that support the AI for networks concept

The intent-driven operation comprises the *first pillar*. Service consumers and stakeholders express desired outcomes in declarative form specifying "what" should be achieved while leaving to AI-

enabled systems the responsibility of determining “how” to achieve them. Intents are managed through collaborative Intent Management Functions (IMFs) that operate at different layers: business, service, and resource. These functions, supported by agentic AI and large language models (LLMs), can interpret complex intents expressed in natural language, translate them into actionable objectives, and coordinate their fulfilment through closed-loop automation. Business-level intents are processed within a Smart Marketplace, where the negotiation and the selection of composite service bundles—including connectivity, computing, and in general AI Services. Service and resource intents are subsequently handled within the NoNCollaborative Framework, which orchestrates the actual provisioning and runtime adaptation of resources from federated infrastructures.

The *second pillar* is the distributed Intelligence Layer. It provides on-demand access to cognitive AI-based functions for planning, self-optimization, self-configuration, and self-healing. These AI services are exposed to a Smart Marketplace through unified interfaces, enabling the integration of customizable pipelines to ensure traceability and lifecycle management of AI models. The resulting architecture supports placement and migration of AI workloads across the cloud-edge continuum, dynamically adjusting connectivity and computing resources to maintain service continuity and compliance with intent-driven constraints. At the business level, the Smart Marketplace constitutes an innovation in how AI/ML functions are delivered and monetized. Beyond traditional network and infrastructure service models, such as the Infrastructure, Platfrom and Software as a Service (IaaS/PaaS/SaaS) the concept of AI-as-a-Service (AIaaS) is introduced as an added-value service, which offers discovery, composition, and brokerage of AI models contributed by multiple stakeholders. Practically, AIaaS functions either as a developer-facing tool built on PaaS, or a highly specialized layer that injects intelligence directly into final SaaS applications. The marketplace operates as a federation broker, providing a unified and multi-service intent Application Programming Interface (API) that enables operators, verticals, and service providers to collaborate through intent-based negotiation and automated service composition. This establishes an ecosystem where AI models, datasets, and inference capabilities can be shared, reused, and commercialized under controlled and auditable conditions.

The *third pillar* is the integration of NDTs. A multi-layer NDT framework features digital replicas of multi-technology infrastructure domains, network and computing resource, network slices and E2Eservice domains, which are interconnected to capture inter-dependencies and cross-domain constraints. NDTs serve as analytical engines for predictive reasoning, policy validation, and conflict detection among concurrent control loops. Before an AI-driven decision is applied in the real system, its outcome is validated through the NDTs, significantly improving accuracy and trustworthiness. Following this approach, NDTs act as arbiters among multiple cognitive loops, supporting both

proactive, reactive, as well as predictive decision-making in complex operational contexts. Closed loops can be specialized by technological scope, layer, or domain, while being capable of inter-loop collaborations. NDT-assisted coordination mechanisms can detect, mitigate, and resolve potential conflicts or temporary oscillations arising from overlapping or contrasting optimization objectives. This results in a cohesive and stable automation framework that extends across multiple domains.

1.2 AI models for networks and services

MLOps are designed to enable the E2E orchestration of ML workflows (as explained in Section 4), streamlining the full lifecycle of ML models (i.e., model development, deployment and monitoring) across distributed network environments. However, the use of AI models for optimizing 6G networks and services is not a straightforward process. Reliable and efficient operation of future 6G systems require AI models capable of reasoning under uncertainty [5]-[8]. In addition, as AI models are integrated across the entire network stack, for instance at the cloud or at the edge and even on-board satellite nodes, computational efficiency, interoperability and energy awareness become critical design objectives [9]-[11]. Figure 3 illustrates four of the most relevant challenges that need to be tackled to efficiently leverage AI in 6G systems.

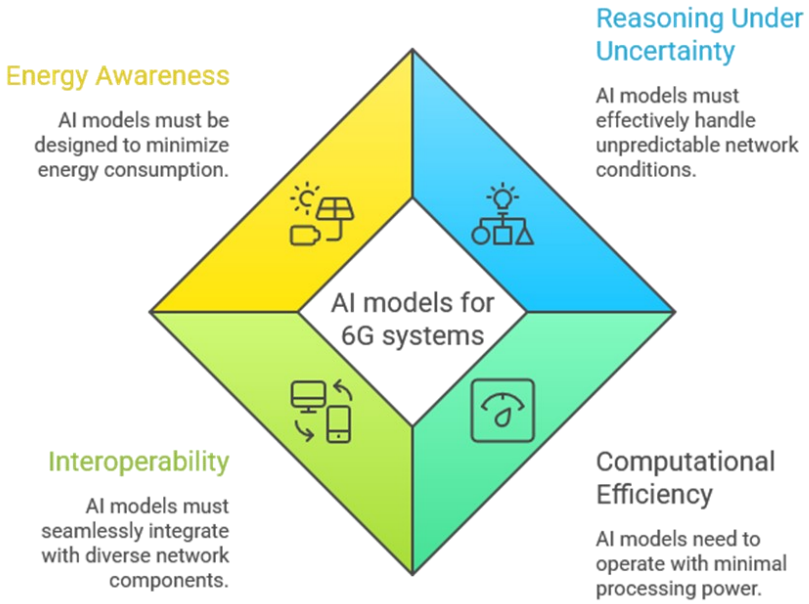


Figure 3: Challenges in using AI models for 6G Network Optimization

1.2.1 Uncertainty-aware AI models

Network environments are characterized by stochastic dynamics resulting from fluctuating channel conditions, user mobility and variable traffic patterns. Deterministic ML models, while effective in stable environments, cannot express confidence in their predictions and are therefore unappropriated

for autonomous decision-making under uncertain conditions. To address these limitations, probabilistic and uncertainty-aware forecasting frameworks are needed to enable proactive and risk-sensitive control of heterogeneous networks. It is proven that a probabilistic reasoning improves spectral efficiency, energy utilization and trustworthiness, forming a foundation for AI-native and risk-aware automation in 6G networks [5]-[8]. By representing predictions as probability distributions or by attaching confidence measures to their outputs, AI models can convey the reliability of their inferences to the control plane. This enables the network to modulate its decisions according to the assessed risk level. For example, a resource allocator may reserve additional capacity when the model's confidence in a congestion forecast is low, whereas a high-confidence scenario may justify more aggressive energy-saving strategies. Similarly, mobility management algorithms can adjust handover thresholds dynamically based on the quantified uncertainty in predicted link quality. This risk-sensitive reasoning enables the network to optimize service continuity, reliability and energy efficiency simultaneously [13].

Embedding uncertainty estimation into AI workflows also strengthens trustworthiness and explainability, as operators gain visibility into both the decision and its associated confidence level. These properties are essential for certification, accountability and operational assurance of autonomous network functions. Uncertainty-aware machine learning therefore provides the conceptual and technical foundation for AI-native networks that are data-driven, self-aware and context-adaptive. Such capabilities will be critical to achieving resilient, trustworthy and risk-sensitive 6G services [5]-[8].

1.2.2 Lightweight AI models

The deployment of large-scale deep models in constrained environments is limited by hardware capacity, latency, and power budgets. These constraints are particularly prevalent in scenarios where resources are scarce, such as in mobile devices, IoT (Internet of Things) sensors, embedded systems, and edge computing nodes. In these environments, the computational power available is often significantly lower than that of centralized data centers, requiring the use of lightweight algorithms and models that can operate efficiently without sacrificing performance.

Use cases demanding such constrained environments are diverse and continually evolving. For instance, in smart cities, numerous small sensors collect vast amounts of data for real-time analysis to enhance traffic management, improve public safety, and optimize energy consumption. In healthcare, portable medical devices equipped with AI for patient monitoring must run on limited battery life while maintaining high accuracy in diagnostics. Autonomous vehicles, which rely on quick decision-making based on sensor inputs, also illustrate the critical need for low-latency AI computations in

restrictive hardware settings. Additionally, applications in remote areas with unstable network connectivity necessitate that AI processes be executed locally, further emphasizing the importance of lightweight architectures.

Functional and parameter-efficient AI models approximate complex nonlinear mappings using adaptive activation functions or compact functional connections instead of large weight matrices. These architectures significantly reduce parameter counts while maintaining predictive expressiveness, allowing inference in resource-constrained systems, for instance at network edge or onboard satellite processors, with minimal delay [9]-[12]. Beyond computational savings, such designs enhance interpretability and explainability, aligning with regulatory demands for trustworthy and transparent AI [10].

Integrating these models within hierarchical or federated network-learning setups enables small local agents to perform context-specific decisions, such as interference mitigation or channel selection, while higher-level controllers aggregate and refine their insights. This distributed intelligence paradigm improves network scalability, resilience and energy proportionality.

2 AI/ML FOR SERVICE ORCHESTRATION

2.1 Intent Driven Management and Orchestration

An *intent* is a formal representation that defines and communicates knowledge about requirements, goals and constraints to a system, enabling automated processes to reason about it and derive suitable decisions and actions. The intent concept is an integral part of Intent-Driven Autonomous Networks (IDAN), primarily affecting management and orchestration procedures. In this context, separated service and resource management are foreseen, thereby creating a novel paradigm that leverages intents as a main driver for the specification of requirements and constraints[2]. The Intent-driven Management and Orchestration (DIMO) framework is presented below, including closed-control loops driven by intents. DIMO aligns closely with recent standardization efforts, such as those promoted by Tele Management Forum (TM Forum) [14] and European Telecommunications Standards Institute Zero-Touch Service Management (ETSI ZSM) [15]. Common notions within TM Forum refer to autonomous domain management, intent-based management, separation of business and service management, and the implementation of closed-control loops at each system level. DIMO also aligns with ETSI ZSM primarily in sharing the concepts of closed-control loops and intent-based management.

A global vision of DIMO is depicted in Figure 4. The main components are: the tenant platforms, which group both verticals and separate tenants that leverage Tenant Domain Manager and Orchestrator (t-DMO); the Domain Manager and Orchestrator (DMO), which operates at a service level, making decisions considering the whole topology; the Network Compute Fabric (NCF), which abstracts the underlying 6G infrastructure, defined as 6G resource pools. Each resource pool has a Local Manager and Orchestrator (LMO) that provides management and orchestration capabilities over the technological/administrative domain.

2 SNS-JU projects 6G-BRICKS (https://6g-bricks.eu/) and 6G-INTENSE (https://6g-intense.eu/)

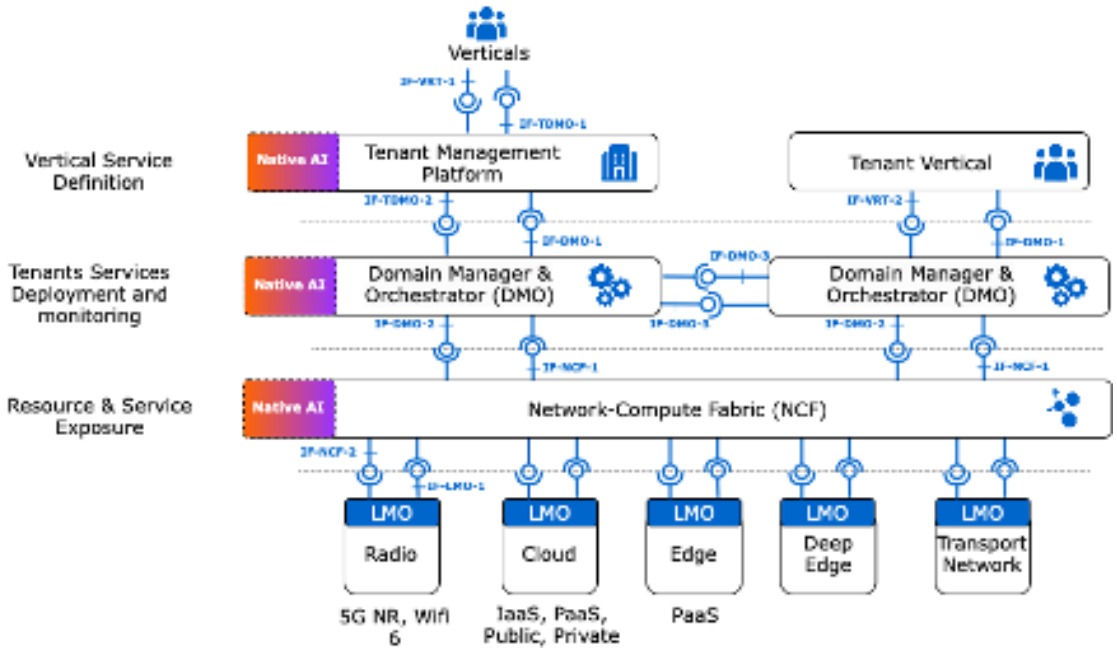


Figure 4: Key components of a DIMO Architecture

With more detail, the main components for management and orchestration are:

- t-DMO: This component is responsible for handling vertical services and processing service intents from the underlying DMO. It incorporates a business closed-control loop mechanism that interprets high-level user expectations, expressed in natural language, and orchestrates services accordingly.
- DMO: It operates at the service level, receiving service intent inputs from the t-DMO or directly from verticals.
- NCF: It works at the resource level, providing an abstraction layer for resource allocation across heterogeneous technological and administrative domains. Given the integration of diverse 6G resource pools, this abstraction is essential for enabling distributed resource management.

Finally, the native integration of AI/ML models is supported through the introduction of a dedicated Native AI plane, leading to the so-called Native AI toolkit. This plane spans three architectural layers of DIMO: t-DMO, DMO, and NCF, and interacts with them via well-defined interfaces. The Native AI plane represents a key innovation in enabling automation for intent reasoning, declaration, negotiation, and decision-making across the autonomous domains. The Native AI components incorporate AI/ML models that are embedded within the closed-control loops operating at the Business, Service, and Resource levels of the architecture. Such an integration strengthens the zero-touch automation framework, leveraging intelligent models to enhance responsiveness, adaptability, and efficiency across all operational domains.

2.2 AI-driven Zero-Touch Closed-Loop Orchestration

AI-driven zero-touch orchestration is emerging as a key enabler for automated service management across the edge–cloud continuum. This section presents a closed-loop orchestration framework for E2E service management in heterogeneous distributed systems [3]. The proposed framework continuously monitors infrastructure and service conditions and automatically adapts network resources and service placement. Through an iterative closed-loop process, the system ensures compliance with SLAs and Quality of Service (QoS) requirements even under dynamic network conditions.

The proposed zero-touch closed-loop orchestration framework, shown in Figure 5, combines an orchestration platform, AI-driven analytics, and an observability system to enable proactive orchestration decisions.

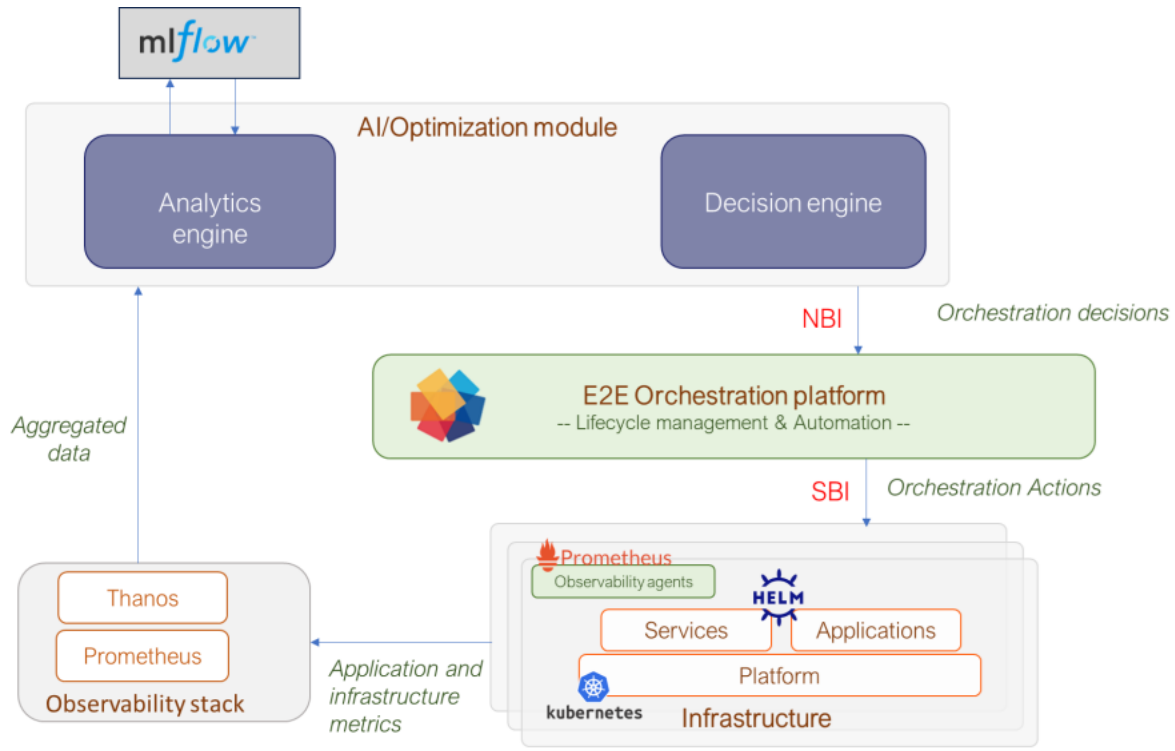


Figure 5: Zero-touch closed-loop framework

E2E edge orchestration platform. The orchestration platform[4] manages the lifecycle of distributed applications and infrastructure across the edge–cloud continuum. It maintains global visibility of

3 SNS JU project ETHER (https://ether-project.eu)

4 Based on the principles and the implementation of NearbyOne - https://www.nearbycomputing.com/nearbyone/

resources through agents deployed across infrastructure layers for monitoring purposes. The platform exposes a northbound interface (NBI) to external applications, such as the AI decision engine, and a southbound interface (SBI) to execute orchestration actions on the underlying infrastructure.

AI/Optimisation module. This module ensures that services and infrastructure remain in the desired operational state. It consists of two components:

- An analytics engine which uses machine learning models to analyse historical and real-time metrics and predict future system states such as link quality, user location, or resource availability.
- A decision engine, which uses these predictions, together with the current system state, to generate orchestration actions such as scaling resources, migrating applications across nodes, or terminating instances.

ML models used by the analytics engine are managed through an MLflow[5]-based framework that supports experiment tracking, model lifecycle management, and scalable deployment of predictive models.

Compute Infrastructure. The underlying infrastructure consists of distributed computing platforms deployed across terrestrial segment while also allowing for extension to aerial, and space segments. Each site is based on a Kubernetes (K8s) cluster supporting cloud-native applications implemented as microservices to ensure scalability, resilience, and flexibility.

Observability stack. An observability stack collects and aggregates infrastructure and service metrics. Built on open-source tools such as Prometheus, Thanos, and Grafana, it provides monitoring, long-term storage, and querying capabilities for operational and analytical components of the framework.

Enabling interfaces. Several interfaces enable communication between components and integration with external systems:

- NBI: Representational State Transfer (REST) APIs connecting AI modules with the orchestration platform.
- SBI: It connects the orchestrator with K8sclusters to execute orchestration commands.
- Aggregated Data Interface: It provides access to monitoring data through Thanos, using PromQL queries.

5 https://mlflow.org/

2.3 Agentic AI for service orchestration

Agentic AI has emerged as a promising paradigm for intelligent orchestration [7][8]. Agent-based systems enable autonomous, context-aware decision-making, continuous learning, and adaptive responses to evolving network conditions. Such capabilities are essential for zero-touch orchestration in dynamic cloud-native environments and have already been explored in domains including software development, information retrieval, mission-critical 6G services, and autonomous network management [15]-[19]. Given that, a multi-agent orchestration framework, named as AgentEdge, is presented for edge-cloud service orchestration. AgentEdge introduces three key innovations [20]:

- ActSimCrit, a simulation-driven validation methodology for orchestration actions.
- PARES (Perceive, Act, Reason, Evaluate and Sustain), a framework defining the core capabilities of autonomous agents.
- Hierarchical graph-of-graphs coordination, enabling distributed orchestration across edge–cloud environments.

In AgentEdge, orchestration intelligence is distributed among specialized agents with each agent designed to conform to the PARESCapability, enabling calibrated autonomy and collaborative decision-making. It consists of three main components, as can be seen in Figure 6. At its core lies the Multi-Agent Graph, which coordinates four specialized agents, responsible for sequential orchestration phases: intent processing, observability analysis, planning, and infrastructure execution. The Agent Engineering module manages the lifecycle and coordination of agents while providing capabilities such as performance evaluation, reasoning traceability, monitoring of orchestration metrics (e.g., resource usage and SLA compliance), and version control for model checkpoints and system rollback. The LLM Toolset provides AI models and communication mechanisms enabling cooperation among agents and interaction with external systems.

