(one6G)

Taking communications
to the next level

AGENTIC AI AND FOUNDATION MODELS IN MOBILE COMMUNICATION SYSTEMS

WHITE PAPER

September 2026

one6g.org

Executive Summary

Agentic AI and foundation models are considered as fast-growing and native enablers for the evolution of mobile communication systems toward 6G and beyond, allowing embedded adaptive and intent-driven intelligence across potentially heterogenous network domains. This position paper considers the transition from connection/transmission-centric networks, largely based on predefined procedures and optimisation mechanisms, toward a dynamic and composable cognitive system with joint communication, computing, sensing, data, and artificial intelligence. We envision an autonomous and intent-oriented agentic system, capable of perception, reasoning, planning, collaboration, and action, while operating under explicit policies and governance. Foundation models provide the underlying capabilities for language understanding, multimodal processing, reasoning, and domain-specific knowledge, including through specialised Large Telco Models (LTMs). Realising this vision requires a shift from conventional network automation towards agentic AI-native mobile communication system. Dynamic intelligent communication is based on collaborative computing as required, leveraging distributed intelligence across the device–edge–cloud continuum. Different functions within an agentic workflow may therefore be executed dynamically at different locations depending on latency requirement, computational and energy resources, privacy requirement, network conditions, and service requirements. This distributed intelligence is native to multi-agent architectures, in which specialised agents coordinate tasks, communicate, access tools and APIs, or exchange amongst themselves, while maintaining intent-driven context.

Foundation models in mobile communication systems are considered from the perspective of their specialisation, distribution, and adaptation. Model components and AI functions can be distributed across devices, edge nodes, and cloud infrastructures, while agents can dynamically adapt model selection and execution according to operational conditions. The lifecycle of foundation models, including training, adaptation, validation, deployment, monitoring, and retraining, is also an important requirement for their deployment in mobile networks.

Building on these architectural foundations, the paper examines how agentic AI can translate application or network requirements (in terms of so-called intents) into adaptive network behaviour under policy-driven control. This includes the use of memory and reasoning for long-horizon decision-making, mechanisms for safe verification and conflict resolution, and approaches for achieving telco-grade behaviour while adapting to changing conditions. A clear distinction between probabilistic reasoning and deterministic network control is advocated. Self-improving supervision, deterministic enforcement, and the integration of foundation models and management functions further support the architecture.

Achieving the vision of a cognitive system fabric requires robust data pipelines, zero-trust governance, explainability, and standardisation to ensure interoperability and telco-grade performance. It must be ensured that autonomous operations remain trustworthy, accountable, and compatible with existing network practices. Security and governance need to evolve alongside the underlying intelligence, providing the conditions under which agents and foundation models can be integrated into mobile communication networks. In addition, the paper considers how this evolution may affect network operation and business models, including the way of managing resources, services, and outcomes, with the potential to improve performance and efficiency, reduce OPEX, and enable new services.

Finally, we identify several open challenges for agentic AI and foundation models to become an integral part of future mobile communication systems. Addressing them is essential for realising a flexible, efficient, composable, and sustainable vision of the intent-driven agentic AI-native 6G to build ubiquitous connected intelligence and enable innovative use cases and

solutions for societies and industries, and towards socio-economic and environmental sustainability.

Table of Contents

1. Introduction

For decades, cellular generations from start through early 5G were engineered around a dominant invariant: keeping the user connected, with growing features such as multimedia, mobile Internet, and smartphones. Under this legacy paradigm, the layered protocol stack and its operational procedures were designed to preserve coverage and continuity. Network optimisation loops largely operated within a rigid framework of fixed procedures and predefined behaviours, despite the introduction of sophisticated automation, self-organising networks (SON), and traffic prediction. Performance was measured exclusively by traditional transport Key Performance Indicators (KPIs) such as throughput, latency, reliability, and availability.

As the telecommunications ecosystem enters the 6G era, this conventional transmission model is reaching its architectural limits. Emerging cyber-physical services such as autonomous robotics, large-scale digital twins, industrial automation, and smart healthcare demand a network that does more than transport bits. These applications require deep context-awareness, real-time adaptability, and strict guarantees of end-to-end execution deadlines spanning communication and computation, while prioritising inclusive and ubiquitous context-driven and simplified connected intelligence.

1.1. The one6G Vision: The Cognitive System Fabric

This position paper establishes the one6G vision, framing 6G not as an incremental upgrade to wireless links but as a qualitative transition toward an intent-driven cognitive system. In this paradigm, communication, sensing, computing, and artificial intelligence are natively co-designed to support advanced cyber-physical workflows.

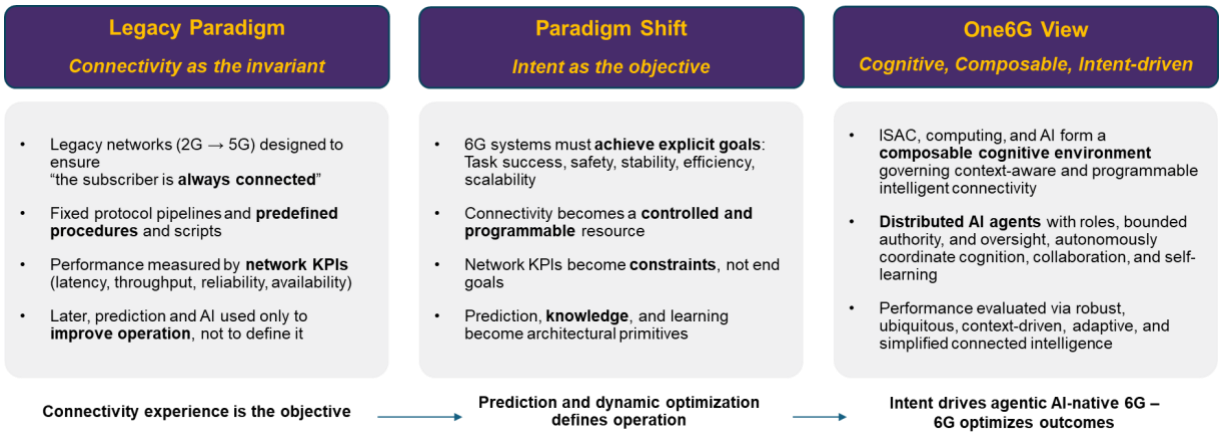

Figure 1.1. Transition from connectivity-based networking towards an intelligent system fabric

Under this system view, knowledge, prediction, and behavioural synthesis become first-class architectural primitives rather than auxiliary, vendor-specific add-ons. The network evolves from a passive transport substrate into an environment that perceives, reasons, and dynamically instantiates its own control loops to satisfy declarative business intents. Consequently, network performance can no longer be evaluated only by raw link metrics; it must be assessed via intent-oriented outcomes, such as task success probability, safety margins, control stability, ubiquity, and information freshness (Age of Information).

1.2. The Shift Toward Agentic and Composable Networks

While prior generation targeted leveraging closed-loop automation to tune parameters within fixed-protocol pipelines, it lacked the semantic reasoning capacity to dynamically orchestrate multi-dimensional, time-varying constraints. This structural deficiency mandates a shift toward Agentic AI, a framework wherein autonomous, intent-directed AI agents are embedded natively into the system fabric.

Agentic AI introduces the capability to handle seemingly open-ended operational complexity without relying on pre-engineered, static code paths. Rather than optimising isolated parameters defensively, these agents utilise Large Telco Models (LTMs) to comprehend abstract intents, evaluate trade-offs, and map multi-agent operations across varying network timescales. By transitioning from deterministic automation to an agentic ecosystem capable of behaviour synthesis, 6G networks can safely manage the extreme flexibility required by next-generation applications while operating under strict policy guardrails and safety-by-design boundaries.