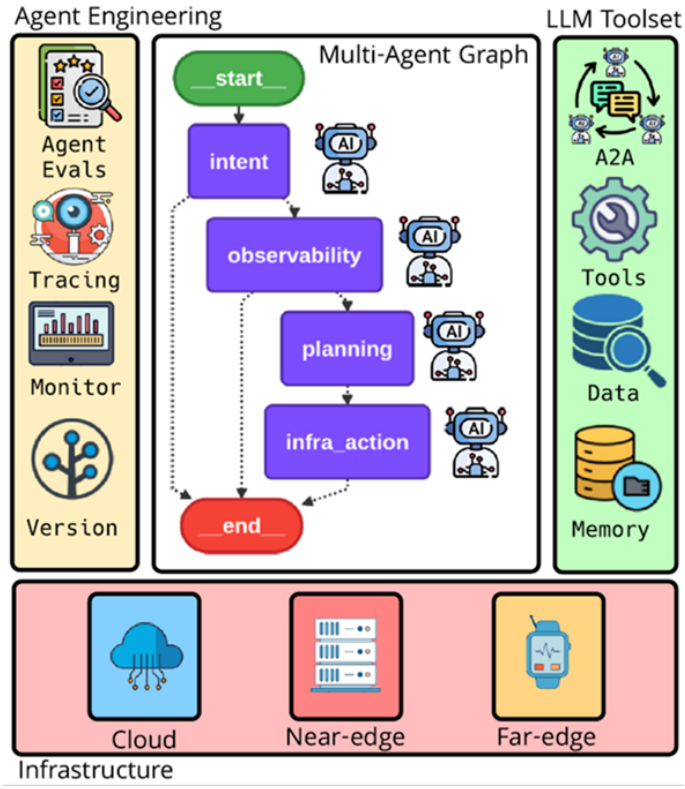


Figure 6: The AgentEdge Orchestration framework

The orchestration system follows a three-tier architecture that spans cloud, near-edge, and far-edge nodes, and is implemented as loosely coupled microservices communicating through standardized APIs. Using the PARES framework, orchestration tasks are distributed among specialized agents that continuously perceive system state, reason about actions, evaluate outcomes, and sustain long-term operation.

The *Intent Agent* serves as the interface between operators and the orchestration system, translating high-level natural language requirements into structured deployment objectives and constraints.

The *Planning Agent* generates orchestration plans using reasoning and historical context. It employs the ActSimCrit methodology, where candidate actions are validated through simulation before deployment. A Critic LLM evaluates simulated outcomes to ensure feasibility, policy compliance, and alignment with user intents.

The *Infrastructure Action Agent* executes validated orchestration actions on the infrastructure through deterministic API operations. It also incorporates LLM reasoning to interpret failures and initiate recovery procedures, ensuring resilient execution across distributed environments.

The *Observability Agent* continuously analyzes telemetry from edge and cloud nodes to detect deviations from expected performance or SLA targets. By providing contextual feedback to the planning and execution agents, it enables proactive adaptation of orchestration strategies.

2.4 Frameworks for AI-enabled federated networks

AI is expected to play a transformative role in next-generation networks, turning communication infrastructures into distributed intelligence systems capable of autonomously optimizing performance across domains [21]-[23]. However, current AI-driven solutions for networking remains siloed, focusing on specific domains, such as the RAN or Core Networks (CNs), without addressing cross-domain integration and coordination. Achieving real E2E AI-driven automation requires seamless coordination across multiple network domains, including radio, optical transport, edge, and cloud infrastructures.

At the same time, several systemic barriers still limit the widespread adoption of AI-driven network control. In first place, AI deployments remain siloed, confined to specific technological domains. Second, access to training data is limited, due to privacy and ownership constraints. Third, 6G is expected to have highly heterogeneous infrastructures with diverse hardware and software environments. Finally, the absence of standardized AI cooperation mechanisms further complicates the adoption of AIaaS and multi-domain AI control.

Achieving pervasive, trustworthy, and efficient AI-enabled networking requires architectural frameworks capable of supporting decentralized decision-making, model lifecycle management, multi-stakeholder collaboration, and robust security.

2.4.1 Challenges in distributed and federated networks

Future networks are envisioned as highly distributed systems, integrating communication, computation, and intelligence into a unified fabric. However, the implementation of pervasive AI in networking brings several challenges:

- **Scalability and Heterogeneity:** 6G environments will combine dense edge deployments with diverse device capabilities and rapidly changing network conditions. Ensuring consistent performance requires adaptive algorithms that can operate under heterogeneous hardware configurations and dynamic traffic patterns. East–West (E-W) coordination across equivalent control domains is needed for consistent decision-making across technologies.
- **Orchestration of Intelligence:** AI models have different requirements during training and inference. While training often relies on resource-rich environments, inference may occur at constrained edge nodes. This creates a need for orchestration mechanisms that can distribute, update, and coordinate models efficiently, while balancing latency, computation, and bandwidth constraints. Synchronization across devices and domains becomes particularly complex when real-time responsiveness is required.

- **Decentralized learning:** Moving intelligence closer to the data source reduces latency and preserves privacy, while also introduces challenges such as high communication overhead for distributed learning, heterogeneous data distributions across devices, the straggler effect from low-capable nodes, slower convergence in dynamic topologies. These challenges highlight the need for hierarchical North–South (N-S) coordination between local and global AI control loops, enabling scalable and resilient learning processes.
- **Resource constraints:** Edge devices typically operate with limited computing, storage, and energy resources. Deploying advanced AI models in these environments requires lightweight inference mechanisms, hardware-aware optimizations, and intelligent offloading strategies that consider both performance and sustainability.
- **Privacy and security concerns:** AI-driven network control must ensure the confidentiality and integrity of data and models. Distributed environments increase the attack surface and introduce new threats, such as model poisoning or unauthorized access to local datasets. Trustworthy AI in networking requires decentralized trust mechanisms without single points of failure, verifiable model integrity, secure data sharing protocols, and robust defences against adversarial manipulation.

2.4.2 Architectural principles for AI-enabled federated networks

To address the abovementioned challenges, future 6G systems will benefit from **modular and composable AI control frameworks** that provide a structured way to manage data, models, and intelligence across stakeholders and network domains. At a high level, such frameworks can be described through three foundational architectural blocks.

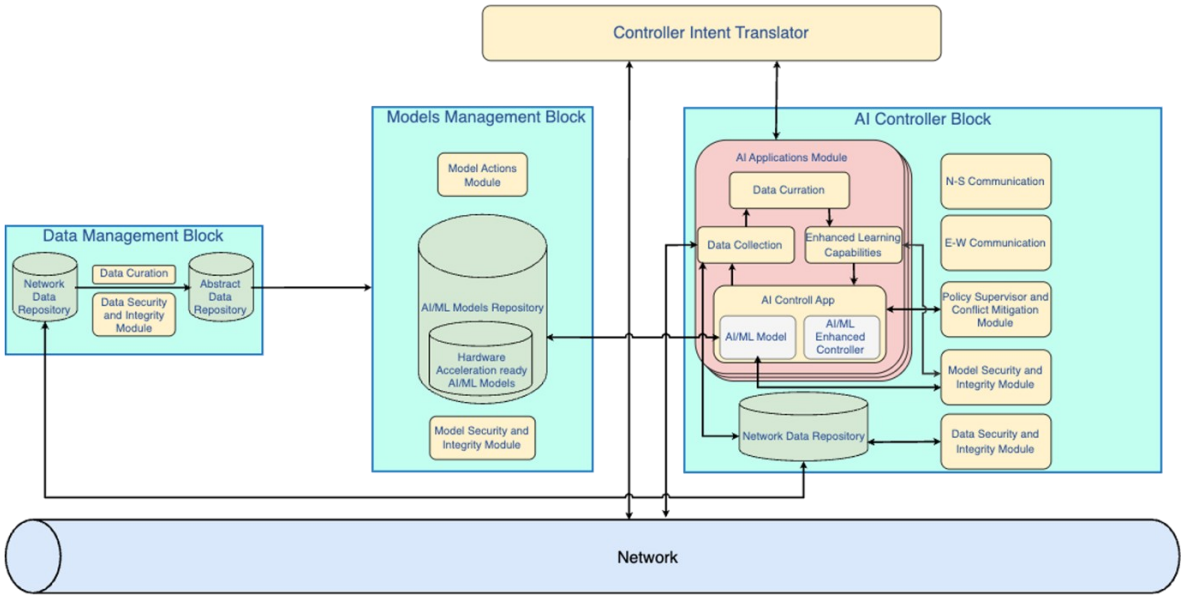


Figure 7: DMMAI High-Level Architecture.

The **Data Management Block** facilitates the collection, curation, and abstraction of telemetry and contextual data from the underlying network infrastructure. It includes a dedicated Network Data Repository, an Abstract Data Repository, and is protected by a Data Security and Integrity Module. The data pipeline is designed to support both raw telemetry for real-time AI decision-making and pre-processed insights for long-term optimization strategies.

The **Models Management Block** is responsible for handling the lifecycle of AI/ML models used throughout the system. This includes a Model Repository for storing both baseline and hardware-optimized AI/ML models, as well as a Model Actions Module for deploying, updating, or retracting models as needed. Another feature of this block is its Model Security and Integrity Module, ensuring the trustworthiness of models before they are integrated into the network. This separation supports experimentation with FL, adaptive models, and accelerates inference at the network edge without compromising security or performance.

The **AI Controller Block** integrates the core intelligence of the framework. It includes capabilities for AI model execution, real-time decision-making, inter-controller communication, and policy enforcement. This block represents the multiple controllers deployed along the network and closer to the end user. The modularity of the AI Controller enables flexible deployment and composability, supporting advanced orchestration and learning strategies across domains. The block integrates a central AI Control App, which interacts with AI/ML models and enhanced control logic. Bidirectional communication is supported via North-South (N-S) and East-West (E-W) interfaces, facilitating coordination across hierarchical and peer controller instances. The Policy Supervisor and Conflict Mitigation Module ensure operational harmony when multiple AI agents interact across domains. Importantly, this block also includes its own security components, ensuring that both data and models remain verifiable and tamper resistant. The E-W and N-S communication interfaces are essential for ensuring interoperability across different domains and layers of the network, facilitating scalable and responsive AI-driven operations.

2.5 AI-Native Multi-Domain Resource Orchestration

The backend systems of 6G services across heterogeneous multi-cloud and edge domains must be dynamically deployed across a distributed cloud-edge continuum, and some of their components may need to remain close to the data sources to meet bounded latency, limited bandwidth, and reliability constraints. These backend systems are inherently cloud-native, composed of microservices distributed across heterogeneous cloud, edge, and far-edge domains. As users and devices move, or as service demand fluctuates, many microservices may migrate, scale, or be re-provisioned in real time. This places enormous complexity on service operators, who must ensure optimal placement,

resource allocation, service mobility, observability, and QoS continuity, across multiple administrative cloud domains. Traditional orchestration solutions operate within single-cluster boundaries, requiring manual coordination of infrastructure provisioning, application placement, and workload migration when dealing with distributed environments. What is needed is a fully autonomous, multi-domain orchestration framework capable of managing the entire lifecycle of cloud-native applications across the edge-cloud continuum, while adapting to dynamic network conditions, user mobility, and changing service requirements. Based in this interpretation, the MIRO[6] - Multi-Domain Intelligent Resource Orchestration – framework, illustrated in Figure 8, is discussed below.

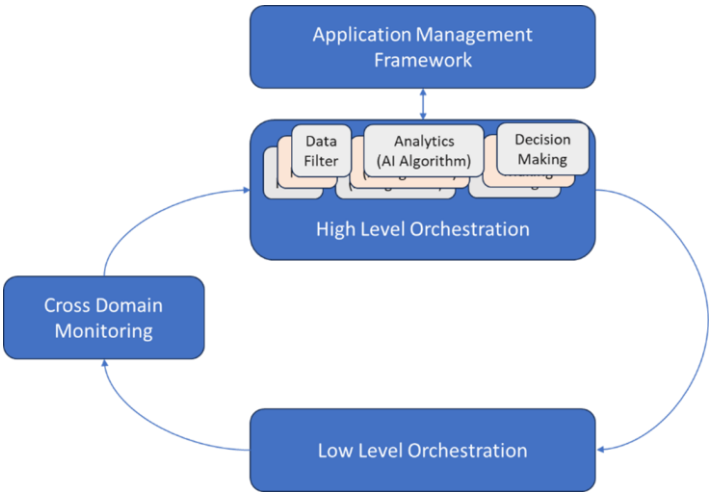


Figure 8: MIRO high-level architecture: Decoupled intelligence and enforcement layers enabling modular, AI-native orchestration by design.

As shown in Figure 8, MIRO introduces a two-layered architecture explicitly designed for autonomy, scalability, and AI-driven decision-making:

High-Level Orchestrator (HLO): AI-Driven Decision and Coordination Layer

HLO is the intelligence core of MIRO, enabling closed-loop and cross-domain orchestration. It hosts the entire decision-making pipeline, including service placement, resource allocation, mobility management, migration policies, and cross-domain optimisation. HLO operates based on closed loops, as defined in ZSM by ETSI [15]; furthermore, it observes system and network states, reasons over them using AI-driven models, and issues actionable decisions that the underlying layer (i.e., the LLO) must enforce.

Low-Level Orchestrator (LLO): Multi-Cloud Execution and Enforcement Layer

6 As part of the 6G-PATH project: https://6gpath.eu/ available from OneSource [Online]: https://miro.onesource.pt/en

LLO interacts directly with the underlying infrastructure, including public cloud platforms, private clouds, edge clusters, and far-edge nodes. It enforces the decisions issued by HLO; deploying microservices, scaling workloads, managing containers and virtual machines (VMs), and provisioning network resources across multiple administrative domains.

A core design aspect of MIRO is the decoupling of intelligence from execution. This separation ensures that orchestration logic can evolve independently of the infrastructure interfaces, enabling rapid innovation in AI-driven control while maintaining compatibility with diverse cloud environments.

Supporting Components

MIRO also includes:

- **Application Management Framework** for interaction with the MIRO users (i.e., 6G service developers, MIRO administrator), supported by a unified intent and policy layer for interpreting user requests/intents and translating them into orchestrator actions.
- **Cross-Domain Monitoring System** to collect metrics, logs, traces, and events from all cloud and edge infrastructure.

Together, these components form a coherent orchestration ecosystem that is inherently extensible, modular, and resilient.

2.5.1 AI-Nativeness in MIRO

AI-nativeness is not an add-on in MIRO; it is architected into the system by design. In this sense, HLO is explicitly designed as a hosting environment for multiple AI algorithms (see Figure 8), each implemented as an independent microservice running in a containerised execution environment (e.g., a K8s pod). This modular design enables MIRO to support a diverse set of intelligence functions simultaneously [24]-[28] such as:

- Cluster creation
- Initial service placement
- Initial resource allocation
- Service migration and mobility
- Service elasticity
- Energy efficiency optimization
- Economic Denial of Sustainability detection and mitigation

As depicted in Figure 8, the algorithms interface with MIRO's cross-domain monitoring system, which filters, aggregates, and contextualises telemetry from heterogeneous infrastructure domains. This ensures that AI components receive only the relevant, pre-processed information required for their respective tasks, avoiding overload and improving inference accuracy.

AI-Driven Decision Output to LLO (i.e., Decision-Making)

Once an AI module filters the data, it generates control actions (e.g., Create / Read / Update / Delete operations on clusters, placement decisions, migration triggers, scaling commands). These actions are then transmitted to LLO for enforcement across distributed edge and cloud environments.

Plug-and-Replace AI Modules

MIRO's AI-nativeness is further reinforced by its ability to:

- Deploy AI modules as containerised services
- Upgrade them dynamically
- Introduce new algorithms without disrupting the orchestration workflow

This design makes MIRO an open, evolvable AI control plane, capable of integrating future advances in AI seamlessly.

Broad Spectrum of AI Techniques

MIRO can host a wide range of state-of-the-art AI techniques [29]-[34] including distributed learning (Swarm Intelligence, FL, Federated Continual Learning), reinforcement learning (Deep Reinforcement Learning, Meta Reinforcement Learning, Actor–Critic Learning, Double Deep Q-Learning), knowledge transfer (Deep Transfer Learning), and model aggregation (Deep Ensemble Learning)

Each technique is used where it best fits the operational problem, enabling adaptive, data-driven optimisation across the entire edge-cloud continuum.

2.5.2 LLM Integration: Toward Human-Centric, Intent-Driven Orchestration

MIRO enables users to manage complex, multi-domain cloud-native systems through natural language, providing an intuitive interface for monitoring, log analysis, and deploying applications. By translating user intents into actionable operations, MIRO simplifies workflows, reduces technical barriers, and delivers insights in clear, human-readable formats.

LLM capabilities in MIRO:

A **conversational application builder** simplifies application deployment by guiding users through infrastructure definition in natural language. Users describe their requirements, and specialised agents capture, organise, and validate this into standardised templates. This approach reduces the operational complexity of deploying multi-component applications across distributed cloud-edge environments.

Intent-driven analytics, allows users to query system performance without requiring knowledge of domain-specific languages. Metrics from infrastructure are embedded in a vector database for

semantic search and intent interpretation. When a user submits a natural language query (such as "What is the average CPU usage for cluster X in the last 24 hours?") the system retrieves relevant metric definitions and context through a Retrieval-Augmented Generation (RAG) process. The LLM then generates a valid Prometheus Query Language (PromQL) query, which is sent to Thanos[7] to federate the request across the underlying distributed metric stores and return the results. The LLM determines the appropriate measurement units, generates dashboard queries via an API, and provides concise, human-readable explanations of the results. This workflow enables both technical and non-technical users to monitor system performance and make informed decisions based on actionable insights.

Explainable logs and insights transform raw log data into human-readable explanations. Infrastructure logs are collected via Grafana Alloy[8], enriched with metadata, indexed in Grafana Loki, and presented in Grafana dashboards. The LLM interprets logs in real time, summarising key events, highlighting anomalies, and identifying probable root causes. By generating human-readable explanations, the system allows users to quickly understand system behaviour and detect issues, transforming log analysis into an interactive and intuitive process.

Intelligent context-aware assistant, which provides an intelligent context-aware assistant to support documentation and technical guidance. System documentation, API specifications, and internal code are pre-processed, embedded in ChromaDB, and indexed for semantic search. When users submit natural language queries, the system retrieves relevant context, and the LLM generates clear, actionable explanations. This capability allows users to navigate complex technical content, understand architectural workflows, or clarify code behaviour without manually searching or interpreting extensive documentation.

To sum up, MIRO represents a new paradigm in orchestrating cloud-native services across the edge-cloud continuum: AI-native, modular, intent-driven, and designed for the extreme dynamism of 6G applications. By combining multi-domain orchestration, state-of-the-art AI algorithms, and powerful LLM-driven interfaces, MIRO delivers an autonomous, intelligent, and human-friendly platform capable of supporting the next generation of data-intensive, latency-critical, and highly mobile 6G services.

7 https://thanos.io/

8 https://grafana.com/docs/alloy/latest/

3 AI-NATIVE SECURITY AND TRUST FRAMEWORKS

The growing integration of AI-native solutions into future communication networks like 6G will bring forth new capabilities, but also significant security and trust challenges. As AI/ML models are increasingly deployed to support autonomous network management, intelligent orchestration and data-driven optimization, ensuring their robustness, trustworthiness and secure execution, become essential. For example, AI-based components in network systems are exposed to training-time and inference-time threats. At the training stage, attackers may introduce data poisoning or model backdoors, aiming to embed malicious behaviours that activate under specific conditions [38]. In FL setups, model aggregation is vulnerable to Byzantine participants, gradient leakage, and malicious update injection, which may compromise the integrity or confidentiality of the global model. At inference time, adversarial examples can be used to manipulate AI-driven classifiers, leading to incorrect or unsafe decisions (e.g., misclassification in intrusion detection or service categorization). Additionally, model extraction and membership inference attacks can compromise proprietary models and reveal sensitive training data, even when raw data never leaves the device.

These emerging threats highlight the need for robust AI-native security and trust frameworks for 6G networks. Hence, this chapter presents AI-native security and trust frameworks for 6G networks, spanning different topics such as AI-Orchestrated E2E Automated Security Management, AI driven Trust-aware Mechanisms, Trusted and Isolated Execution of AI Workloads, AI based Distributed Privacy-Preserving Mechanisms and AI-Enhanced Physical-Layer and Signal-Level Security. Together, these approaches illustrate how AI enables proactive, adaptive, and zero-touch security across the full network stack.

3.1 AI-Orchestrated E2E Automated Security Management

End-to-end Security Orchestration aims to provide a way to apply security policies in different parts of the network, that are managed by different Security Orchestrators. It involves the use of a higher-level Security Orchestrator, named E2E Security Orchestrator (SO), with awareness of the different domains that manages, each one with also the presence of one (or more) SOs. On top of that, an AI-based meta-orchestration framework is proposed for multi-domain 6G deployments to manage Virtual Network Function (VNF) deployment and configuration. The SO includes the meta-orchestration process, as well as the use of AI algorithms to find the most suitable orchestration for each scenario. The meta-orchestration process involves three core phases: first, selecting the most suitable orchestration strategy based on real-time policies and infrastructure constraints; second, constructing a deployment graph of feasible configuration options; and third, selecting the most

appropriate AI-based orchestration algorithm from a pool of candidates. This approach is intended to maximize the effectiveness of VNF allocation.

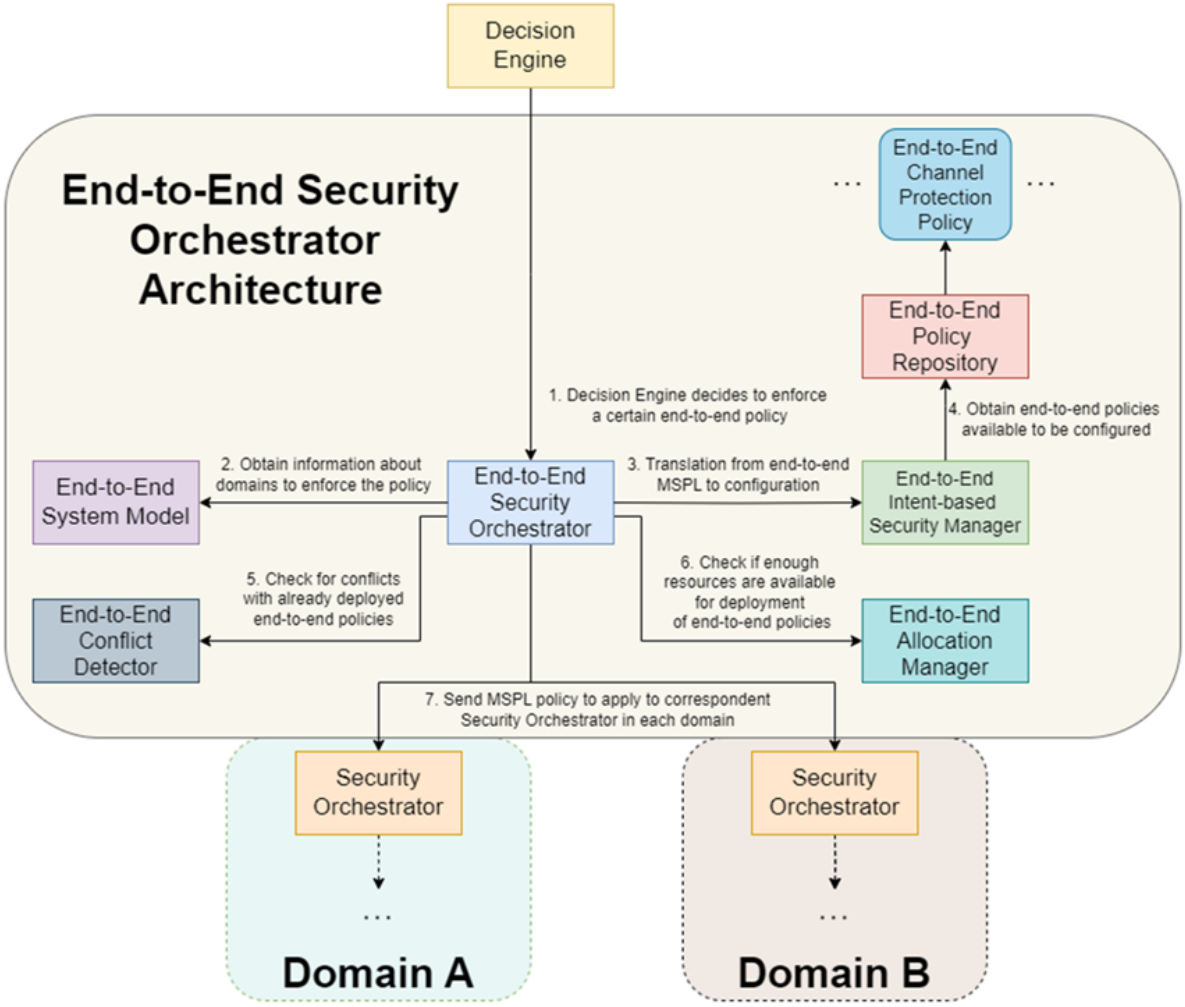


Figure 9: End to End Security Orchestrator Architecture.

The E2E AI-driven SO framework in Figure 9, aims to provide a way to apply security policies in different parts of the network that are managed by different SOs. It involves the use of a higher-level SO, named E2E SO, with awareness of the different domains it manages, each one with the presence of one (or more) SO. This high-level orchestrator receives E2E Medium-level Security Policy Language (MSPL) security policies with information related to the enablers that need to be configured, the type of E2E policy to apply, and other orchestration-related data, such as the name for the rule, the domains affected, or the priority of application. The E2E SO will create individual MSPL messages for each SO in the target domains, containing all the information related to the configuration to apply in the corresponding enabler that the SO manages. Then, each SO will receive the policy and generate the actual configuration for the final enabler, thus applying the policy to the endpoints in each domain. Figure 9 depicts the connection between these components and the E2E SO, which are:

- **E2E System Model**: It stores all the information related to the environment where the E2E SO is deployed, including the domains it manages, the SOs under these domains, and their endpoints to enforce E2E policies.
- **E2E Intent-based Security Manager**: This component is in charge of the translation of E2E MSPL policies into policies that are sent to the SOs in charge of each domain. It communicates with the E2E Policy Repository component in order to obtain the set of available policies to apply.
- **E2E Policy Repository**: It acts as a repository of E2E policies that are available to use for the SOs to apply in their corresponding domains like E2E channel protection, firewall configuration etc.
- **E2E Conflict Detector**: It provides a simple way to check if a conflict exists between the already applied E2E policies before applying for a new one.
- **E2E Allocation Manager**: This component is used only in policies that involve deploying new components in the architecture, as it is in charge of checking if the domains have enough resources to provide for the new component.

Furthermore, the AI-based meta-orchestration framework is proposed for multi-domain 6G deployments to manage VNF deployment and configuration leveraging AI algorithms to find the most suitable orchestration for each scenario. The meta-orchestration process involves three core phases (as can be seen in Figure 10): (1) selecting the most suitable orchestration strategy based on real-time policies and infrastructure constraints; (2) constructing a deployment graph of feasible configuration options; and (3) selecting the most appropriate AI-based orchestration algorithm from a pool of candidates. This approach is intended to maximize the effectiveness of VNF allocation.

The AI based Meta-Orchestrator supports an extensive list of optimization algorithms, including Mixed Integer Linear Programming (MILP), Ant Colony Optimization, Greedy, Simulated Annealing, and Branch and Bound, and is designed to accommodate additional algorithms as needed. The selection of the optimal algorithm is driven by the dual objectives of minimizing response time and maximizing placement efficiency, guided by a predictive Decision Tree model trained on historical execution data.

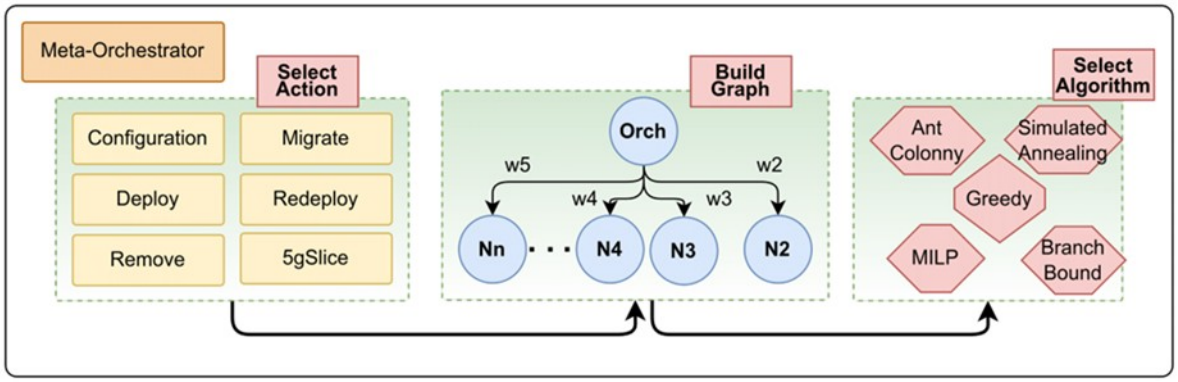


Figure 10: Illustration of the key three steps of the AI-based meta-orchestration process.

The detailed steps to perform meta-orchestration are:

- **Select allocation strategy**: the first decision point is to determine the most appropriate allocation strategy, depending on the capabilities specified in the security policies. This could involve deploying new services, re-configuring, removing or redeploying them. The choice of action considers the type of service, its current state and the load of the deployed instances. After receiving a policy, the SO validates if it is feasible to perform it (checking if service is available, if there are enough resources to allocate, etc.). In most cases, configuration enforcement or update is the most appropriate solution as it takes less time and resources. Actions such as deletion, redeployment or migration are often explicit in the policy.
- **Construct candidate graph**: the candidate graph represents the possible deployment or configuration options. In this graph, nodes represent the workers with their specific characteristics, or the instantiated software/enablers ready to be configured. Each node also has a candidate enabler, a cost associated with that enabler, and the edge weights representing connectivity. The construction of this graph provides a structured view of the orchestration options, serving as input to the orchestration algorithms, that will use it to determine the most appropriate enabler for deployment and its associated cost.
- **Select orchestration algorithm**: the final decision-making is selecting the most suitable orchestration algorithm based on the characteristic of the graph and the requirements of the policy. Currently, SO supports MILP, Ant Colony, Greedy, Simulated Annealing, and Branch and Bound algorithms. To choose the most suitable one, fastest response and maximum placement optimization (depending on policy requirements and infrastructure status) are considered. This information serves as input for the set of models designed to select the most suitable algorithm to use, which are designed with a specific objective, balancing speed and optimization. Once the allocation action, graph and algorithm have been determined, the

meta-orchestration layer ensures that orchestration receives a specific action to perform, a set of potential candidates, and an algorithm aligned with the specific objectives.

Instead of relying in pre-configured options to select orchestration candidates, the AI-based meta-orchestration provides a more fine-grained orchestration process, ensuring that more information is taken into account to select the most suitable solution for the application of a security policy, such as policy requirements, the status of the service or enabler, or the load of the deployed instances.

To achieve true E2E orchestration, the security management framework must be deeply integrated with the network's cloud-native fabric. This integration is enabled by a standardized AI/ML lifecycle management layer that spans the core, edge, and RAN. This layer facilitates the deployment, continuous training, and inference of specialized security agents, such as anomaly detectors, policy optimizers, and forensic analysers, across a distributed and heterogeneous infrastructure.

The security orchestration logic can be implemented using a risk-aware, constrained decision-making paradigm, mirroring the actor-multi-critic architecture. In this context, the "actor" becomes a SO that selects mitigation actions (e.g., isolate a network slice, adjust firewall rules, trigger deep packet inspection). The "critics" are distributed AI models that separately evaluate the predicted security effectiveness (e.g., reduction in attack surface) and the operational cost of each action (e.g., added latency, energy consumption, or SLA impact). By employing distributional critics, the system can account for the inherent randomness in threat behaviour and network state, allowing security policies to be tuned for different risk postures (e.g., high assurance for critical infrastructure slices versus best effort for others).

A key enabler for this vision is a FL plane that allows security models to be trained collaboratively across domains without sharing raw sensitive data. This is critical for building robust threat intelligence while preserving privacy and complying with regulatory boundaries. Furthermore, the framework supports intent-based security interfaces, where high-level security goals (e.g., "ensure integrity for Ultra-Reliable Low-Latency Communications (URLLC) slice") are translated autonomously into low-level, context-aware policies and resource allocations.

Finally, the closed-loop automation is completed by real-time telemetry collection and a scalable event-driven policy engine. This engine can react to threats within milliseconds by leveraging cloud-native scalability, ensuring that the computational load of AI-based security does not become a bottleneck which is a significant advantage over more computationally heavy alternatives

3.2 AI driven Trust-aware Mechanisms

Beyond security, trustworthiness is a foundational property of AI in critical infrastructure. This includes explainability (understanding how AI reaches decisions), fairness, and accountability. As AI takes over decision-making in service provisioning and resource management, network operators and regulators must be confident that AI behaves as expected, especially in edge and zero-touch environments. Trust must be built into the entire lifecycle: training, deployment, monitoring, and governance. In AI-driven network control, trustworthiness therefore extends beyond individual models to the systems and processes that govern how AI decisions are produced, validated, and acted upon.

Similarly, attacks on Network Function Virtualization Management and Orchestration (NFV MANO) or the operator network can lead to orchestration compromises, policy violations, and ultimately service disruptions. While access control mechanisms provide some security, current Network Functions Virtualization Orchestrator (NFVO) implementations lack the capability to detect anomalous policies, potentially resulting in resource depletion and outages [39]. To address this challenge, we propose a novel Policy Verification Function (PVF) within NFV MANO framework, illustrated in Figure 11, as part of Secure E2E SO, adapting the zero-trust principles as a trust-aware mechanism for continuous verification and control in the process of policy driven orchestration.

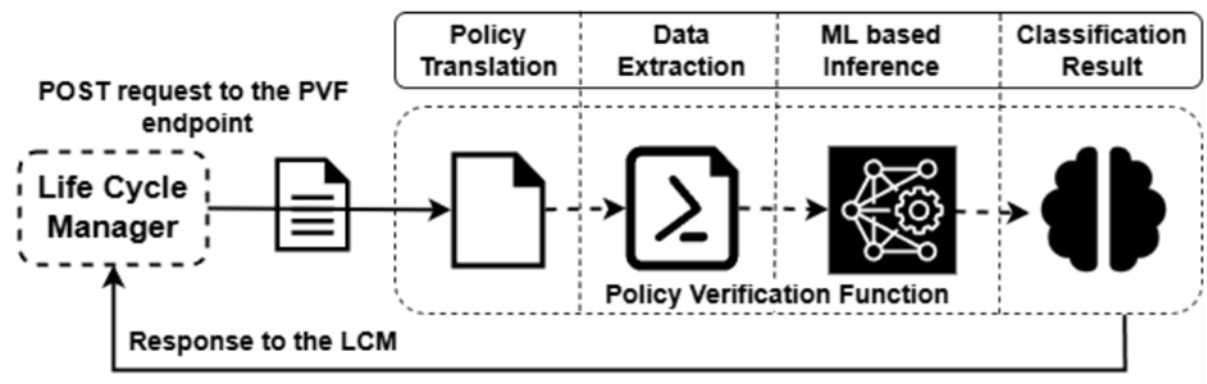


Figure 11: Design of the Policy Verification Function

The PVF enhances the secure service orchestrator integrated in Open-Source MANO (an ETSI Software Development Group) where the policies are verified based on their parameters (such as replica count, CPU, memory allocation, region, intent etc.) of the VNFs or Network Slice using an AI/ML approach to detect anomalous policies. Key approaches in general anomaly detection can be achieved by deep learning techniques such as auto encoders or variational auto encoders, as well as lightweight and traditional ML approaches such as One Class Support Vector Machine (OCSVM), Isolation Forest (IF), Local Outlier Factor (LOF), etc [39]. As the PVF adds an extra layer of security to the process of orchestration by verifying the policies and results in additional latency, the inference time for the

OCSVM is also observed to be significantly lower than the other algorithms making it ideal choice for the given scenario [39]. The PVF is triggered for any CRUD (Create, Read, Update, Delete)) operation or security configurations and mechanism such as deployment of security control functions, implemented as a Python flask server. This module receives POST requests from the Life Cycle Manager (LCM) with inputs as policies needed for verifications. In response, the PVF sends the decision based on each VNF or NS to the LCM with decisions if the policy is anomalous or not. Such scenarios may occur during any security configuration update of the VNFs or when the VNFs need to be scaled or de-scaled and the policies for those are misconfigured. To accommodate detection of such anomalies before the service is orchestrated, the PVF has modules as listed below:

- **The Policy Translation Module:** Based on the received request, the Policy Translation Module formulates a low-level policy description from the play books received as input from theLCM.
- **Data Extractor:** Data values are extracted from the formulated low level polices in the previous step as structured data for the pre-trained ML model.
- **The ML Module**: The ML Classifier generates inference on the received data in the form of vectors for every input from the bundle of policies.
- **The Classifier Module:** The Classification Result is generated in the form of JSON with the inputs from the AI/ML model with the granular report of each VNF.

The integration of AI-driven solutions like the PVF represents a significant step forward in enhancing the robustness, security and efficiency of telecommunication networks, paving the way for more resilient and adaptive infrastructures in the face of emerging challenges.

3.3 Trusted and Isolated Execution of AI Workloads

TEEs provide a hardware-based security boundary for sensitive computations. They allow AI models and data to be processed in isolated enclaves, protecting them even if the host operating system is compromised. TEEs further support remote attestation, enabling verification of the integrity of the executing environment, particularly important in untrusted edge deployments.

WebAssembly Community Group (WASM)[9] is a portable, low-level bytecode format and runtime ecosystem designed to enable safe and efficient execution of code across heterogeneous platforms [34]. It provides software-level sandboxing through a constrained and well-defined execution model, which limits direct access to the host system and reduces the attack surface of deployed workloads.

9 WebAssembly Community Group. WebAssembly Core Specification. W3C Candidate Recommendation Draft, 30 April 2026. https://www.w3.org/TR/wasm-core-2/ (official standard citation by W3C)

WASM enables isolated and portable execution of AI services, with performance characteristics and runtime overhead depending on the selected runtime implementation and workload profile.

In addition, WASM supports fine-grained control over system resources and interactions with the host environment via explicit, capability-based interfaces. This makes it well suited for deploying AI components in distributed and resource-constrained environments, either as a standalone isolation mechanism or in combination with hardware-based security technologies such as TEEs. Together, TEEs and WASM enable a layered and adaptable execution model for AI workloads, allowing security mechanisms to be aligned with heterogeneous hardware capabilities and deployment constraints.

3.4 AI based Distributed Privacy-Preserving Mechanisms

FL has introduced new challenges, such as model poisoning, data leakage, and the need for robust aggregation mechanisms. To mitigate these, a range of privacy-preserving mechanisms are being explored, such as:

- Differential Privacy (DP): Introducing calibrated noise into model updates to limit the leakage of information about individual participants, while requiring careful tuning to balance privacy guarantees and model accuracy.
- Secure Multi-Party Computation (SMPC): Enabling collaborative computation across multiple parties without revealing individual inputs, at the cost of increased communication overhead and coordination complexity.
- Homomorphic Encryption (HE): Allowing computations to be performed directly on encrypted data, providing strong confidentiality guarantees but often incurring significant computational and latency overheads that limit scalability.
- TEEs: Providing hardware-isolated execution for sensitive computations, protecting data and models from the host system. At the same time, TEEs introduce operational constraints, including limited enclave memory, attestation and key management complexity, and the need to address potential side-channel risks.

3.5 AI-Enhanced Physical-Layer and Signal-Level Security

In contrast to higher-layer attacks, physical layer threats directly target the radio medium, making them difficult to detect and mitigate using conventional security mechanisms. The highly dynamic, virtualized, and multi-tenant nature of 6G further amplifies this challenge. As a result, physical layer security is increasingly recognized as a foundational pillar of future network resilience. This challenge is addressed by exploring AI-driven physical layer security mechanisms, with a strong emphasis on

jamming detection, localization, and mitigation. The proposed approaches leverage machine learning and deep learning to enable autonomous, adaptive, and real-time defence capabilities that align with the vision of zero-touch, self-protecting 6G networks.

3.5.1 Jamming Detection in 6G Networks

Jamming attacks deliberately disrupt wireless communications by injecting interference that overwhelms legitimate signals [35]. In future 6G environments, such attacks can potentially target not only the air interface, but also virtualized radio access components, fronthaul links, and control signalling. Jammers may operate continuously or adaptively, exploiting protocol characteristics and environmental dynamics to evade detection. Traditional jamming detection approaches often rely on fixed thresholds, protocol-specific features, or Supervised Learning (SL) models trained on labelled attack data. These methods struggle to generalize across technologies, frequency bands, and attack strategies, particularly in highly dynamic and heterogeneous 6G environments.

To address these challenges, a novel approach called JAmming detection method baSed on Modulation scheme IdeNtification and Outlier Detector of un-jammed data (JASMIN) is introduced [36]. JASMIN is designed to operate effectively across a broad range of network protocols and jamming scenarios, including constant, periodic, and reactive attacks. Unlike traditional methods, JASMIN relies solely on unjammed data during its training phase. This design eliminates the need for jamming data, enhancing its adaptability and applicability in real-world environments.

JASMIN is a real-time jamming detection method designed for modern wireless networks, including B5G and 6G applications. It employs a two-stage decision process, integrating Modulation Scheme Identification (MSI) and an Outlier Detector (OD) to effectively identify jamming attacks.

- **Modulation Scheme Identification (MSI):** The MSI module determines the modulation scheme (e.g., BPSK, QPSK, 16-QAM, 64-QAM) of the received signal by analyzing sequences of I/Q samples. Implemented using a deep learning architecture, MSI ensures high classification accuracy across varying SNR conditions and continuously identifies the modulation scheme employed by the transmitter.
- **Outlier Detector (OD):** The OD module quantifies channel noise through anomaly detection. For each modulation scheme, a dedicated OD model learns the characteristic noise profile of unjammed signals. By comparing the received signal's constellation points with those expected from an ideal, noise-free signal, OD detects deviations that indicate jamming.

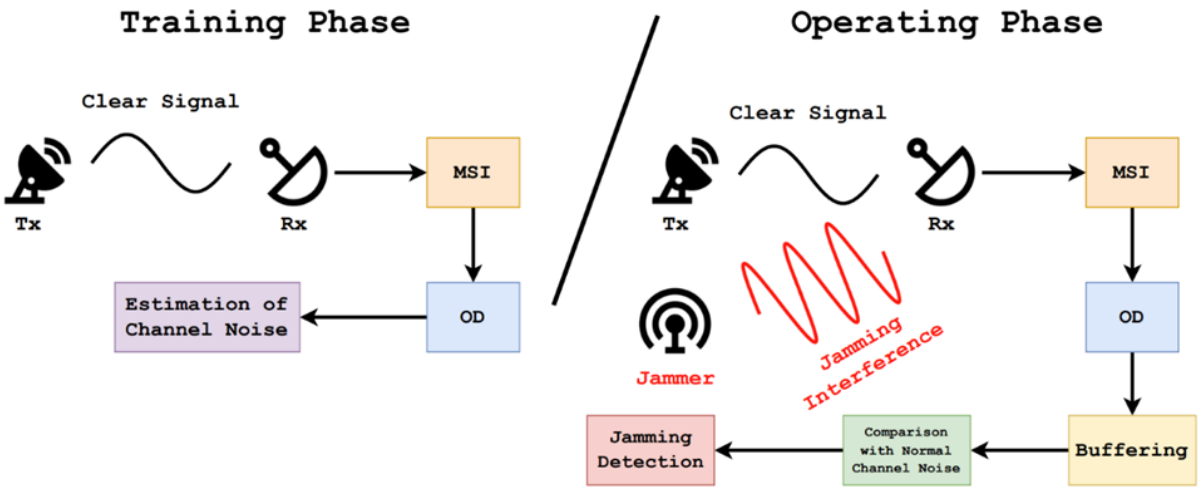


Figure 12: The training (left) and the operating phase (right) of the JASMIN method.