1.3. New Paradigms in Enabling Applications in Mobile Communications

The fusion of 6G systems and agentic reasoning fundamentally alters how intelligence is deployed, orchestrated, and executed across live communication substrates:

- Industrial Cyber-Physical Systems (CPS): Powering real-time, safe cooperative motion of mobile robots and automated guided vehicles (AGVs) on factory floors. Agents manage hard end-to-end deadlines across transport and compute execution to preserve control stability and mission safety.
- Dynamic Workload & Split-AI Partitioning: Coordinating complex machine learning inference pipelines across the device–edge–cloud continuum. Local and supervisory agents dynamically assign model execution layers based on real-time channel conditions, compute loads, energy footprint limitations, and data privacy boundaries.
- Supervisory Physical Layer Configuration: Embedding self-improving agentic closed loops natively within the air-interface to dynamically configure massive MIMO arrays and advanced physical layer parameters, ensuring dependable connectivity for transient, high-priority nodes.
- Semantic and Multimodal Perception-Aware Networks: Leveraging Integrated Sensing and Communication (ISAC) alongside Large Multimodal Models (LMMs) to turn the network substrate into an environmental perception system. By communicating semantic meaning rather than raw data blocks, the network optimises for state consistency and situational awareness.
- On-Demand Service Provisioning via Code Generation: Integrating automated code generation ("vibe coding") pipelines within the cognitive network control. This allows the system to synthesise, test, and execute on-demand micro-services and control mechanisms in real-time digital twin sandboxes to meet immediate, highly localised business requirements

1.4. Structure of the Position Paper

The paper is organised into eight subsequent sections that detail the technical, operational, and governance pillars of this emerging cognitive fabric.

Section 2 outlines the key architectural principles of an agentic-native network framework, within a combined architectural and operational context such as the distribution of intelligence across the device-edge-cloud continuum, multi-agent coordination and agent communication aspects, and policy-driven control framework.

Section 3 examines the AI models driving these networks, focusing on large multimodal models and specialised Large Telco Models (LTMs) in the context of relevance, semantic perception and communication, collaboration, lifecycle, and self-learning

Section 4 explores mechanisms for translating application requirements into intent, focusing on memory models for long-horizon network optimisation, multi-dimensional constraint reasoning, and autonomous decision-making boundaries.

Section 5 outlines security, trust, and governance requirements for safe operation, from Zero Trust architecture for agentic systems to identity management, observability, and oversight, among others.

Sections 6 and 7 assess the realities of standardisation and commercialisation. Section 6 outlines the evolutionary standardisation through 3GPP releases, and across global standards bodies such as ITU, IETF, ETSI, TMF, and others. Section 7 translates these milestones into economic terms, examining token-based models for resource settlement, the impact of automation on OPEX and workforce design, multi-vendor interoperability, and the shift from volume billing to outcome-based SLAs and Sensing-as-a-Service.

Finally, Section 8 outlines a summary of key points and the road ahead along with open engineering and research questions. Section 9 provides concluding remarks and outlines the path toward mature 6G cognitive systems.

2. Architectural Principles for Agentic Mobile Networks

2.1. Agentic AI Native Network Architecture

The transition from legacy cellular systems to 6G requires a fundamental redesign of the core network's management and control architecture. While Software-Defined Networking (SDN) introduced programmable transport and separation of control and forwarding in IP networks, mobile core evolution has followed a different trajectory. Network Function Virtualisation (NFV) and the Service-Based Architecture (SBA) in 5G mobile core network enabled cloud-native service provisioning, service exposure, and feasibility of scaling of core network functions. Network slicing further added resource isolation and SLA enforcement for multi-tenant operations. Although these steps significantly improved flexibility, they remain bound to predefined control logic and static policy frameworks that limit the network's ability to autonomously adapt to emerging AI-native service behaviours.

Integrating AI/ML capabilities into 5G-Advanced network management represents a further step toward data-driven optimisation, intent-based management, and autonomous operation. These capabilities enable networks to use operational data for a variety of functions such as network traffic prediction, anomaly detection, resource optimisation, and energy efficiency, to support more effective and adaptive management based on context. Although such intelligence enhances network control, it does not fundamentally alter how operational decisions are shaped, coordinated and executed across the network. Addressing this gap requires a higher-level control capable of combining intent, shared knowledge, predictive intelligence and autonomous decision-making across programmable network functions.