Over the operational phase, illustrated in Figure 12, the receiver continuously accumulates I/Q data over a predefined time window (T). Once sufficient data is collected, the MSI module generates multiple predictions (P) to determine the most likely modulation scheme. Simultaneously, the OD model assesses channel noise across successive data segments. A jamming alert is triggered if either of the following conditions is met:

- Inconsistencies appear in MSI predictions over the observation period, or
- The OD detects noise levels that exceed the expected baseline.

3.5.2 AI-Enabled Anti-Jamming Strategies in Virtualized RAN

As 6G builds on disaggregated and virtualized RAN architectures, including O-RAN (Open Radio Access Network), physical layer security must extend beyond standalone detection algorithms. This explores how AI-enabled jamming mitigation can be integrated into programmable RAN control frameworks, enabling coordinated and adaptive responses. One line of defence enhances traditional Adaptive Modulation and Coding (AMC) mechanisms by incorporating multiple physical and link-layer indicators. Instead of relying solely on channel quality indicators, the system correlates metrics such as signal strength, channel state information, hybrid ARQ feedback, and block error rates.

By jointly interpreting these indicators, AI-assisted logic can distinguish between natural channel degradation and intentional interference. When jamming is suspected, the system proactively adapts transmission parameters to improve robustness and maintain connectivity. This approach is lightweight, backward compatible, and well suited as a first line of defence in 6G deployments.

3.5.3 AI-Driven Jamming Detection and Mitigation via RAN xApps

To support more sophisticated scenarios, the concept of a **Jamming Detection and Mitigation xApp** deployed in the Near-Real-Time RAN Intelligent Controller (RIC) is explored in this section. This xApp

continuously monitors radio metrics from distributed RAN nodes and enforces mitigation policies through standardized control interfaces.

Two complementary operational modes are envisioned:

- **Rule-Based Mode**, which uses predefined thresholds to trigger mitigation actions with minimal complexity.
- **ML-Based Mode**, which applies Unsupervised Learning (UL) to classify jamming severity and dynamically adapt scheduling and modulation policies.

In the rule-based mode, the applies threshold logic such as flagging jamming when BLER (Block Error Rate) exceeds 10% for example, or when CQI (Channel Quality Indicator) drops abruptly triggering mitigation actions that limit Modulation and Coding Scheme levels for specific User Equipments. This mode is deterministic, simple to deploy, and does not require prior training or data labelling.

For environments where jamming patterns may evolve or exhibit non-linear characteristics, the **ML-based mode** approach introduces intelligence through unsupervised learning techniques like clustering or anomaly detection. These models analyze historical metric trends to identify outlier's indicative of jamming and classify the severity level. The xApp then dynamically enforces adaptive scheduling policies based on this classification. It also incorporates a feedback mechanism to adjust its behavior over time, thereby improving its resilience to new or stealthy jamming methods. Although more complex, this mode is especially useful in high-mobility and dense deployments, or adversarial environments.

Therefore, the xApp utilizes a hybrid approach, combining deterministic rule-based triggers with unsupervised machine learning to detect and mitigate evolving jamming threats through closed-loop adaptive scheduling. This forms an example illustrating how AI-driven physical layer security can be embedded into the control fabric of 6G, enabling coordinated, network-wide responses rather than isolated countermeasures.

3.5.4 Leveraging UE-Reported Metrics for AI-Based Detection

In addition to network-side signal analysis, exploiting User Equipment (UE)-reported radio metrics as a complementary source of information for jamming detection has been investigated. Commercial devices already report indicators such as channel quality (Channel State Information - CSI), and Signal-to-Interference-plus-Noise Ratio (SINR), among others, that can be observed, making them attractive inputs for lightweight AI models, because these metrics are easy to measure and light to analyse. Modern User Equipment (UE) provide CSI logs, including Channel Quality Indicator (CQI), to indicate the channel quality.

The main challenge lies in distinguishing malicious interference from benign channel impairments caused by mobility or environmental factors. AI-based dimensionality reduction and anomaly detection techniques are being explored to extract robust indicators of jamming from these measurements. This line of work supports the vision of distributed, collaborative security, where both network and device perspectives contribute to threat detection.

3.5.5 Adaptive Frequency Strategies Enabled by AI

Frequency agility is a powerful countermeasure against jamming. Inspired by adaptive frequency hopping techniques, it explores AI-enabled mechanisms that dynamically reallocate frequency resources when interference is detected.

Although such mechanisms are not natively supported in current O-RAN specifications, AI-controlled xApps can emulate similar behaviour by monitoring radio conditions and reconfiguring bandwidth parts or subbands in real time. This capability aligns closely with the expected flexibility of 6G radio systems and highlights the role of AI in bridging architectural gaps.

To move beyond current emulation workarounds, future Open-RAN (Open Radio Access Network) specifications must undergo structural adaptations across the Near-Real-Time RIC (RAN Intelligent Controller) and the underlying E2 interface protocols to natively support dynamic frequency hopping (E2 is the term for the Open Interface of the Near-Real-Time RIC).

In current Open-RAN architectures, the E2 interface lacks the specific E2SM (E2 Service Models) required to issue direct, sub-millisecond physical-layer frequency reconfiguration commands to the O-DU (Open Distributed Unit) and O-RU (Open Radio Unit). To fully unlock AI-driven frequency agility, the ecosystem must transition through the following architectural advancements:

- **Standardization of Security-Specific E2 Service Models (E2SM-SEC):** Future frameworks must introduce dedicated service models that expose real-time, low-level physical layer metrics such as CSI (Channel State Information), raw SINR (Signal-to-Interference-plus-Noise Ratio), and BLER (Block Error Rate) directly to the Near-Real-Time RIC. This standardized telemetry pipeline allows xApps (Near-Real-Time RIC Applications) to bypass the application-layer processing bottlenecks of current setups.
- **Sub-Millisecond E2 Control Loops:** Traditional Open-RAN control loops operating over the E2 interface function within a 10-millisecond to 1-second window. Dynamic countermeasures against tactical, reactive jamming require sub-millisecond execution times. Future 6G Open-RAN profiles must support accelerated data-plane paths (leveraging technologies like eBPF (Extended Berkeley Packet Filter) hardware offloading) to execute frequency shifts before the UE (User Equipment) connection drops.

- **Dynamic Bandwidth Part and Sub-band Hopping Policies:** The Near-Real-Time RIC must utilize advanced management interfaces to feed long-term, machine learning-derived jamming signature trends down to the Near-Real-Time RIC. Instead of relying on rigid, pre-configured thresholds, the xApp can then dynamically reconfigure active BWP (Bandwidth Part) allocations or shift subband allocations on the fly based on the specific intent classification of the attacker.
- **Frequency Allocation Coordination and Conflict Resolution:** As AI-driven frequency agility xApps manipulate radio resource allocation and spectral mapping in real time, their actions must be coordinated with standard RRM (Radio Resource Management) xApps, such as traffic steering or load balancing. Future Open-RAN frameworks must embed a native conflict resolution engine within the Near-Real-Time RIC to ensure that security-driven frequency hops do not create secondary internal network interference, overlap active channels, or violate active SLA (Service Level Agreement) targets.

By embedding these capabilities directly into the core specification, future Open-RAN architectures will shift from passive, threshold-based monitoring to an AI-native, closed-loop resilient control fabric, making frequency agility a native, plug-and-play security feature for 6G networks.

3.5.6 The DetAction Framework

Integrated Detection-Reaction Paradigm. Traditional approaches treat jamming detection and mitigation as separate processes. DetAction challenges this model by jointly designing both phases within a single AI-enabled system. The detection component identifies not only the presence of jamming but also its spectral location, while the reaction component immediately adapts resource allocation to avoid affected frequencies. This tight coupling reduces reaction time and improves resilience, particularly in fast-changing attack scenarios.

Deep Learning for Spectral Localization. DetAction employs convolutional neural networks operating on frequency-domain representations of received signals. By analysing spectral fragments, the AI model localizes jammed frequency regions with fine granularity. This information enables targeted countermeasures rather than coarse, system-wide adaptations.

AI-Assisted Reaction and Resource Control. The reaction phase maps detected interference to physical resource blocks and informs the scheduler of unavailable resources. This enables reactive frequency hopping and more informed scheduling decisions, effectively shielding ongoing communications from jamming without unnecessary performance degradation.

3.5.7 ML-Based MIMO Defence Mechanisms in 6G

As the telecommunications ecosystem transitions from 5G to 6G, the demand for massive data speeds, near-zero latency, and connecting billions of smart devices requires a fundamental shift in how we

transmit data. To meet these demands, Massive MIMO (Multiple-Input and Multiple-Output) technology plays a critical role in next-generation infrastructures, packing hundreds of miniature antennas into a single base station to transmit multiple data streams simultaneously. By manipulating the spatial dimension of radio waves, Massive MIMO moves away from broadcasting signals blindly across a neighbourhood, instead focusing the network's energy into tight, high-precision beams directed straight at specific user devices.

Massive Multiple-input and multiple-output (MIMO) is a cornerstone of 6G, offering unprecedented spatial resolution and flexibility leverages this high-dimensional signal space to develop ML-based MIMO defence mechanisms that detect, localize, and suppress jamming at the physical layer.

While this spatial resolution offers unprecedented flexibility and network capacity, it also expands the digital attack surface, introducing complex vulnerabilities at the **physical layer** of the network. Traditional 5G security frameworks were primarily designed to protect software interfaces and data streams further up the network protocol stack, leaving the raw radio waves exposed. In a 6G landscape where cellular networks will underpin time-critical infrastructure like autonomous vehicle fleets, smart energy grids, and remote medical systems a malicious entity can deploy tactical, low-power **RF (Radio Frequency)** jamming devices to disrupt these focused beams, creating isolated communication blackouts that can compromise safety and paralyze local operations.

Historically, defending against such radio-frequency threats required complex mathematical calculations and rigid, pre-programmed channel templates that struggled to adapt to changing environments. To bridge this critical architectural gap, modern 6G whitepapers propose a transition to AI-native (Artificial Intelligence) wireless security. By embedding Machine Learning (ML) algorithms directly into the antenna's processing fabric, future networks can analyze massive volumes of raw radio telemetry in real time. Rather than guessing the attacker's method based on generic rules, these data-driven defense mechanisms learn the unique physical features of the local environment. This approach allows the network to distinguish human-made cyber-attacks from natural signal fading, autonomously neutralizing interference and ensuring unbroken service continuity for legitimate users.

ML models analyse spatial features such as Direction of Arrival (DoA) and received power distributions to distinguish legitimate users from malicious interferers. The models could include deep neural networks for DoA estimation with anomaly detection and mitigation modules. By learning from data rather than relying on explicit channel models, these approaches remain robust under dynamic and adversarial conditions.

Practically, the ML models (or in general AI) replace computationally intensive traditional signal processing blocks, enabling real-time operation with reduced complexity. The modular design allows

seamless integration into future 6G base stations and distributed antenna systems. Detected spatial directions are classified as benign or malicious based on learned power and spatial characteristics. Once identified, jamming components are suppressed using subspace projection techniques that remove dominant interference directions while preserving legitimate signals. This process operates autonomously and adapts to changing attack conditions.

Overall, with the approach mentioned above, an ML-based MIMO defence service can be offered to provide autonomous physical layer protection by:

- Detecting jamming without prior knowledge of attack strategies,
- Localizing jammers using AI-assisted DoA estimation,
- Suppressing interference through adaptive spatial filtering,
- Preserving service continuity for legitimate users.

4 FRAMEWORKS FOR AI/ML LIFECYCLE

The evolution and integration of AI and ML into modern computing and network systems have created the need for designing efficient mechanisms to manage the entire lifecycle of ML models.

Under this paradigm, the concept of MLOps has emerged as a natural evolution of the DevOps methodology, extending its principles to the development and management of AI/ML systems. MLOps defines a set of practices and tools that automate and standardize the entire lifecycle of ML models, ensuring that data, models, and software artifacts can be continuously integrated and deployed in production environments.

In contrast to traditional ML workflows, where development and deployment are often separated, MLOps introduces Continuous Integration and Continuous Delivery/Deployment (CI/CD) pipelines for AI models. This approach enables automation, scalability, and governance across distributed infrastructures, such as cloud and edge environments. Through these mechanisms, MLOps facilitates collaboration between data scientists, ML engineers, and operations teams, reducing the time and complexity required to bring AI models to production systems, while ensuring reproducibility, traceability, and security.

A standard MLOps pipeline is typically structured around four main stages:

1. Data management: Acquisition, preprocessing, and feature extraction.
2. Model development and storage: Training, evaluation, storage and version control.
3. Deployment: Serving trained models for inference.
4. Monitoring and retraining: Tracking model performance and triggering updates to prevent drift or degradation.

To manage MLOps pipelines, modular and cloud-native solutions (e.g. Docker and K8s-based) are required to ensure reproducibility, traceability, scalability and interoperability. In this context, open-source solutions like Kubeflow or Airflow are typically used for ML pipeline orchestration, while others like KServe[10] or MinIO[11] are leveraged for cloud-native model serving and object-based storage, respectively. These solutions are usually complemented with deployment and lifecycle management solutions like Helm[12] and ArgoCD[13], and with the exploitation of CI/CD and GitOps principles, aiming

10 https://kserve.github.io/website/

11 https://www.min.io/

12 https://helm.sh/

13 https://argo-cd.readthedocs.io/en/stable/

to ensure an automated, version-controlled delivery of components and continuous integration, testing, and deployment of models and components in a secure and traceable manner.

In the network domain, MLOps enables AI-driven optimization of communication, resource allocation, and service orchestration. It supports the deployment of intelligent functions across the compute continuum, from cloud to edge, ensuring that models can adapt dynamically to changing network conditions, traffic patterns, and user demands.

Within the context of 6G networks, MLOps becomes a fundamental enabler for the realization of AI-native architectures and frameworks. Future 6G systems will rely on adaptive, self-optimizing, and autonomous behaviours supported by distributed intelligence. Implementing robust MLOps techniques and integrating them is therefore essential to manage this intelligence efficiently, while guaranteeing reliability, privacy, and explainability across the full lifecycle of AI models.

As a result, the realization of consolidated frameworks for AI/ML lifecycle becomes essential to address the migration towards a fully AI-native 6G network architecture. These integrate at their core MLOps concepts and techniques and augment them with advanced functionalities for distributed AI workloads management, combined data and AI assets management, AI-enabled federated networks and AI experimentation capabilities within federated infrastructures.

The following subsections provide an overview of illustrative AI frameworks, which are being developed under the SNS JU initiative. They focus on concepts and solutions for distributed and Crowdsouring AI, combined automation of ML and data operations, AI/ML models management across O-RAN deployments, and federation of AI capabilities across multiple infrastructures and testbeds.

4.1 Distributed AI framework for CrowdSourcing AI

CrowdSourcing AI is an innovative vision of the Machine Learning as a Service (MLaaS) approach in distributed systems. The concept enables collaborative training of ML models and the execution of inference tasks through such models, using distributed techniques and enabling a continuous cycle for learning creation, validation, refinement, and transfer of ML models across domains. Crowdsourcing also enables savings in computational power and memory associated with ML training, where a domain requiring an ML model can retrieve a pre-trained model and further refine it using its own data. In addition, the concept uses meta-learning techniques to support discovery and selection of the best available ML model for the execution of a given task and for the training of a model with the aim of establishing the best solution for the amount of energy consumed by ML model training/updating/inference and the quality of the decisions made. The orchestration of ML model

training and inference takes advantage of the number and the multiple types of computing entities (devices, network nodes) as well as the amount of computing and network resources that can be dedicated by such entities to the training and inference tasks.

In this context, a Crowdsourcing AI/ML framework enables a cooperative, energy-efficient approach to AI management and use. The framework covers deployment of AI/ML services across multiple domains, allows an efficient sharing of ML models, and supports the execution of the methods and functions for Crowdsourcing Intelligence in ML model training and inference. In particular, the framework fully supports Crowdsourcing functionalities [37], which are deemed essential to creating a distributed ecosystem for the lifecycle of AI/ML services. This includes integrating a variety of innovative AI/ML techniques, including cooperative and collaborative model training, transfer and FL, intelligent model selection, and intelligent placement and management of AI/ML services in distributed execution environments. Specifically, the main goal of these various solutions is to deploy and manage the AI/ML operational pipeline with minimum cost in terms of computational, network, and energy resource consumption. In the proposed framework, the focus is on implementing ML model training and inference in a sustainable manner, while still guaranteeing the target performance values of the whole AI/ML pipeline. Through this approach, the Crowdsourcing AI framework enables a set of AI/ML services to support 6G network and service management, therefore delivering an AI-native solution that can make the implementation and use of AI/ML models for AI-enabled network services more efficient, as well as more robust, resilient, and efficient. Relevant examples of AI/ML services in support of network and service management include: (i) Traffic prediction, where algorithms analyse historical and real-time data to forecast network traffic patterns, facilitating proactive network resource allocation and minimizing congestion; (ii) RAN configuration, where AI/ML enables dynamic adjustment of network parameters, optimizing performance to adapt to changing demands and conditions, thus improving coverage and capacity; (iii) Path computation, where AI/ML optimizes routing decisions based on network status, latency, and congestion, ensuring efficient data transmission; (iv) Traffic scheduling, which is intelligently managed by AI/ML algorithms, prioritizing critical data flows and managing bandwidth allocation, crucial for maintaining QoS and Quality of Experience (QoE) in networks supporting diverse applications.

In practice, the proposed Crowdsourcing AI framework deploys and manages AI/ML training and inference functions (as shown in Figure 13) via agents on virtualized infrastructures spanning across multiple domains (i.e. from terrestrial and non-terrestrial far edge and cloud. The Crowdsourcing AI framework architecture is broken-down into several functional blocks (as depicted in Figure 14 and summarized below), integrating a mix of both innovative (blue blocks) and state-of-the-art elements.

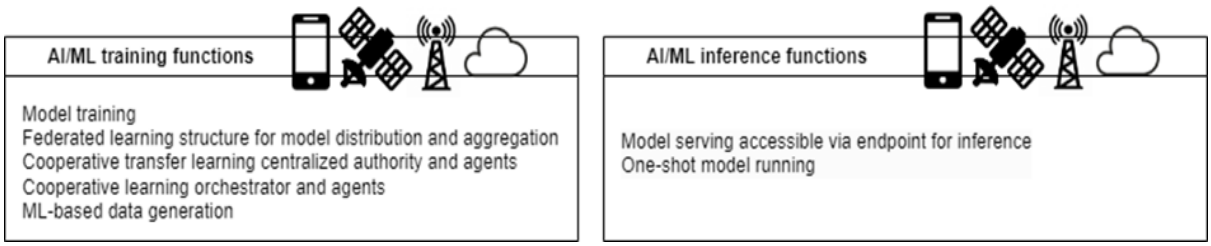


Figure 13: AI/ML training and inference functions

AI/ML Orchestrator: it serves as the central component within the framework, its main functionalities include executing requests for model inference deployment service, model/pipeline/data onboarding, model section, and model training from users. The Orchestrator interfaces with the AI/ML Catalogue for model and training information, the Data Registry for training stat access, and the AI/ML Resource Manager for deployment of training and inference model services. The AI/ML Orchestrator contains sub-modules of the AI/ML Pipeline Manager for the execution of training and inference functions and the AI/ML Orchestration Engine for deciding a service orchestration approach.

Model Selector: it leverages a distributed learning approach that reuses the versions of a model that have been previously trained, and possibly pruned, in the system and that have been stored in the AI/ML model storage. It is implemented through a selection mechanism that leverages pruning techniques and selects and aggregates such versions of the model to create up-to-date ML-based radio models of the right size (hence, complexity level) to fit the computing constraints of the individual edge node. It can deal with both deep neural networks and reinforcement learning models.

AI/ML Orchestrator Engine: it is tasked with determining the orchestration approach for each requested AI/ML task. It is supported in its decision-making processes by the other functional blocks of the AI/ML Framework such as the AI/ML Pipeline Manager and the AI/ML Resource Manager, either by supplying input data or executing directives. The Engine provides the necessary functionalities to deploy the required ML models and deliver the services that the AI/ML Framework may be requested to provide. It integrates a wide set of AI/ML pipelines and solutions for: 1) optimal device-to-device topologies, 2) inter-domain VNF migration through RL agents, 3) migration of models, 4) lost modalities detection, 5) cross-generation of synthetic data with generative models, 6) integration of synthetic data, 7) dependable and fair distributed training, 8) 2-step transfer learning, 9) energy-efficient inference, 10) energy-efficient BDIx agents configuration, 11) energy efficient decentralized learning, 12) energy-efficient FL.

AI/ML Pipeline Manager: it manages ML pipelines in the Crowdsourcing scheme, including pipeline cataloguing, deployment, and tracking. It exploits central and FL schemes, through the onboarding of customized model, data handlers and model training parameters. The Pipeline Manager deploys training pipelines as a set of virtualized functions/applications (e.g. containers and K8s pods) in the

target execution environment based on available resources. Pipelines are integrated with the AI/ML Catalogue for automated model and model artifact tracking. Crowdsourcing users can also perform a model retraining by providing the registered model name and new training parameters/data sources.

AI/ML Catalogue: it serves as a centralized repository for storing and querying ML models within the Crowdsourcing platform. It is designed to store all necessary model information, facilitating the seamless execution of ML services by the AI/ML Orchestrator.

The four other blocks which form the framework include state-of-the-art solutions to provide the needed services and functions of the AI/ML Orchestrator. The AI/ML Resource Manager is responsible for assigning and reserving computational and communication resources necessary for ML algorithm execution. It implements resource allocation decisions set forth by the AI/ML Orchestrator. The Data Registry is responsible for managing metadata, schemas, and data definitions across the network infrastructure. The Data Repository securely stores and efficiently organizes data within a structured environment. The Model Storage consists of a database used for storing the ML models that have been onboarded in the system, either through training pipelines or directly as already trained, or yet to be trained models. The associated model metadata are stored in the AI/ML Model Catalogue.

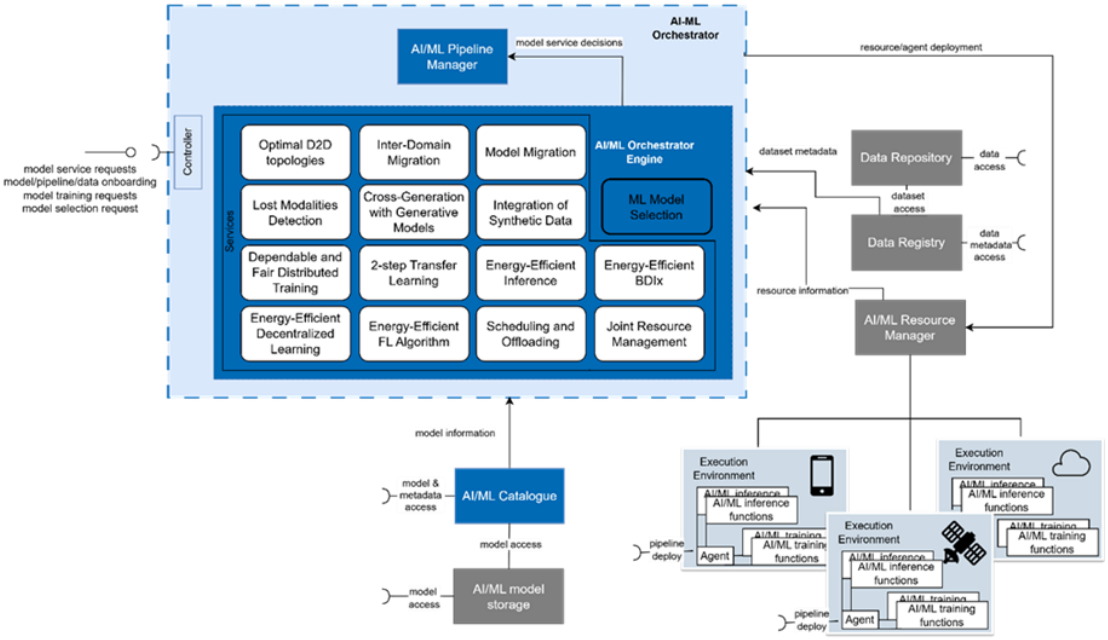


Figure 14: Crowdsourcing AI Framework architecture.

4.2 AI Framework for automating ML and data operations

Native support of AI and ML is one of the pillars of future 6G networks. Despite the opportunities, there are several gaps that are hindering the seamless adoption of AI/ML in 6G. The lack of rich and high-quality datasets needed to train and fine-tune the models, and the need to test and evaluate

them in a representative staging environment while ensuring ethics and regulatory compliance, becomes a challenge without access to an E2E 6G testbed or a representative Digital Twin replica environment.

The increasing computing power available, the scaling laws of ML together with theoretical advances, have driven progress in many areas of AI/ML, such as distributed and FL, reinforcement learning, and more recently with the breakthrough of Generative AI (GenAI) and LLM. However, two major remaining gaps limit the adoption of AI/ML in 6G. First, the need of extensive datasets to train models as well as the quality of their data have a critical impact on the accuracy of the resulting model. For this purpose, the International Data Spaces (IDS)[14] Association and Gaia-X[15] initiatives aim at providing a framework for the secure and sovereign exchange and sharing of data, which can be discovered and used to fuel the lifecycle of ML models. While these initiatives focus on the creation of Data Spaces (DS) for critical sectors such as health, agriculture, and manufacturing, DS efforts for 6G datasets are indeed very limited. Second, when an AI/ML model is developed, testing and validating its performance in a representative environment remains a challenge. While model validation platforms may help to validate a ML model, they have a strong limitation when it comes to representing real and complex environments such as 6G networks and systems. To this end, MLOps can provide a set of reproducible workflows for the development, test and validation of AI/ML models in pre-production 6G testbed environments for performance benchmarking prior to deployment in production.

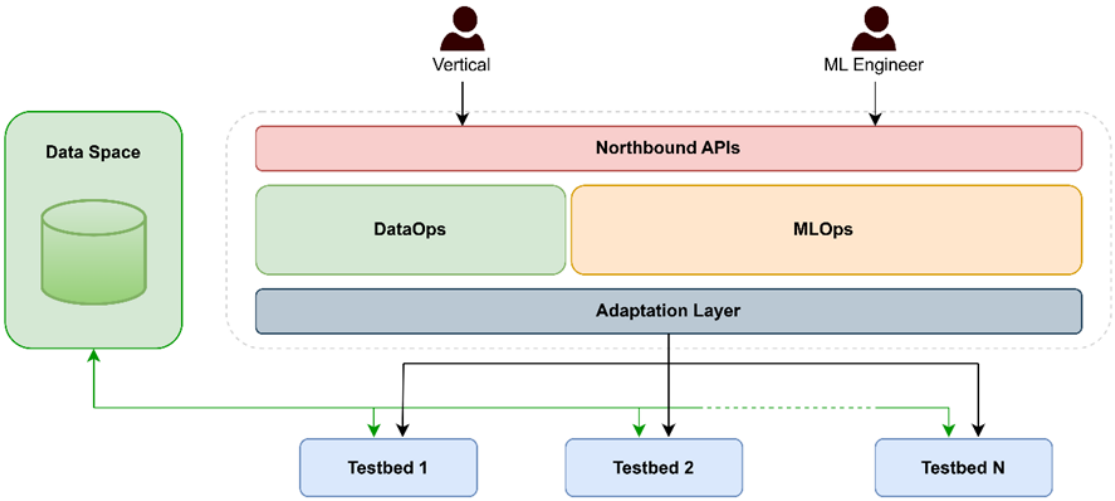


Figure 15: High-Level Architecture of the framework for AI/ML and data experimentation

14 IDSA, “International Data Spaces: Secure Data Sharing,” 2024.. Available: https: //internationaldataspaces.org/

15 GAIA-X, “A Federated Secure Data Infrastructure,” 2024.. Available: https: //gaia-x.eu/

In this context, an E2E solution is required to connect 6G data with vertical industries, ML developers and 6G experimenters, while relying on multiple 6G testbed infrastructures to execute AI and data pipelines. To this end, an efficient, realistic, and trustworthy framework for E2E AI/ML experimentation in support of advanced 6G use cases and services can be realized. As shown in Figure 15, such an E2E AI framework is organized around two interdependent pillars: MLOps and DataOps. To facilitate the integration of testbed infrastructures capable to execute AI/ML and data related experiments, the proposed framework adopts a flexible and modular plugin-based approach.

The proposed AI framework embeds advanced MLOps functionalities to support AI/ML experiments that cover the development, continuous training, testing, validation, hyperparameter optimization, and continuous monitoring of models throughout their lifecycle. Reinforcement Learning Operations (RLOps) and automated hyperparameters optimization are also supported to fine-tune models for optimal use of testbed network and computing resources. Trustworthiness is addressed through the principles of models explainability (XAI) and robustness enabled by their automated testing and assessment via dedicated MLOps functionalities. In addition, the proposed framework makes use of an MLOps meta-orchestrator for enabling unified and coordinated MLOps pipelines across different MLOps stacks, tools and APIs possibly adopted in the various 6G testbed infrastructures. This allows to realize an automated, scalable, and collaborative E2EAI experimentation framework that ensures efficiency, trustworthiness and adaptability for AI/ML services as enablers for AI-driven 6G networks and services.

On the other hand, the proposed AI framework integrates advanced DataOps functionalities to support intent-based and GenAI-enabled 6G datasets dynamic generation and query using natural language. In addition, the DataOps system integrates the concept of ELT (Extract, Load, Transform) enabling the execution of on-demand pipelines to clean, structure augment and categorize data upon requests from experimenters or specific necessities from the MLOps system. In this context, the framework also provides a pan-European 6G Data Space that connects the 6G testbed infrastructures and enable 6G data sharing, ensuring that datasets generated through the DataOps pipelines are easily discoverable, catalogued and stored using standardised IDS approaches in order to be available for 6G experimentation and AI/ML models training and validation. Therefore, the proposed AI framework streamlines the process of data collection, preparation and sharing, and position its 6G Data Space and DataOps solution as a strong foundation for effective and scalable AI/ML model training and validation in 6G networks and systems.

4.3 AI/ML models management across O-RAN deployments

AI and ML play a vital role in enabling automation and intelligence across the O-RAN architectural vision. Through the integration of intelligent control and learning functions in distributed RAN elements, they form the cornerstone of data-driven network optimization and adaptive network management in 6G.

4.3.1 AI/ML Functionalities in O-RAN

The O-RAN proposed framework standardizes three control loops and several AI/ML nodes to support intelligent capabilities and autonomous decisions [41]-[43]. The key O-RAN entities that facilitate AI/ML functionalities are the Non-Real-Time RAN Intelligent Controller (Non-RT RIC) that resides in the Service Management Orchestrator (SMO) layer, as well as the Near-Real-Time RIC (Near-RT RIC). These components provide support for both offline and online model training and inference across various network domains. On the one hand, offline training corresponds to the process of training ML models using historical data, that are typically conducted outside the real-time operation of the network and are usually very time-consuming. In this context, offline training is the key mechanism for deploying pre-trained and optimized models in time-sensitive applications, thus leveraging real-time operation of the O-RAN network. On the other hand, online learning is associated with training of adaptive agents, for example RL models, that continuously learn the network behaviour and adapt immediately to changes. This is achieved through continuous interaction with the environment, feedback mechanisms, and trial-and-errors.

As mentioned before and depicted also in Figure 16, there are two fundamental O-RAN modules for the implementation of the AI/ML workflow. The Near-RT RIC operates at timescales between 10 and 500 milliseconds and performs control actions or data collection processes by interacting with the lower-level RAN components through the E2 interface. Similarly, the logical function of Non-RT RIC resides within the SMO and includes model training, inference and Performance Monitoring (PM) for intent-driven control and updates. In this way, both near-real-time and non-real-time control and optimization of RAN components and resources are enabled. At a more practical level, when it comes to SL or UL models, the training typically occurs in the Non-RT RIC, while the inference can be deployed either in the Non-RT or the Near-RT RIC. As for the RL scenarios, both functions have to be co-located in one of the modules, ensuring efficient learning cycles.

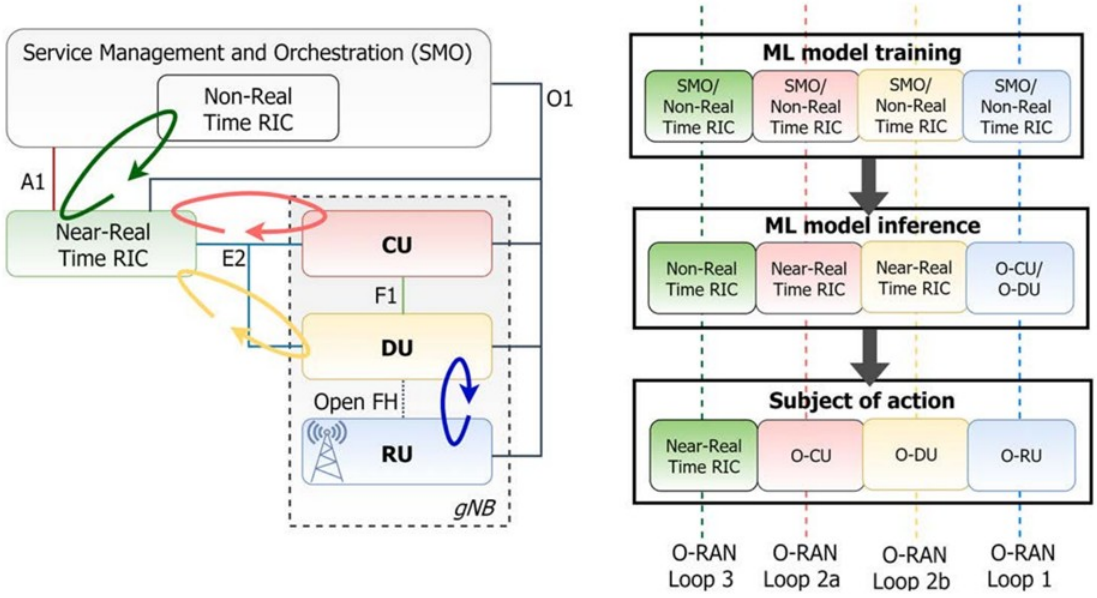


Figure 16: O-RAN Control Loops for AI/ML workflow (derived from [44])

In Figure 16, three control loops are depicted, along with the involved O-RAN modules:

- *Control Loop 1*: It operates at the level of the Transmission Time Interval (TTI) or even below (milliseconds). It is primarily associated with scheduling and resource allocation at the edge, as it hosts the pre-trained models at the O-DU for rapid inference and communicates predictions or actions to the execution node of O-RU [45].
- *Control Loop 2*: It utilizes functions within the Near-RT RIC timescales, focusing on resource optimization and control. For the real-time adaptation it uses the E2 interface.
- *Control Loop 3*: It refers to the Non-RT RIC operations, handling decisions with timescales above 500 ms, such as policy management and network orchestration through SMO interfaces.

4.3.2 Workflow of O-RAN intelligence

This section outlines the workflow of AI/ML integration within O-RAN-based 5G and 6G architectures. A relevant UML diagram is shown in Figure 17. The description below is mainly aligned with Control Loop 2, where an xApp is hosted in the Near-RT RIC to perform intelligence actions. The workflow is divided into six core stages.

Model Construction: The process begins with a human-in-the-loop that designs and programs an ML model using common toolkits such as TensorFlow[16], PyTorch[17], or Keras[18], or via orchestration platforms such as Acumos[19] or AiFlow[20]. Once the model architecture and objectives are defined, a description file is generated and forwarded to the Orchestrator within the SMO. This is the entity responsible for checking the ML Catalogue to determine whether an existing model already meets the intent-define objectives.

Model Training: The next step involves sending the description file from the Orchestrator to the ML Model Trainer, constructing the described model, and initializing the training process for relevant datasets collected by the Data Collector. Hyperparameter tuning and validation are performed using standard tools such as TensorBoard[21].

Model Deployment: If training was successful, the optimized model returns to the Orchestrator to be packaged and containerized (via Docker or MLOps like KubeFlow[22]) and finally stored in the ML Catalogue. Afterwards, the model is also being deployed at the Near-RT RIC, leveraging real-time inference and model reuse regardless of the particular vendor**.**

Model Inference: During network operations, the Near-RT RIC performs continuous inference iterations based on live data collected from the different O-RAN nodes (O-RU, O-DU, O-CU, or eNB). For SL, the input may consist of traffic load measurements, while for RL the input usually represents real-time radio metrics. As a result, the prediction either triggers specific network policies or perform control adjustments.

16 https://www.tensorflow.org/

17 https://pytorch.org/

18 https://keras.io/

19 https://github.com/acumos

20 https://app.ai-flow.net/

21 https://www.tensorflow.org/tensorboard

22 https://www.kubeflow.org/

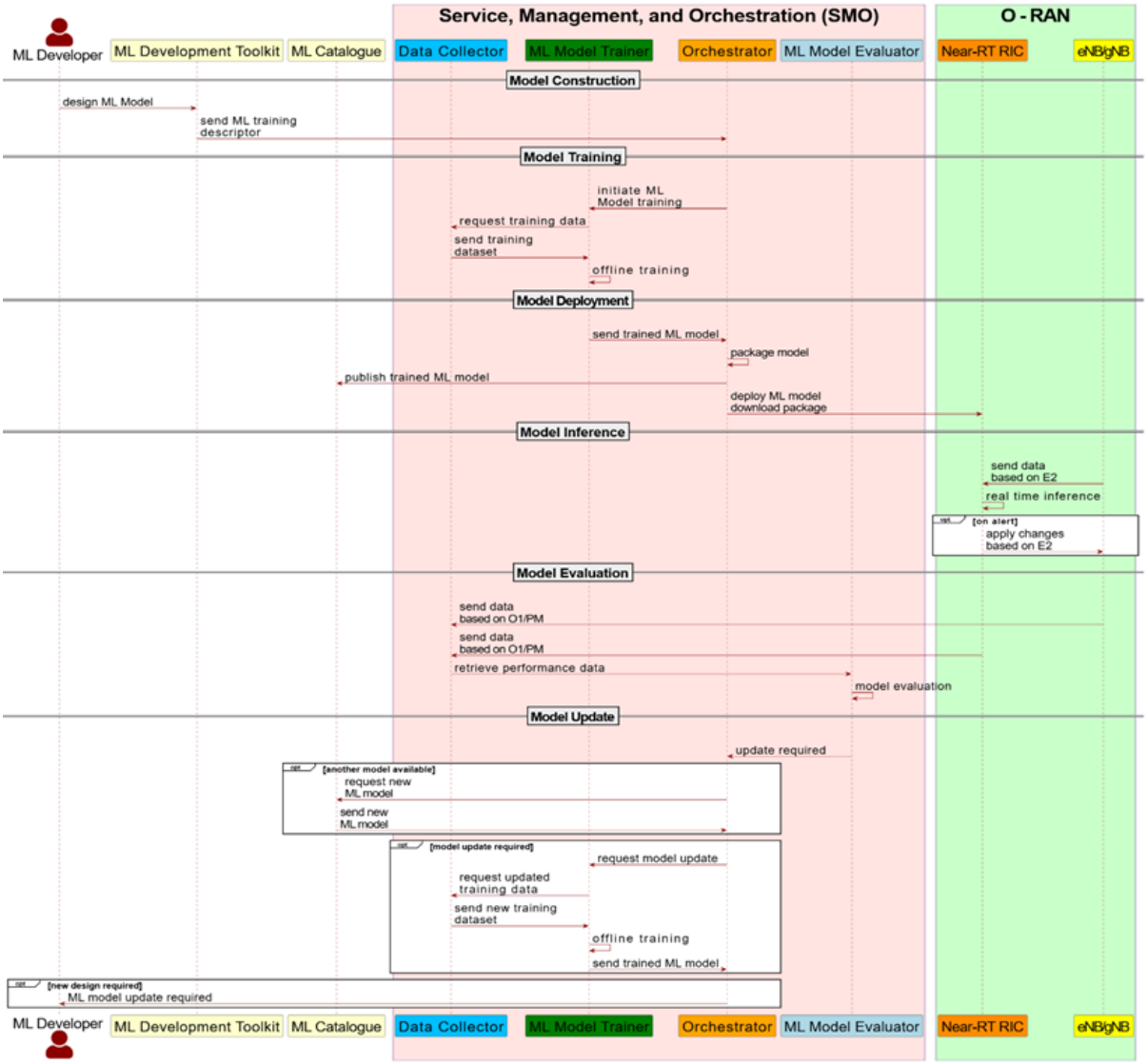


Figure 17: AI/ML lifecycle in O-RAN-based system architectures (derived from [44])

Model Evaluation: In this phase, performance data from RAN components are reported to the Data Collector via the O1/PM interface. The one responsible for analysing this data and determining whether the model continues to meet the targets is the ML Model Evaluator. Otherwise, if a threshold policy is not fulfilled or a trend degradation is detected, a model re-training can be triggered.

Model Update: This stage is meaningful only if re-training is required and is activated by a notification message from the ML Evaluator to the Orchestrator. In this case, the Orchestrator either retrieves a more accurate model from the ML Catalogue or instruct an ML Model Trainer to initiate a new training cycle using an extended and recently updated dataset. More rarely, and if both options fail, a new model design may be required.

The presented workflow can be also adapted to the other two control loops. As far as Control Loop 1 is concerned, inference can be executed directly at the O-DU/O-CU element and with data coming

from the Open-FH interface. In case of Control Loop 3, the Non-RT RIC can host the model and the O1 interface applies the corresponding actions. Finally, hierarchical or collaborative intelligence is also possible, with models from lower loops supervise or refine those of higher loops.

4.4 AI Federation Framework

Existing fragmented and proprietary AI/ML approaches, calls for AI federation frameworks that envision sharing and collaborative deployment of AI workflows across multiple infrastructures. Currently, most infrastructures develop and deploy their own AI models, optimised for their specific network setups, often using proprietary datasets and local ML workflows. Additionally, heterogeneity in data formats, APIs, and orchestration frameworks makes it difficult to share or federate models. To address this, the development and sharing of AI/ML models and workflows across all federated facilities is promoted, enabling them to collaborate and share resources and AI models while continually improving their performance.

4.4.1 Challenges in distributed and federated networks

The AI/ML lifecycle in future networks is shaped by the distributed and federated nature of 6G infrastructures, where intelligence is expected to operate across edge, cloud, transport, and radio domains. This introduces a set of lifecycle-specific challenges:

- *Scalability and heterogeneity*: AI/ML lifecycle frameworks must operate across highly heterogeneous infrastructures, including cloud platforms, edge nodes, domain controllers, accelerators, and resource-constrained devices. The same model may need to exist in multiple forms, for example as a baseline version for centralized training and as a compressed or hardware-optimized version for inference at the edge. Lifecycle frameworks must therefore support portability, versioning, compatibility management, and adaptation to diverse execution environments.
- *Continuous data management*: AI/ML models depend on reliable and representative data throughout their lifecycle. In distributed networks, telemetry is collected from multiple sources with different formats, granularities, trust levels, and updating frequencies. A lifecycle framework must support data ingestion, normalization, abstraction, storage, access control, and integrity verification. It must also distinguish between data used for real-time inference and data retained for offline training, validation, and long-term optimization
- *Model deployment, adaptation, and retraining*: AI/ML models in operational networks are not static artifacts. Their performance may degrade over time due to traffic evolution, concept drift, topology changes, software updates, or shifts in user behaviour. Lifecycle frameworks

must therefore support mechanisms for model deployment, runtime monitoring, rollback, retraining, replacement, and retirement. This is especially relevant in multi-domain settings where different stakeholders may operate different parts of the lifecycle.