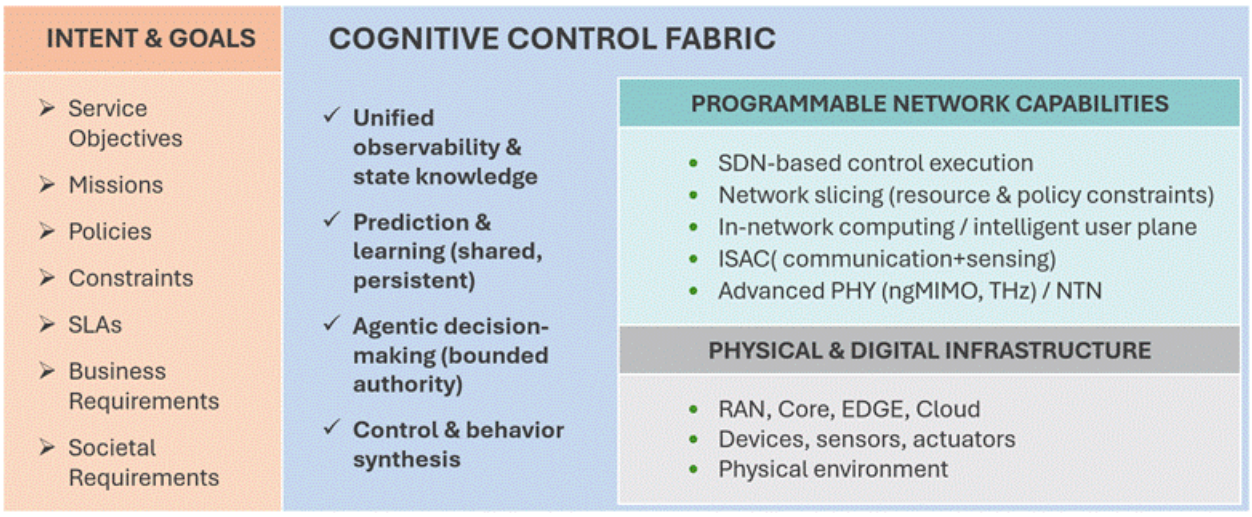

Figure 2.1. Architecture in the one6G composable-network view

As illustrated in Figure 2.1, the one6G architectural viewpoint introduces a cognitive network control that sits above these programmable capabilities. While slicing partitions resources, composability actively synthesises network behaviour from declarative intent templates. This framework functions as a system-level control abstraction that maintains unified observability, shared knowledge, and persistent predictive models. Within this architecture, autonomous, intent-driven AI agents possess the reasoning capacity to operate safely across varying time scales and authority levels, translating abstract objectives into real-time operational policies without relying on fixed procedures. Crucially, this agentic architecture

does not imply unconstrained autonomy; it enforces bounded authority, strict auditability, and stability-by-design to ensure full compatibility with legacy telecommunications standards.

This move toward autonomous, intent-oriented control introduces Agentic AI-native networking as an emerging architectural paradigm. Agentic-AI native networking builds upon the AI-native paradigm and goes beyond the integration of AI/ML models by embedding autonomous AI agents as native architectural entities. These agents are expected not only to predict or optimise but also to perceive the environment, reason, plan, collaborate execute, and self-improve, to achieve the intended goal, with autonomy and adaptability, working under constraints and guardrails, and meeting the reliability and telco-grade requirements.

As illustrated in Figure 2.2, realising this paradigm requires new architectural capabilities to efficiently integrate and enable the functions, services, communication, orchestration, and management needed for intent-based cognition, collaboration and action, and self-learning, by autonomous and adaptive AI agents. These capabilities span the infrastructure, intelligence, service, and application domains, with governance and control applied across them. Communication, computing, sensing, data, models, and storage provide the underlying resources to support agentic operations. Memory and knowledge enable agents to maintain context and share experience, while agent connectivity, communication, and tooling support interaction and access to network capabilities. Intelligence and cognition provide the perception, reasoning, execution, and learning functions required to translate intent into policies. Figure 2.3 provides an operational perspective, where intent-driven objectives are translated into agentic tasks, assigned to specialised agents according to their skills, roles, context, and constraints, and realised through multi-agent coordination, orchestration, communication, and knowledge sharing. Distributed intelligence and collaborative computing provide the execution foundation, while memory preserves context. Across layers policy-driven control governs what agents can do at every stage while they act autonomously.