- *Decentralized and FL*: Bringing learning processes closer to the data source improves privacy and reduces data transport overhead, but it also complicates lifecycle management. Distributed learning introduces issues such as communication overhead, non-independent and non-identically distributed data, asynchronous updates, straggler effects, and inconsistent resource availability. A robust lifecycle framework must coordinate local and global learning loops, manage model aggregation and validation, and ensure that updates remain consistent with network-wide objectives.
- *Resource and energy constraints*: Lifecycle management must also consider resource efficiency. Training, retraining, and inference put varying demands on compute, memory, storage, and energy. In practical deployments, some lifecycle tasks may need to be shifted between cloud and edge depending on urgency, resource availability, and sustainability goals. Frameworks should therefore support intelligent placement of lifecycle functions and lightweight model adaptation strategies for constrained environments.

4.4.2 Architectural principles for AI-enabled federated networks

To address the challenges, AI/ML lifecycle in federated networks should be designed as modular, composable, and lifecycle-aware architectures. Rather than viewing AI as a standalone decision engine, these frameworks should organize the network intelligence stack around three tightly coupled capabilities: data management, model lifecycle management, and AI-driven execution and control. The architecture illustrated in Figure 7 can be effectively used for this purpose, reading it as a high-level AI/ML lifecycle framework for federated networks.

The **Data Management Block** provides the foundation for the AI/ML lifecycle by supporting the collection, curation, protection, and abstraction of network and service data. In lifecycle terms, this block is responsible for enabling the early phases of the pipeline, including data acquisition, preparation, storage, and controlled exposure to downstream model functions. The Network Data Repository stores telemetry and contextual information collected from the underlying infrastructure, while the Abstract Data Repository supports higher-level representations derived from raw observations. This distinction is important because lifecycle frameworks must support both low-latency operational data for inference and richer historical datasets for training and model refinement. The Data Security and Integrity Module ensures that the data entering the lifecycle is protected against tampering and unauthorized access, and that data consumers can rely on its trustworthiness. In a

lifecycle-oriented interpretation, this block underpins not only runtime intelligence but also dataset governance, data lineage, and reproducibility of training and validation workflows.

The **Models Management Block** is the central lifecycle enabler. It governs the creation, storage, optimization, deployment, update, and deletion of AI/ML models throughout their operational life. This block should be interpreted as the architectural anchor for the MLOps-like functions required in AI-native networks. The Model Repository stores multiple versions of models, including baseline, domain-specific, and hardware-optimized variants. This is essential in federated environments where one model may be trained centrally, adapted locally, and executed under different resource constraints. The Model Actions Module supports lifecycle operations such as onboarding, deployment, update, rollback, replacement, or withdrawal of models. The Model Security and Integrity Module ensures that only verified and trusted models are introduced into operation. From an AI/ML lifecycle perspective, this block should also be understood as supporting version control, provenance tracking, validation checkpoints, and coordinated rollout strategies. It enables the transition from experimental models to production-ready AI assets and supports their continuous evolution as network conditions change.

The **AI Controller Block** manages the lifecycle by executing models, enforcing decisions, and closing the loop between observation and action. It represents the runtime phase of the lifecycle, where models are consumed to produce decisions, policies, and control actions across distributed network domains. This block integrates the AI Control App, real-time decision logic, and mechanisms for communication between controllers. It is particularly important in federated settings where lifecycle execution is not centralized but distributed across multiple controllers located near the edge, in domains, or in higher orchestration layers. The N-S and E–W interfaces enable coordination between hierarchical and peer instances, thereby supporting the propagation of model updates, policy alignment, and collaborative learning or inference workflows. The Policy Supervisor and Conflict Mitigation Module is also essential from a lifecycle standpoint. As multiple AI models and controllers may coexist, the framework must ensure that updates or actions generated by one component do not create inconsistencies or conflicts elsewhere. This is particularly relevant when lifecycle events such as model replacement, retraining, or policy adaptation occur during live operation.

Taken together, the three architectural blocks define a coherent AI/ML lifecycle framework for federated networks.

- The **Data Management Block** supports data collection, preparation, abstraction, and protection.

- The **Models Management Block** supports model onboarding, storage, optimization, validation, deployment, update, and retirement.
- The **AI Controller Block** supports runtime execution, inference-driven control, inter-domain coordination, and closed-loop adaptation.

This separation of concerns is particularly valuable because it allows the lifecycle to be managed in a modular way, while still supporting tight integration across phases. It also enables different stakeholders to contribute to various lifecycle stages without losing E2E consistency. For example, one domain may own the telemetry, another may train or optimize the models, and another may execute them in real-time, all within a common framework

4.4.3 Federation in support of AI experimentation

When considering the specific case of AI experimentation across multiple research testbeds, the AI federation framework enables collaborative experimentation and lifecycle management of AI assets. In this case, the AI Federation framework leverages the advancements in MLOps, AI Operations (AIOps), and Foundation Models (FM) to facilitate the realization of incremental repositories of reusable AI/ML models and extracting synergistic knowledge from serving models and workflows to edge intelligence agents. It builds upon the principles of MLOps and AIOps to deliver a centralised MLOps stack that supports the development, sharing, and delivery of joint AI/ML models and experimentation workflows, spanning multiple federated testbeds that interface with the centralised (i.e., global) stack via their local MLOps deployments. The AI Federation framework also provides functionalities for transfer, federated and continual learning.

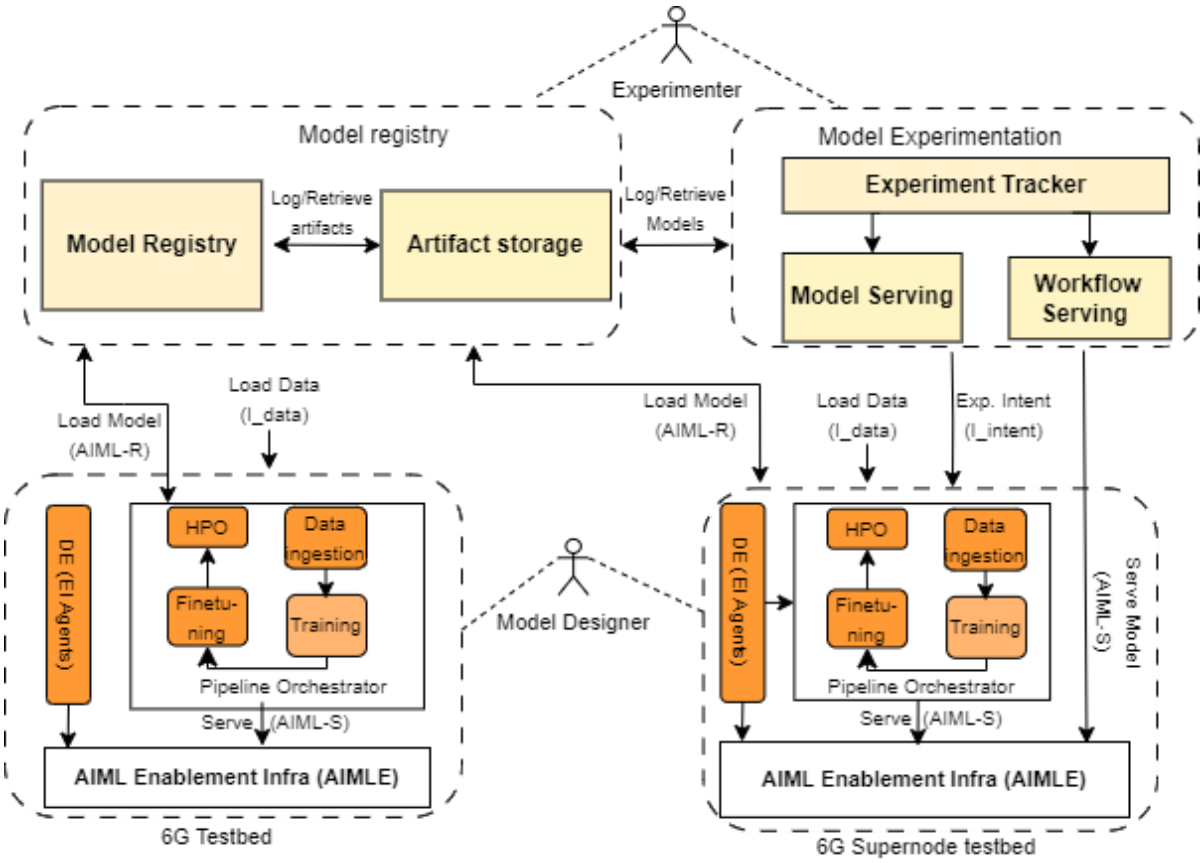


Figure 18: AI Federation Framework

The architecture of the AI federation framework illustrated in Figure 18 is designed to support experiment-driven AI/ML models and workflows through two main centralised components, which jointly constitute the so-called global MLOps stack:

- **Model Registry** module, which manages the metadata and lifecycle of ML models (e.g., foundation time-series models), including versioning, stage transitions, lineage, tagging and annotations. Model metadata includes model name, descriptions and annotations, model tags (key-value pairs), model creation time and model schema (input and output schema). The registry also provides a centralised catalogue for tracking registered models and their deployment status, as well as a centralised repository responsible for binary artefacts, datasets, and pipeline outputs with a backend storage for ML experiments, enabling reproducibility
- **Model Experimentation**, which allows for the registration of high-level workflow templates (dockerised components as building blocks/yaml/manifest files) and receives as output either the optimal model or ML pipeline. Here, only the workflow is transferred and exchanged, and the optimal model could be thus re-created by applying it with the local data from the testbed. The Model Experimentation module enables access to pre-trained models, and the

deployment of remote AI/ML experiments, leveraging computational resources offered by testbeds termed as “Supernodes”

Moreover, Local Pipeline Orchestration components are part of the local testbed’s MLOps stack and are involved in managing all steps and processes within the local MLOps workflow pipelines. Each involved testbed deploys its own pipeline orchestrator, such as Kubeflow, MLFlow, or Argo Workflows[23], and integrates its own Hyperparameter Optimization components, including Katib[24], Optuna[25], or AutoML[26] libraries, to optimise pipeline structures and hyperparameters. For every ML pipeline, the core steps include: "Load/Ingest Data" and "Training". The finetuning step is also critical for enabling transfer learning and FL across the AI federation plane, by adapting pre-trained models from the centralised ML Registry to local datasets or tasks.

4.4.3.1 AI Federation Paradigm for Telemetry Support

Beyond the federation capabilities dedicated to ML models operations and management, the AI Federation framework aims to support the automation and centralised storage of experimentation data collected through Telemetry APIs, as illustrated in Figure 19. This telemetry data is collected by Prometheus and Thanos sidecars from the K8s infrastructure, covering both hardware and software components. Thanos telemetry refers to the collection of operational metrics and performance data from the Thanos system, a highly available, long-term storage and querying layer for Prometheus. Thanos gathers information such as query latency, data ingestion rates, component health, and storage usage across the distributed architecture. This telemetry enables observability into Thanos's internal processes, helping operators monitor system behaviour, detect anomalies, and optimise resource usage for large-scale Prometheus deployments.

23 https://argoproj.github.io/workflows/

24 https://www.kubeflow.org/docs/components/katib/overview/

25 https://optuna.org/

26 https://www.automl.org/automl/

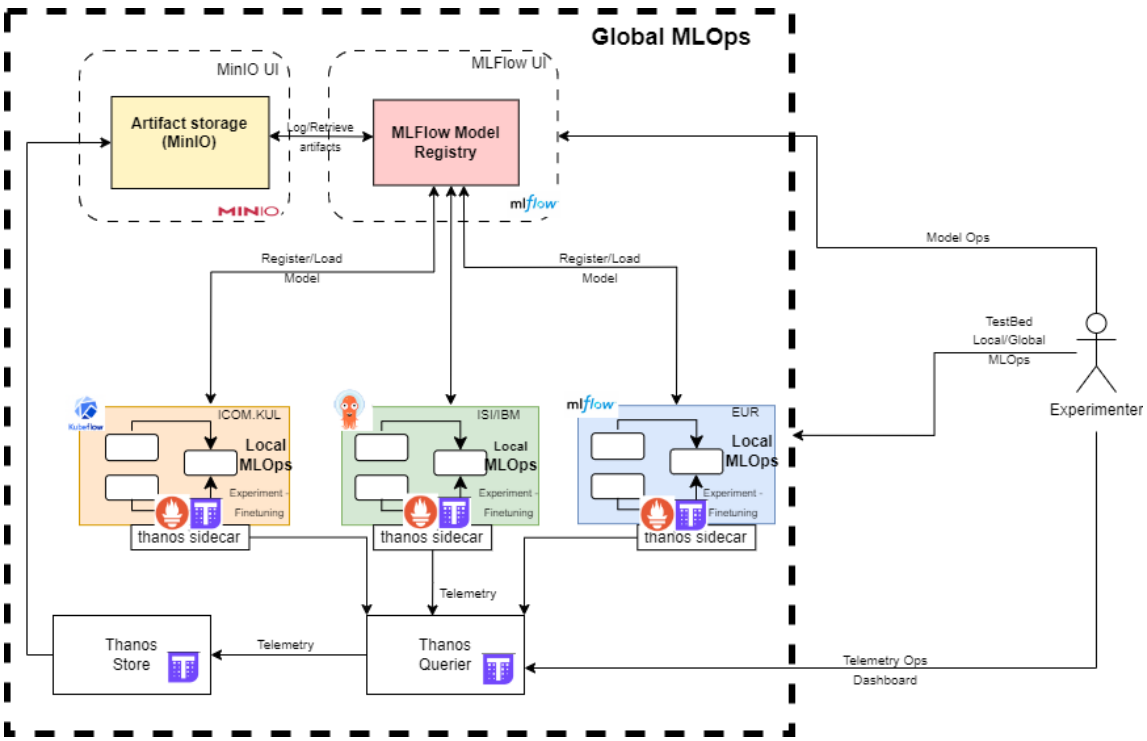


Figure 19: Telemetry Support in AI Federation Framework

Telemetry data are TS data consisting of logs, events, and metrics to monitor the health, performance, and behaviour of the cluster. The most common telemetries include CPU and memory requests/limits, nodes/pods status, network I/O, pod scheduling attempts/retries, etc. Monitoring CPU/memory requests help ensure that containers get the correct number of resources they need to run properly. On the other hand, monitoring CPU/memory limit helps avoid any single container consuming all the resources on a particular node, which may affect the stability of other containers running in the same node. In addition, monitoring CPU/memory usage helps in the decision whether to horizontally scale up or scale down. For network I/O, monitoring this metric helps to diagnose issues causing network latency and identify potential bottlenecks. By leveraging on Thanos to perform aggregated queries across Federated testbeds, the model designer can use PromQL, supported by the Thanos Querier, to extract a range of time-series data (for instance, CPU usage across testbeds) and use it as input to the MLOps workflow for regression or classification or anomaly detection using a certain workflow template. By having an optimal model that can predict CPU/memory usage for each node across testbeds, its predictions can be used to balance workload across testbeds.

4.4.3.2 Reusable AI/ML Assets and Foundation Models

The AI Federation framework supports federated management of AI/ML assets, via the development of incremental repositories of reusable models and the support for finetuning and domain adaptation. The relevance of time series analysis in the context of the AI Federation infrastructure lies in its ability to capture temporal dependencies and trends of both hardware and software services utilisation,

which are relevant for tasks covering forecasting (e.g. demand, workload, traffic, etc.) and anomaly detection (network latencies, CPU/memory usage, model drift, etc.). While many of the traditional methods for time series analysis such as AutoRegressive Integrated Moving Average (ARIMA) [46], moving average (MA), Support Vector Machines (SVMs) [47], Random Forests (RF)[48], Gradient Boosting Machines (GBMs) [49], Artificial Neural Networks (ANN) [50], Recurrent Neural Networks (RNN) [51], etc., have been successful in many previous application domains, they lack the ability to perform zero-shot or transfer learning, which are becoming a necessity in the context of cloud computing, where the volume and dimension of time series data to be processed are relatively high and dynamically changing. Traditional methods rely on individual time-series for training, which have limited capacity to scale up because of their inherent shortcomings of not being able to generalise on varying temporal patterns and resolutions, particularly with large volumes of heterogenous time series datasets [52]. One recent development in time series forecasting relies on the Transformer-based technology, with notable models such as PatchTST [53], Autoformer [54], and Fedformer [55] demonstrating significant improvements from the traditional statistical and ML techniques. In addition, MLP-mixer-based architectures, such as TSMixer [56] and TimeMixer [57], have shown competitive results with their transformer counterparts at notably reduced computational time.

Another recent development in time series applications is inspired by the remarkable success in the application of LLMs in Natural Language Processing (NLP) and Computer Vision (CV). One angle of research is to pre-train a FM from scratch using a large collection of time series data. Pioneering efforts have been initiated by TimeGPT [58] and TimesFM [59]. Another angle of research is anchored in the strong generalisation ability of LLMs in various downstream tasks, such as zero- or few-shot learning for prediction and anomaly detection, transfer learning, and explainability. Several massive pre-trained models for time series with strong benchmark results in zero-shot learning include Chronos [60], Moirai [61], Lag-llama [62], Moment [63], TimesFM [64], LLM-time [65], Time-LLM [66], and GPT4TS [67]. These large pre-trained TSFMs (Time Series Foundation Models) demand extremely high computing resources, paving the way for the development of relatively small language models that are cost-effective for real-world industrial deployments. Both angles of research are useful to the goal of having a unified architecture that consumes large datasets with powerful generalization, prediction, and explanation abilities.

5 AI/ML-ENABLED APPLICATIONS

5.1 AI/ML-enabled programmability

The evolution toward 6G networks introduces a paradigm shift in how vertical services are conceived, deployed, and optimized. This transformation goes beyond connectivity to enable intelligent, adaptive, and sustainable service ecosystems where the network itself becomes context-aware and AI-driven. At the core of this shift lies a novel architectural construct based on three distributed yet interlinked software components: the Network Application (N-App), the Vertical Application (V-App) and AI-Assisted Application (AI-App), which together enable the realization of intelligent, sustainable, and self-adaptive digital services. This triadic model represents a fundamental transformation from rather static network architectures to programmable, AI-driven ecosystems, where service delivery, network control, and cognitive intelligence operate in a closed feedback loop [68]. The triplet leverages open 5G/6G APIs and standardized interfaces - such as 3GPP Network Exposure Function (NEF) and Common API Framework (CAPIF) - to expose network capabilities securely and dynamically, allowing third-party developers and vertical industries to compose customized services.

The above-mentioned software components contribute to the transformation of a typical use case into a 6G Application, which entails the triplet of distributed but fully interacting software components that together realize an AI-assisted vertical service. Each 6G Application is composed of i) the V-App, ii) the N-App, and iii) the AI-App. The V-App serves as the domain-specific, user-oriented layer of the framework. It is tailored to the operational needs of vertical sectors. The N-App functions as the conduit between the V-App and AI-Apps , wrapping and exposing network capabilities through standardized 5G/6G interfaces such as NEF, NWDAF, and CAPIF. It leverages these open APIs to facilitate data exchange and control signalling, ensuring that the intents defined by the V-App and the insights produced by the AI-App can be applied dynamically within the network.

6G networks can be thus viewed as inherently efficient AI platforms, with network performance significantly enhanced by leveraging AI/ML capabilities. To fully harness the potential of this synergy, a Data Fabric Layer can be introduced, spanning across the far-edge cloud continuum and incorporating an ML-Ops framework, as the ones discussed above in this document. This layer provides processing capabilities and collects a new set of monitoring and analytics data across the network, administrative domains, and existing planes. Except for the data collected from various planes, the AI-app can contribute user-driven data to the Data Fabric Layer, enhancing context awareness, which is crucial for valuable processing chains, as well as facilitating information fusion for appropriate decision-making and orchestration at network elements and during the execution of the

6G pilots. The AI-App embodies the cognitive and analytical core of the framework. Using machine learning, data analytics, and context reasoning, it continuously processes data collected from both the V-App and N-App layers. In essence, it is the brain of the triad, transforming raw operational data into actionable knowledge. Furthermore, and within the Data Fabric Layer, the raw information received from various sources will be handled in a homogenized way, enabling seamless communication between all the components. The collected data undergo further analysis to extract the requirements of the V-App or the N-App, while the AI models support network operations, providing enhanced capabilities via predictions, detection, recognition, forecasting, recommendations, etc. The AI models will have, when necessary, the ability to interact with both the network and vertical use cases to perform actions and adaptations.

The related AI models are deployed within the network infrastructure using a flexible and unified architecture designed to support diverse use cases with varying performance, latency, and resource requirements. At the core of this deployment lies a highly adaptable and generic AI/ML framework capable of operating across different cluster scenarios while meeting the specific constraints and objectives of different use cases. To accommodate this variability, the deployment follows a hybrid central–edge model. The core AI/ML framework is hosted on a centralized server, ensuring consistency, scalability, and centralized management of data and models. In parallel, an AI-App is deployed on an edge server, where it interfaces directly with real-time data streams, local network resources, and end-user environments.

Two complementary architectural options are supported within this unified framework. In the first one, both model training and execution are handled by the centralized AI/ML framework. This approach avoids data duplication by maintaining all raw data, processed datasets, and model outputs in a single location, thereby improving data integrity and reducing storage requirements. It also lowers the computational load on edge systems, as resource-intensive tasks such as model training, dataset construction, and statistical report generation are performed exclusively on the central server, allowing the edge AI-App to remain lightweight.