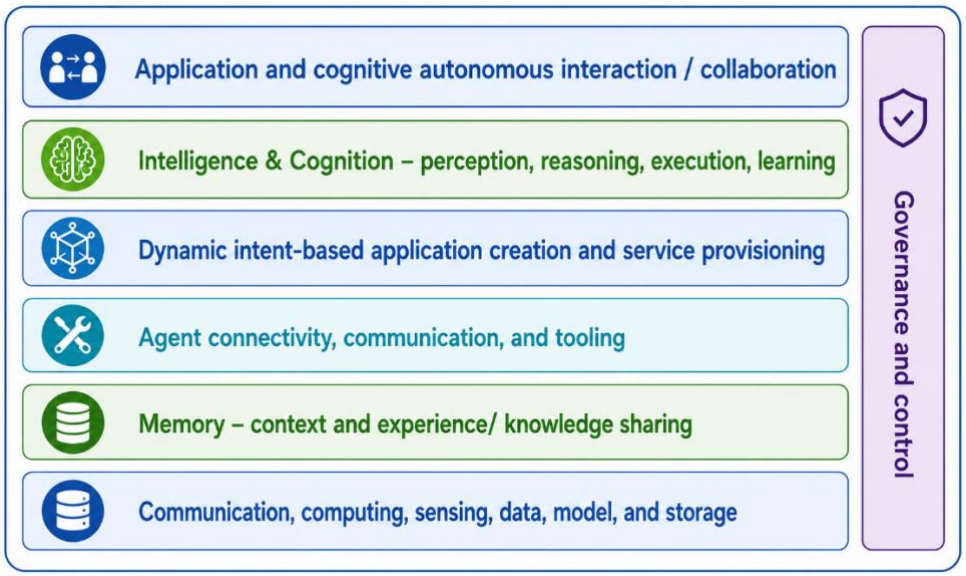

Figure 2.2. A high-level layered view of the agentic AI-native network architecture