In the second option, model training is performed centrally within the AI/ML framework, while model execution is deployed at the edge-based AI-App. This approach is particularly effective for scenarios requiring real-time or low-latency responses, as inference is executed close to the data source. By distributing computational responsibilities, the central framework can focus on data ingestion, storage, training, and reporting, while the edge handles time-critical model execution. This division enhances scalability, responsiveness, and efficient utilization of network resources.

5.2 AI/ML-enabled sensing

The AI/ML enabled sensing concept aims at providing an architecture to appropriately manage sensing information coming from different sources and technologies. While Integrated Sensing and Communications (ISAC) is traditionally considered to be used on a single technology, and more traditional sensing relies on different devices (cameras, LiDAR, etc), a framework that goes beyond this constrained vision is introduced in [69]. Within such architecture the sensing data are processed, and the resulting information, which already combines multiple sources, is made available to different network entities and external stakeholders. This secure and structured access is facilitated by a Data Access and Security Hub (DASH).

A key strategic objective of such framework is to contribute to energy-efficient AI architectures for adaptive, event-based ISAC receivers and distributed learning from multi-static sensing devices. The goal is to achieve a reduction of AI-related computing energy consumption by, at least, 20%. Towards this direction four main research lines are of importance [70]:

- Energy-efficient, neuromorphic, and event-based ISAC receivers. This specifically deals with the energy efficiency of the AI architectures themselves. New architectures for ISAC signal processing, both event-based and neuromorphic, will be designed and validated.
- Resource optimization with AI. This leverages AI to optimize the energy efficiency of ISAC-enabled RAN. One specific example is the optimal management, exploiting AI techniques, of RIS-based systems.
- Multi-modal sensing. refers to the ability to combine different sensing sources. To achieve this, multi-modal sensing devices are needed, where AI techniques can be exploited to optimally combine the sensing information coming from various sources and technologies.
- AI/GenAI for multi-band and multi-static ISAC. This explores the use of AI and GenAI to enable or improve multi-band and multi-static ISAC, tackling the problem of incomplete channel observations in frequency and space that arises when considering multi-band and multi-static ISAC nodes.

Besides these points, and with a system-wide perspective, the leverage of AI techniques, solutions and algorithms to: (1) appropriately manage the sensing information and (2) exploit it to yield benefits to both communication mechanisms and services.

Advanced techniques to address data fusion, exploiting Dynamic DNN requires the potential integration of the Quantile-Constrained Inference and Configuration (QUIC) framework for the multi-modal sensor fusion inference task.

One of the interesting use cases is how the sensing information can be exploited to yield improvements in the performance of different communication mechanisms. In this sense, ML techniques have been already used to yield a more efficient use of radio resources. One promising method involves using a DRL-based agent that, considering sensing information, optimally configures a MAC scheduler. Initial results evince that the use of such sensing information can bring substantial benefits, in terms of resource efficiency.

Furthermore, the sensing information can be leveraged to provide benefits to certain services/applications. In this sense, there are some initial results on how the use of AI/ML techniques can be effectively used to combine (data fusion) sensing information originated from different technologies, both 3GPP and non-3GPP (e.g., WiFi).

5.3 AI/ML-enabled Network Digital Twins

The convergence of AI and NDTs is reshaping the landscape of network management, optimization, and innovation. This section explores the bidirectional relationship between AI and NDTs: how NDTs serve as a foundation for AI-driven decision-making and dataset generation and how AI, in turn, enables the creation, calibration and operation of sophisticated digital twin models. We delve into real-world applications, architectural considerations, and future research directions that highlight the transformative potential of this synergy.

5.3.1 Network Digital Twins: Foundations and Capabilities

NDTs are a dynamic, virtual representations of a physical network, including its topology, traffic flows, protocols, and performance metrics. They typically consists of:

- Data Ingestion Layer: Collects real-time telemetry and historical data.
- Modeling Engine: Simulates network behaviour under various conditions.
- Analytics and Visualization: Provides insights through dashboards and predictive analytics.
- Control Interface: Enables bidirectional communication with the physical network.

NDTs offer a powerful solution by enabling real-time simulation, analysis, and prediction. When integrated with AI, they become intelligent agents capable of supporting autonomous decision-making and continuous learning.

Considering the above, we provide below some illustrative examples of such decision making and learning scenarios. For each, the corresponding use cases are identified, as well as the benefits brought by the combination of AI and NDT.

5.3.2 NDTs as Enablers of AI

AI models, especially those based on deep learning, require vast amounts of labelled data. NDTs can generate high-fidelity synthetic datasets that reflect diverse network conditions, including rare edge cases that are difficult to capture in real-world data.

- Use Case: Training intrusion detection systems with simulated attack scenarios.
- Benefits: The NDT reduces the dependency on costly, time-consuming data collection. It allows the generation of very rare or unpredictable events.

In addition, NDTs provide a controlled, risk-free environment for training RL agents or testing AI-driven network policies, such as training an AI agent to dynamically optimize routing protocols under varying traffic loads, from routine operations and predictable peaks to sudden congestion spikes, link failures, or slice-specific overloads. ensuring zero disruption to the performance of the network, while enabling rapid iteration, robust policy validation, and safe experimentation in highly realistic simulated conditions

When it comes to decision support systems, AI-enabled NDTs allow operators to receive intelligent, data-backed recommendations for network configuration, fault resolution, and performance tuning across complex, multi-slice environments, such as AI-driven root cause analysis that leverages high-fidelity NDT simulations to systematically test and validate hypotheses (e.g., pinpointing slice interference, backhaul bottlenecks during mobility handovers, or configuration errors) before applying changes to production networks, thereby minimizing downtime and optimizing resource allocation.

5.3.3 AI for Building and Operating NDTs

Creating accurate digital twins requires detailed modelling of network behaviour. AI automates this process by learning patterns from historical data (e.g., performance logs, simulation traces) and inferring underlying network dynamics, employing techniques such as Graph Neural Networks (GNNs) for topology-aware spatial predictions, time-series forecasting and unsupervised clustering. As a result, this AI-driven approach enables significantly faster and more accurate digital twin deployment.

AI algorithms can continuously align the digital twin with the physical network by detecting and correcting model drift arising from evolving conditions such as traffic shifts, hardware faults, or configuration changes. This is achieved through online learning for real-time parameter updates from streaming data, complemented by adaptive filtering techniques to ensure precise synchronization. The key challenge lies in balancing high model fidelity—critical for accurate predictions with computational efficiency preventing excessive load on resources.

AI can also manage the full lifecycle of NDTs, from dynamic resource allocation and intelligent update scheduling to scenario prioritization, ensuring optimal utilization across cloud-edge continuum in evolving 6G environments. For instance, AI algorithms can proactively prioritize simulations targeting predicted network congestion hotspots, detected via real-time traffic forecasting and anomaly analysis.

5.4 AI/ML-enabled data plane inference

With the current push towards network automation, many control-plane network management functions are expected to transition from human-driven to data-driven processes. Specifically, ML models trained on large amounts of data collected from the network are seen as a prime candidate to implement the intelligence that will enable full network automation: indeed, Standard-Defining Organizations (SDO) are already working on a native integration of MLOps in MANO frameworks [71]. However, the fact that ML operation is bounded to the control plane in Software Defined Networks (SDN) and upcoming standards creates a barrier for latency reduction, since it forces a closed-loop communication between the user and control plane with an inherent delay in the order of tens of milliseconds. This prevents ML models from making very fast decisions and necessarily limits the frequency of their intervention in low-latency application scenarios.

In this context, next-generation programmable network hardware that will characterize next-generation CNs has paved the road for the deployment of ML models in the user plane. Offline trained Decision Trees (DTs), RFs or Neural Networks (NNs) can be directly implemented into programmable switches or smartNICs, where they perform complex inference tasks on every transiting packet at line-rate, i.e., at forwarding capacity (e.g., on hundreds of Gbps of traffic) and with ultra-low latency (e.g., in the order of tens of nanoseconds). User-plane ML removes the need for interactions with the control plane where classifiers have traditionally operated, with multiple advantages that include reduced delays (cut by orders of magnitude), higher scalability (as overheads, e.g., for traffic mirroring, are eliminated) and fewer costs (since expensive dedicated hardware, e.g., for Deep Packet Inspection, becomes unnecessary).

However, embedding ML models into programmable network hardware is a hard task, as it requires tailoring the design and operation of the original models to highly constrained targets. Specifically, programmable switches and smartNICs have severe limitations in terms of computing capabilities and memory resources, in addition to internal architectures that are optimized for packet forwarding processes and not inference task.

While solutions have been proposed in the past few years to adapt ML models to user plane environments, by performing classification in switches and smartNICs, using DTs RFs, or NNs, and at packet-level, flow-level or at both levels jointly, they have invariably sought to implement monolithic ML models into a single switch or smartNIC. Such a strategy unnecessarily curbs the potential of ML-driven inference in network user planes that are inherently composed of tens or hundreds of switches and middleboxes.

A much more flexible strategy can be achieved by distributing intelligence across multiple network elements, enabling inference tasks to be performed on packets and flows as they traverse the network rather than at a single specific point. Figure 20 illustrates this concept, for the case where the target network is composed of programmable switches only: unlike monolithic approaches proposed in the literature where the whole inference problem is solved by a single model deployed in one switch, each programmable switch performs only a subset of the classification problem, tagging (and forwarding accordingly) the portion of traffic it can classify, and transmitting to a downstream switch the other traffic it cannot identify as belonging to one of the classes it is responsible for. Hence, all switches within the network collaboratively classify the whole traffic.

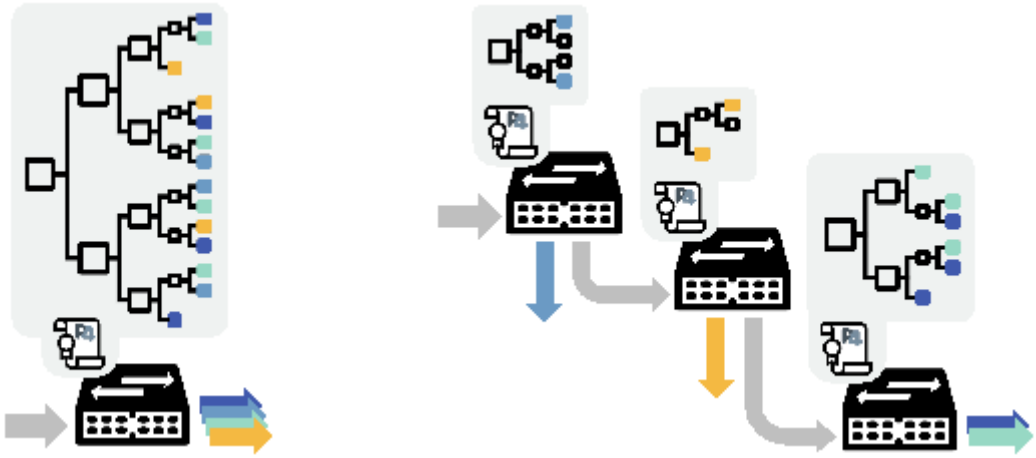


Figure 20: Conceptual illustration of user-plane inference strategies based on (left) a traditional monolithic approach and (right) a distributed paradigm.

A comprehensive pipeline for distributed ML models has been developed to enable deployment across different programmable switches, achieving performance comparable to or better than that of large ML models executed in the control plane or monolithic ML models implemented on a single programmable switch. The solution Distributed User plaNe inference (DUNE) supports distributed inference across network elements.

DUNE is capable of (i) decomposing the original ML task into a set of ML sub-problems that can jointly solve the original task, (ii) designing models that can solve such ML sub-problems and are compatible

with programmable network hardware constraints, (iii) ordering the models to achieve the best inference performance, (iv) deploying copies of the models in complex network topologies to minimize resource utilization at the user-plane. All these functions are fully automated and allow increasing the accuracy of the inference by up to 7.5%, while reducing per-switch resource consumption, in challenging traffic classification tasks with tens of target categories as demonstrated by tests in experimental platforms with production-grade programmable network equipment [72].

CONCLUSIONS

AI/ML frameworks are essential for the evolution of smart network infrastructures, enabling autonomous, adaptive, and secure 6G services. Continued innovation in orchestration, lifecycle management, security, and applications will drive the realization of AI-native networks. As presented in this white paper, AI/ML frameworks from SNS JU projects enable intent-driven, zero-touch, and agentic orchestration across multi-domain and federated environments. As new security and trust challenges emerge (including data poisoning, adversarial attacks, and model extraction) a lot of attention is given to automated security management, trust-aware mechanisms, isolated execution, privacy-preserving techniques, and AI-enhanced physical-layer security. At the application layer, AI/ML-enabled applications drive innovation in programmability, sensing, NDTs, and data plane inference for 6G networks. These solutions enhance context-awareness, real-time analytics, safe experimentation, and distributed intelligence for efficient and adaptive service delivery.

Overall, MLOps principles are widely adopted to automate the lifecycle of AI/ML models in those frameworks, ensuring reproducibility, traceability, and continuous integration across distributed infrastructures. There is also a wide range of open-source AI/ML tools for Model definition, training, and orchestration, which has not only accelerated research and experimentation processes, but also affected the way that smart networks and services are designed.

REFERENCES

[1]. ETSI White Paper, AI in the evolution of Autonomous Networks ETSI perspectives and major achievements First edition – Nov. 2025. online: https://www.etsi.org/images/files/ETSIWhitePapers/ETSI-WP-69-AI-in_the_evolution_of_Autonomous_Networks.pdf.

[2]. Tsolkas, D., Paul, S., Tranoris, C., Skarmeta Gómez, A., Vilalta, R., Agüero, R., Papagianni, C., Hsu, C. S.-H., Trichias, K., & Gavras, A. (2026). The AI/ML landscape for Smart Networks and Services. Zenodo. https://doi.org/10.5281/zenodo.17708460.

[3]. M. Gramaglia et al., "Network Intelligence for Virtualized RAN Orchestration: The DAEMON Approach," 2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), Grenoble, France, 2022, pp. 482-487, doi: 10.1109/EuCNC/6GSummit54941.2022.9815816.

[4]. A. Boutouchent et al., "6G-INTENSE: Intent-Driven Native Artificial Intelligence Architecture Supporting Network-Compute Abstraction and Sensing at the Deep Edge," in IEEE Vehicular Technology Magazine, vol. 20, no. 1, pp. 44-54, March 2025, doi: 10.1109/MVT.2024.3525001.

[5]. Vaca-Rubio, C. J., Kasuluru, V., Zeydan, E., Blanco, L., Pereira, R., Caus, M., & Dev, K. (2025). Probabilistic Forecasting for Network Resource Analysis in Integrated Terrestrial and Non-Terrestrial Networks. IEEE Communications Standards Magazine.

[6]. Kasuluru, V., Blanco, L., & Zeydan, E. (2025). Enhancing Open RAN Operations: The Role of Probabilistic Forecasting in Network Analysis. IEEE Transactions on Network and Service Management.

[7]. Vaca-Rubio, C. J., Pereira, R., Blanco, L., Zeydan, E., & Caus, M. (2025). A Primer on Kolmogorov-Arnold Networks (KANs) for Probabilistic Time Series Forecasting. arXiv preprint arXiv:2510.16940.

[8]. Keramidi, I., Zervou, P., Lakoumentas, J., Ramantas, K., & Verikoukis, C. (2024, October). Towards the uncertainty quantification of AI-models in 6G networks. In 2024 IEEE 29th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD) (pp. 1-6). IEEE.

[9]. Liu, H. I., Galindo, M., Xie, H., Wong, L. K., Shuai, H. H., Li, Y. H., & Cheng, W. H. (2024). Lightweight deep learning for resource-constrained environments: A survey. ACM Computing Surveys, 56(10), 1-42.

[10]. Mittal, P. (2024). A comprehensive survey of deep learning-based lightweight object detection models for edge devices. Artificial Intelligence Review, 57(9), 242.

[11]. Hoffpauir, K., Simmons, J., Schmidt, N., Pittala, R., Briggs, I., Makani, S., & Jararweh, Y. (2023). A survey on edge intelligence and lightweight machine learning support for future applications and services. ACM Journal of Data and Information Quality, 15(2), 1-30.

[12]. Liu, L., & Xu, Z. (2025). Optimizing lightweight neural networks for efficient mobile edge computing. Scientific Reports, 15(1), 22056.

[13]. Das, B. R., Hasan, S. R., Sabuj, S. R., Hossain, M. A., & Ray, S. K. (2025). A Comprehensive Survey on Emerging AI Technologies for 6G Communications: Research Direction, Trends, Challenges and Opportunities. International Journal of Intelligent Networks.

[14]. TM Forum IG1230 (V1.1.1): "Autonomous Networks Technical Architecture"

[15]. ETSI GR ZSM 020 V1.1.1 (2026-01) “Zero-touch network and Service Management (ZSM);Study on the Utilization of Agents in Autonomous Networks [online] available at: https://www.etsi.org/deliver/etsi_gr/ZSM/001_099/020/01.01.01_60/gr_ZSM020v010101p.pdf

[16]. J. White et al., "Building Living Software Systems with Generative & Agentic AI," arXiv preprint arXiv:2408.01768, 2024

[17]. R. Zhang et al., S. Tang, Y. Liu, D. Niyato, Z. Xiong, S. Sun, et al., "Toward Agentic AI: Generative Information Retrieval Inspired Intelligent Communications and Networking," arXiv preprint, arXiv:2502.16866, 2025.

[18]. S. A. Khowaja et al., "Integration of Agentic AI with 6G Networks for Mission-Critical Applications: Use-case and Challenges," arXiv preprint, arXiv:2502.13476, 2025.

[19]. A. S. Araujo et al., "An agentic approach for dynamic software-defined network management using large language models," 2024 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2024

[20]. B. Gort, G. M. Kibalya, and A. Antonopoulos, “AgentEdge: Agentic AI for service orchestration in the edge-cloud continuum,” TechRxiv, Aug. 12, 2025, doi: 10.36227/techrxiv.175503000.00051702/v1.

[21]. R. Singh and S. S. Gill, “Edge AI: a survey,” Internet Things Cyber-Phys. Syst., vol. 3, pp. 71–92, 2023.

[22]. R. Boutaba et al., “A comprehensive survey on machine learning for networking: evolution, applications and research opportunities,” J. Internet Serv. Appl., vol. 9, no. 1, pp. 1–99, 2018.

[23]. E. Coronado et al., “Zero touch management: A survey of network automation solutions for 5G and 6G networks,” IEEE Commun. Surv. Tutor., vol. 24, no. 4, pp. 2535–2578, 2022.

[24]. T. Z. Benmerar et al., "Intelligent Multi-Domain Edge Orchestration for Highly Distributed Immersive Services: An Immersive Virtual Touring Use Case," 2023 IEEE International Conference on Edge Computing and Communications (EDGE), Chicago, IL, USA, 2023, pp. 381-392, doi: 10.1109/EDGE60047.2023.00061.

[25]. C. Benzaïd, T. Taleb, A. Sami and O. Hireche, "A Deep Transfer Learning-Powered EDoS Detection Mechanism for 5G and Beyond Network Slicing," GLOBECOM 2023 - 2023 IEEE Global

Communications Conference, Kuala Lumpur, Malaysia, 2023, pp. 4747-4753, doi: 10.1109/GLOBECOM54140.2023.10436891.

[26]. T. Taleb, A. Boudi, L. Rosa, L. Cordeiro, T. Theodoropoulos, K. Tsepes, P. Dazzi, A. Protopsaltis, and R. Li, “Toward Supporting XR Services: Architecture and Enablers,” in IEEE IoT Journal, Vol. 10, No. 4, Feb. 2023, pp. 3567 – 3586

[27]. Theodoros Theodoropoulos, Luis Rosa, Abderrahmane Boudi, Tarik Zakaria Benmerar, Antonios Makris, Tarik Taleb, Luis Cordeiro, Konstantinos Tserpes and JaeSeung Song, “Cross-Cluster Networking to Support Extended Reality Services,” in IEEE Network Magazine, Vol. 39, No. 2, Mar. 2025, pp. 250-258.

[28]. Tarik Zakaria Benmerar, Theodoros Theodoropoulos, Diogo Fevereiro, Luis Rosa, Joao Rodrigues, Tarik Taleb, Paolo Barone, Giovanni Giuliani, Konstantinos Tserpes, and Luis Cordeiro, “Towards Establishing Intelligent Multi-Domain Edge Orchestration for Highly Distributed Immersive Services: A Virtual Touring Use Case,” in Springer Cluster Computing – J. of Networks Software Tools and Applications, Vol. 27, Jul. 2024, pp. 4223–4253.

[29]. Dias, T., Ferreira, L., Fevereiro, D., Rosa, L., Cordeiro, L., Fernandes, J. (2025). Cloud-Native Scheduling and Resource Orchestration: A Deep Dive into AI-Driven Approaches. In: Artificial Intelligence Applications and Innovations. AIAI 2025 IFIP WG 12.5 International Workshops. AIAI 2025. IFIP Advances in Information and Communication Technology, vol 753. Springer, Cham. https://doi.org/10.1007/978-3-031-97317-8_8

[30]. Dias, T., Velasquez, K., Ferreira, L., Fevereiro, D., Rosa, L., Cordeiro, L. (2025). Resource-and-Latency-Aware PSO — A novel algorithm for intelligent and contextual cloud native resource scheduling. To be published in 2026 IEEE Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 2026

[31]. Masoud Shokrnezhad and Tarik Taleb, “An Autonomous Network Orchestration Framework Integrating Large Language Models with Continual Reinforcement Learning,” in IEEE Communications Magazine, Vol. 63, No. 8, Aug. 2025, pp. 78 – 84

[32]. M. Shokrnezhad, T. Taleb, and P. Dazzi “Double Deep Q-Learning-Based Path Selection and Service Placement for Latency-Sensitive Beyond 5G Applications," in IEEE Transactions on Mobile Computing, Vol. 23, No. 5, May 2024, pp. 5097 - 5110.

[33]. C. Benzaid, T. Taleb, A. Sami, and O. Hireche, “FortisEDoS: A Deep Transfer Learning-empowered Economical Denial of Sustainability Detection Framework for Cloud-Native Network Slicing,” in IEEE Transactions on Dependable and Secure Computing, Vol. 21, No. 4, Jul. 2024, pp. 2818 – 2835

[34]. H. Yu, T. Taleb, and J. Zhang, “Deep Reinforcement Learning based Deterministic Routing and Scheduling for Mixed-Criticality Flows,” in IEEE Transactions on Industrial Informatics, Vol. 19, No. 8, Aug. 2023, pp. 8806-8816.

[35]. Y. Arjoune, F. Salahdine, M. S. Islam, E. Ghribi and N. Kaabouch, "A Novel Jamming Attacks Detection Approach Based on Machine Learning for Wireless Communication," 2020 International Conference on Information Networking (ICOIN), Barcelona, Spain, 2020, pp. 459-464, doi: 10.1109/ICOIN48656.2020.9016462.

[36]. Papadopoulos, Alexandros I., et al. "On Jamming Detection: A Unified Method for Real-Time Detection Across Multiple Protocols & Attack Types." 2025 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit). IEEE, 2025.

[37]. J. Vardakas, G. Famitafreshi, C. Christoforou, C. F. Chiasserini, A. Molinaro, C. Ben Issaid, J. J. Vegas Olmos, T. Charemis, C. Guimarães, M. Iordache, M. Sofocleous, G. Bernini, C. Verikoukis, "Enabling a Robust and Scalable 6G Network Architecture through Distributed and Crowdsourced Artificial Intelligence", IEEE Vehicular Technology Magazine, special issue on 6G Technologies and Applications, doi: 10.1109/MVT.2026.3684194.

[38]. Zhibo Wang, Jingjing Ma, Xue Wang, Jiahui Hu, Zhan Qin, and Kui Ren. 2022. Threats to Training: A Survey of Poisoning Attacks and Defenses on Machine Learning Systems. ACM Comput. Surv. 55, 7, Article 134 (July 2023), 36 pages. https://doi.org/10.1145/3538707.

[39]. Policy Verification Reference: S. Paul, S. B. M. Baskaran, A. Kunz and A. Salkintzis, "Evolution of Policy Verification and Control for 6G Secure Service Orchestration in Telco Clouds," 2025 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), Poznan, Poland, 2025, pp. 363-368, doi: 10.1109/EuCNC/6GSummit63408.2025.11036971.

[40]. Andreas Haas, Andreas Rossberg, Derek Schuff, Ben Titzer, Dan Gohman, Luke Wagner, Alon Zakai, JF Bastien, and Michael Holman. 2017. Bringing the Web up to Speed with WebAssembly. In Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2017). DOI: 10.1145/3062341.3062363

[41]. AI/ML workflow description and requirements-V01.01, ORAN Alliance, Alfter, Germany, 2020

[42]. O-RAN AI/ML workflow description and requirements 1.03, O-RAN.WG2.AIML-v01.03, October 2021.

[43]. Polese, M., Bonati, L., D'oro, S., Basagni, S., & Melodia, T. (2023). Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges. IEEE Communications Surveys & Tutorials, 25(2), 1376-1411.

[44]. A. E. Giannopoulos et al., "Supporting intelligence in disaggregated open radio access networks: Architectural principles, AI/ML Workflow, and Use Cases," vol. 10, pp. 39580–39595, Jan. 2022, doi: https://doi.org/10.1109/access.2022.3166160.

[45]. S. D'Oro, M. Polese, L. Bonati, H. Cheng, and T. Melodia, "dapps: Distributed applications for real-time inference and control in o-ran," IEEE Communications Magazine, vol. 60, no. 11, pp. 52–58, 2022.

[46]. G. E. Box, G. M. Jenkins and J. F. MacGregor, "*Some recent advances in forecasting and control*," Journal of the Royal Statistical Society: Series C (Applied Statistics), vol. 23, no. 2, pp. 158--179, 1974

[47]. M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support Vector Machines," IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28, 1998

[48]. L. Breiman, "Random Forests," Machine Learning, no. 1, pp. 5-32, 2001.

[49]. J. H. Friedman, "Greedy function approximation: A gradient boosting machine," The Annals of Statistics, vol. 29, no. 5, p. 189–1232, 2001.

[50]. P. J. Werbos, "Backpropagation through time: what it does and how to do it," Proceedings of the IEEE, 1990, pp. 1550 - 1560.

[51]. J. T. Connor, R. D. Martin and L. E. Atlas, "Recurrent Neural Networks and Robust Time Series Prediction," IEEE Transactions on Neural Networks, vol. 5, no. 2, p. 240–254, 1994.

[52]. S. Palaskar, V. Ekambaram, A. Jati, N. Gantayat, A. Saha, S. Nagar, N. H. Nguyen, P. Dayama, R. Sindhgatta, P. Mohapatra and others, "Foundation Models for Time Series: A Survey," 2025. Available: https://arxiv.org/abs/2504.04011

[53]. Y. Nie, N. H. Nguyen, P. Sinthong and J. Kalagnanam, "A Time Series is Worth 64 Words: Long-term Forecasting with Transformers," in ICLR, 2023.

[54]. M. Chen, H. Peng, J. Fu and H. Ling, "Autoformer: Searching transformers for visual recognition," in Proceedings of the IEEE/CVF international conference on computer vision, 2021.

[55]. T. Zhou, Z. Ma, Q. Wen and X. W. et.al., "FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting," in Proc. 39th International Conference on Machine Learning (ICML 2022), Baltimore, Maryland, 2022.

[56]. V. Ekambaram, A. Jati, N. Nguyen, P. Sinthong and J. Kalagnanam, "TSMixer: Lightweight MLP-Mixer Model for Multivariate Time Series Forecasting," in Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, 2023, pp. 459--469.

[57]. S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang and J. ZHOU, "TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting," in International Conference on Learning Representations (ICLR), 2024.

[58]. Garza and M. Mergenthaler-Canseco, "TimeGPT-1," arXiv preprint arXiv:2310.03589, 2023.

[59]. Das, W. Kong, R. Sen and Y. Zhou, “A Decoder-Only Foundation Model for Time-Series Forecasting,” arXiv:2310.10688, 2023

[60]. F. A. Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. P. Arango, S. Kapoor and others, “Chronos: Learning the Language of Time Series,” arXiv preprint arXiv:2403.07815, 2024

[61]. G. Woo, C. Liu, A. Kumar, C. Xiong and others, “Unified training of universal time series forecasting transformers,” arXiv preprint arXiv:2402.02592, 2024

[62]. K. Rasul, A. Ashok, A. R. Williams, A. Khorasani, G. Adamopoulos, R. Bhagwatkar et al, “Lag-llama: Towards foundation models for time series forecasting,” arXiv preprint arXiv:2310.08278, 2023

[63]. M. Goswami, K. Szafer, A. Choudhry, Y. Cai, S. Li and A. Dubrawski, “Moment: A family of open time-series foundation models,” arXiv preprint arXiv:2402.03885, 2024

[64]. Google TimesFM: https://cloud.google.com/bigquery/docs/timesfm-model

[65]. N. G. Wilson, M. A. Finzi, S. Qiu and A. Gordon, “Large Language Models Are Zero-Shot Time Series Forecasters,” in Thirty-seventh Conference on Neural Information Processing Systems, 2023

[66]. M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan and Q. Wen, “Time-LLM: Time Series Forecasting by Reprogramming Large Language Models,” in Thirty-seventh Conference on Neural Information Processing, 2023

[67]. T. Z. Jin, P. Niu, X. Wang, L. Sun and Rong, “One Fits All:Power General Time Series Analysis by Pretrained LM,” ArXiv: 2302.11939, 2023.

[68]. 3GPP, “Study on concept, requirements and solutions for AI/ML-based network management and orchestration,” TR 28.810, v. 19.1.0, Mar. 2024.

[69]. D. Raddino et al. D2.1 MultiX Perceptive 6G-RAN system design v1 – focus on initial DASH Design. MultiX project.

[70]. J. Pegoraro et al. D3.1 MultiX perception system enablers v1: Focus on perception enablers design including AI and multi-technology, multi-static perception. MultiX project.

[71]. European Telecommunications Standards Institute (ETSI), “Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Report on enabling autonomous management in NFV-MANO,” ETSI GR NFV-IFA 041, 2021.

[72]. B. Bütün, D. de Andrés Hernández, M. Gucciardo, M. Fiore, DUNE: Distributed Inference in the User Plane, IEEE INFOCOM 2025, London, UK, May 2025.

ABBREVIATIONS AND ACRONYMS

Acronym	Description
3GPP	3rd Generation Partnership Project
6G	Sixth Generation Networks
AI	Artificial Intelligence
AI-App	AI-Assisted Application
AIaaS	AI-as-a-Service
AIOps	AI Operations
AMC	Adaptive Modulation and Coding
ANN	Artificial Neural Networks
AN	Autonomous Networks
API	Application Programming Interface
ARIMA	AutoRegressive Integrated Moving Average
BDIx	Belief-Desire-Intent Extended
CAPIF	Common API Framework
CI/CD	Continuous Integration and Continuous Delivery
CNs	Core Networks
CV	Computer Vision
DASH	Data Access and Security Hub
DIMO	Intent-driven Management and Orchestration
DMO	Domain Manager and Orchestrator
DN	Decision Trees
DP	Differential Privacy
DS	Data Spaces
DUNE	Distributed User plaNe inference
E2E	End-to-End
E2E SO	End-to-End Security Orchestrator
ETSI	European Telecommunications Standards Institute
E-W	East-West
FL	Federated Learning
FM	Foundation Models
GBMS	Gradient Boosting Machines

GenAI	Generative AI
GNNs	Graph Neural Networks
HE	Homomorphic Encryption
HLO	High-Level Orchestrator
IaaS	Infrastructure-as-a-Service
ICT	Information and Communication Technologies
IDAN	Intent-Driven Autonomous Networks
IDS	International Data Spaces
IF	Isolation Forest
IMFs	Intent Management Functions
ISAC	Integrated Sensing and Communications
JASMIN	JAmming detection method baSed on Modulation scheme IdeNtification
K8s	Kubernetes
LCM	Life Cycle Manager
LLMs	Large Language Models
LLO	Low-Level Orchestrator
LMO	Local Manager and Orchestrator
LOF	Local Outlier Factor
MA	Moving Average
MANO	Management and Orchestration
MIMO	Massive Multiple-input and multiple-output
MILP	Mixed Integer Linear Programming
MIRO	Multi-Domain Intelligent Resource Orchestration
ML	Machine Learning
MLaaS	Machine Learning as a Service
MLOps	Machine Learning Operations
MSI	Modulation Scheme Identification
MSPL	Medium-level Security Policy Language
N-App	Network Application
NBI	Northbound Interface
NCF	Network Compute Fabric
NDTs	Network Digital Twins
Near-RT RIC	Near-Real-Time RIC

NEF	Network Exposure Function
Network-as-a Service	Network-as-a-Service
NFV	Network Function Virtualization
NFVO	Network Functions Virtualization Orchestrator
NNs	Neural Networks
NoN	Network of Networks
Non-RT RIC	Non-Real-Time RAN Intelligent Controller
N-S	North-South
OCSVM	One Class Support Vector Machine
OD	Outlier Detector
O-RAN	Open Radio Access Network
PARES	Perceive, Act, Reason, Evaluate and Sustain
PM	Performance Monitoring
PromQL	Prometheus Query Language
PVF	Policy Verification Function
QIC	Quantile-Constrained Inference and Configuration
QoE	Quality of Experience
QoS	Quality of Service
RAG	Retrieval-Augmented Generation
RAN	Radio Access Network
REST	Representational State Transfer
RF	Random Forests
RIC	RAN Intelligent Controller
RL	Reinforcement Learning
RLOps	Reinforcement Learning Operations
RNN	Recurrent Neural Networks
SBI	Southbound Interface
SDN	Software Defined Networks
SDO	Standard-Defining Organizations
SL	Supervised Learning
SLA	Service Level Agreement
SMO	Service Management Orchestrator
SMPC	Secure Multi-Party Computation

SO	Security Orchestrator
SVMs	Support Vector Machines
t-DMO	Tenant Domain Manager and Orchestrator
TEEs	Trusted Execution Environments
TM Forum	Tele Management Forum
TTI	Transmission Time Interval
UE	User Equipment
UL	Unsupervised Learning
URLLC	Ultra-Reliable Low-Latency Communications
V-App	Vertical Application
VBS	Virtual Base Station
VMs	Virtual Machines
VNF	Virtual Network Function
ZSM	Zero-touch network and Service Management

LIST OF EDITORS & REVIEWERS

Name	Company / Institute / University	Country
Document editors		
Andreas Gavrielides	University of Antwerp/ IMEC	BE
Antonio Skármeta	University of Murcia	ES
Dimitris Tsolkas	Fogus Innovations & Services P.C.	GR
George Makropoulos	NCSR Demokritos	GR
Giacomo Bernini	Nextworks S.r.l	IT
Maria A. Serrano	Nearby Computing	ES
Mohammed al-rawi	University of Aveiro	PT
Ramón Agüero	University of Cantabria	ES
Souvik Paul	Lenovo	DE

Name	Company / Institute / University	Country
External Reviewer		
Luis Miguel Contreras Murillo	Telefonica	ES
Mikael Fallgren	Ericsson	SE

LIST OF CONTRIBUTORS

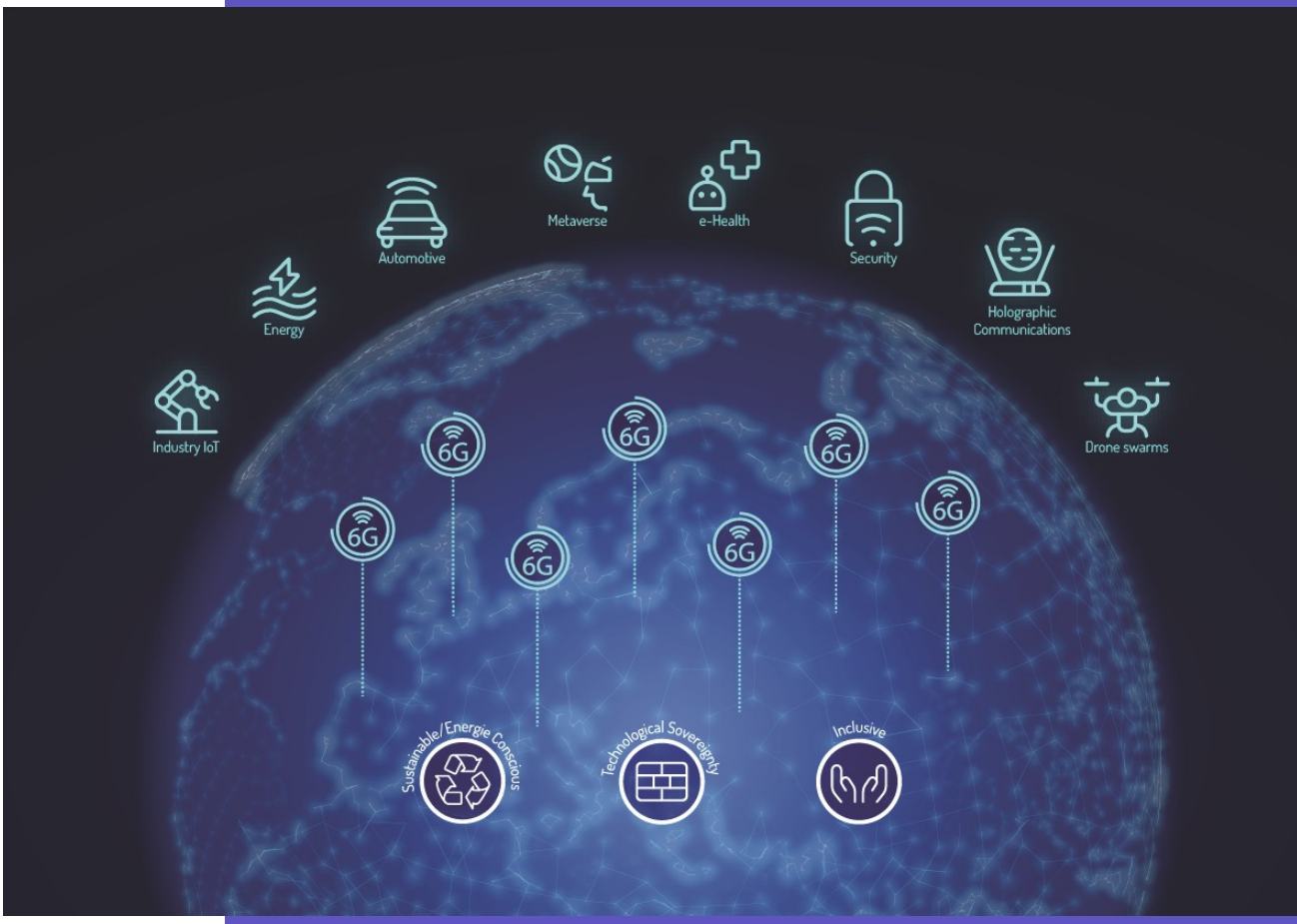
Name	Email
Adlen Ksentini	ksentini@eurecom.fr
Alejandro Fornes	alforlea@upv.es
Ana Kovacevic	ana.kovacevic@zentrixlab.com
Anastasios Giannopoulos	angianno@fourdotinfinity.com
Andreas Gavrielides	andreas.gavrielides@imec.be
Andreas Kunz	akunz@lenovo.com
Antonio Skarmeta	skarmeta@um.es
Antonios Lalas	lalas@iti.gr
Apostolis Salkintzis	salki@motorola.com
Artur Krukowski	krukowa@intracom-telecom.com
Beyza Bütün	beyza.butun@imdea.org
Christos Verikoukis	cveri@isi.gr
Cristian Vaca Rubio	cvaca@cttc.es
David de Andrés Hernández	david.deandres@imdea.org
Dimitris Tsolkas	dtsolkas@fogus.gr
Evangelos V. Kopsacheilis	ekops@iti.gr
Fabrizio Granelli	fabrizio.granelli@unitn.it
George Makropoulos	gmakropoulos@iit.demokritos.gr
George Tziavas	g.tziavas@ece.upatras.gr
Giacomo Bernini	g.bernini@nextworks.it
Giada Landi	g.landi@nextworks.it
Godfrey M. Kibalya	godfrey.kibalya@nearbycomputing.com
Harilaos Koumaras	koumaras@iit.demokritos.gr
Henning Sanneck	h_sanneck@apple.com
Jesus Perez Valero	jesus.perezvalero@um.es
Joao Fernandes	joao.fernandes@onesource.pt
Joaquín Cáceres	joaquin.caceres@eviden.com
Johann Marquez-Barja	Johann.Marquez-Barja@imec.be
Jorge Bernal	jorgebernal@um.es
José Luis	jose.carcel@eviden.com
Jose M Alcaraz Calero	j.alcarazcalero@aston.ac.uk
Juan Brenes	j.brenes@nextworks.it
Katerina Giannopoulou	kgiannopoulou@fogus.gr
Kostas Ramantas	kramantas@iquadrat.com
Luis Cordeiro	cordeiro@onesource.pt
Luis Ferreira	luis.ferreira@onesource.pt
Lusi Rosa	luis.rosa@onesource.pt
Marco Fiore	marco.fiore@imdea.org
Maria A. Serrano	maria.serrano@nearbycomputing.com

Maria Lamprini Bartsioka	mbarts@fourdotinfinity.com
Marius Iordache	marius.iordache@orange.com
Maxime Compastié	maxime.compastie@i2cat.net
Mir Ghoraishi	mir@gigasys.co
Mohammed Al-Rawi	al-rawi@av.it.pt
Nenad Gligoric	nenad.gligoric@zentrixlab.eu
Nikos Dimitriou	nikodim@iit.demokritos.gr
Nora Taleb	nora.taleb@ictficial.com
Paola Soto Arenas	paola.sotoarenas@imec.be
Pedro Tomas	pedro.tomas@onesource.pt
Qi Wang	qw96@leicester.ac.uk
Ramon Agüero	ramon.agueroc@unican.es
Sheeba Basharan	smary@lenovo.com
Shuaib Siddiqui	shuaib.siddiqui@i2cat.net
Sotirios Spantideas	sospanti@fourdotinfinity.com
Souvik Paul	spaul5@lenovo.com
Spyros Georgoulas	spygeorgoulas@iit.demokritos.gr
Tao Chen	Tao.Chen@vtt.fi
Tarik Taleb	tarik.taleb@ictficial.com
Vasileios Theodorou,	theovas@intracom-telecom.com,
Vasiliki Parousidou	vparousidou@intracom-telecom.com
Yan Chen	yan.chen@ictficial.com

SUPPORTING SNS JU PROJECTS

- 6G-BRICKS
- 6G-DALI
- 6G-INTENSE
- 6G-PATH
- 6G-VERSUS
- 6G-XCEL
- 6GCLOUD
- ADROIT6G
- ELASTIC
- ETHER
- FLECON-6G
- HORSE
- iTRUST-6G
- MULTI-X
- NATWORK
- ORIGAMI
- RIGOUROUS
- SAFE-6G
- SUNRISE-6G
- UNITY-6G

The SNS JU projects have received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme.



Website: https://smart-networks.europa.eu/sns-ju-working-groups/

Co-funded by the European Union

6GSNS