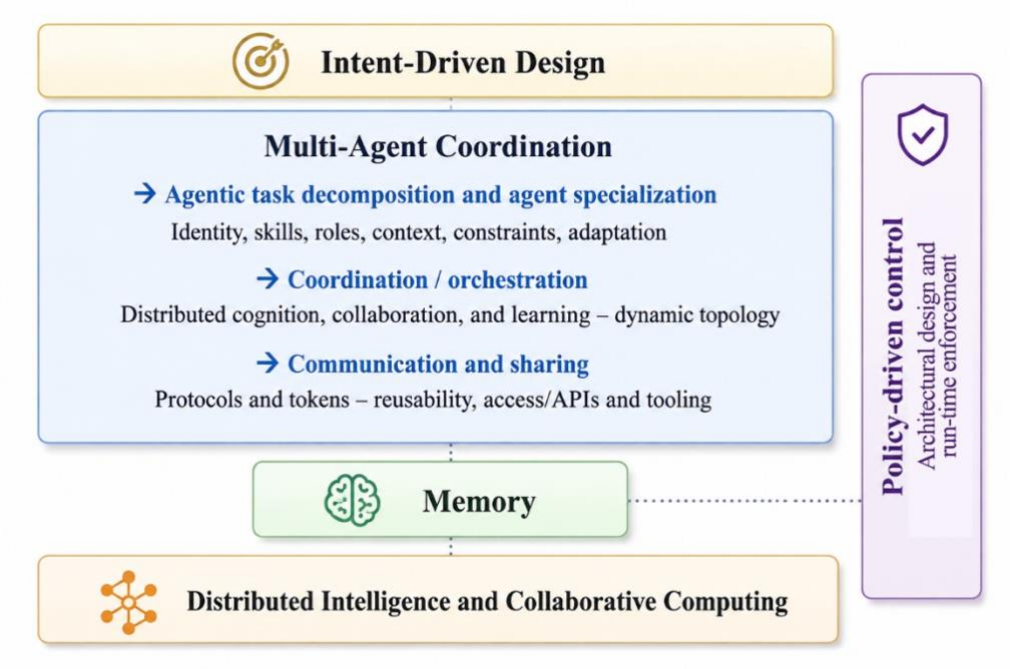

Figure 2.3. An operational simplified view of an agentic AI-native network

2.2. Distributed Intelligence Across Device–Edge–Cloud

As advanced cyber-physical applications increasingly depend on real-time inference and localised closed-loop control, performance is no longer determined merely by wireless link metrics. Instead, system efficiency, along with agility, service-awareness, and responsiveness, are affected by end-to-end execution pipelines that necessitate the joint coordination of communication and computing, moving away from the static assumption that computation resides permanently at the edge or within a centralised cloud.

Communication and Computing to Support Distributed Intelligence

Within this distributed intelligent framework, computing is dynamically composed alongside connectivity under explicit intent and policy constraints. Through mechanisms such as In-Network Computing (INC), the system coordinates data delivery, compute placement, and execution scheduling across devices, edge nodes, and domain-level compute pools. This architectural coupling is essential for supporting collaborative computing, such as Split-AI. Future mobile agentic AI architecture should dynamically support local, split, distributed, federated and centralised modes, rather than assuming a single deployment model. This context is also well explored in one6G work on INC [1], [2], [3]. By managing these boundaries via an intelligent agentic resource layer, the network dynamically navigates trade-offs between local execution constraints and global objectives to meet strict performance requirements.

The device-edge-cloud continuum is the substrate on which agentic distributed intelligence in 6G is realised. An agentic network distributes cognition across architectural layers with fundamentally different cost and value profiles:

- On-device compute offers low-latency response and data locality but is more bounded in memory, energy, and model capacity. It is suitable for

local sensing, privacy-sensitive processing, immediate inference and safety-related reaction. User equipment, sensors, vehicles, robots, wearable devices and XR terminals should not be treated only as passive data sources. They may also act as local intelligence points, extracting task-relevant context, executing lightweight models, enforcing local privacy requirements, and deciding which features, events, model updates or intermediate outputs should be shared upward. This supports a raw-data-minimising design, where sensitive or high-volume data can remain close to the source and only necessary information is exchanged.

- Edge, deep-edge and RAN-domain intelligence add substantial compute within a limited propagation delay. They can support low-latency coordination, context fusion, mobility, resource prediction, local optimisation, and near-real-time control. This layer is close enough to users and network elements to react to changing radio, traffic and mobility conditions, while still having more computing capability and local visibility than individual devices. Edge/RAN intelligence may also aggregate telemetry, coordinate nearby devices or cells, host regional models, and support policy-aware local decisions.
- Cloud and cross-domain intelligence supply the larger foundation models and the broader context at higher and more variable round-trip latency. Cloud can provide broader learning, governance and coordination functions, including large-scale model training, global optimisation, model repositories, lifecycle management, cross-domain analytics, policy coordination and large-scale validation. In this role, cloud intelligence complements device and edge intelligence by maintaining global consistency, traceability and operator-level control.

The optimal placement of a given agent, model, or sub-task thus depends on the instantaneous latency budget, available bandwidth, energy envelope, privacy constraints, and reachability.

Distributed Intelligence in Agentic Workflow Lifecycle

In an AI-native mobile network, AI functions should be considered as part of the network design, deciding which intelligence functions should run where, at which lifecycle stage, and under which policy or operational constraints [4], [5]. Different functions within the same agentic workflow may benefit from being executed at different locations. Representative examples can be found across the agent lifecycle, including perception, planning, execution, and memory updating. These examples illustrate how distributed learning and inference can reduce sequential bottlenecks, support real-time scaling, preserve data locality and privacy, and enable context-aware collaboration across heterogeneous computing resources:

- Perception: By deploying lightweight computer vision or sensory-processing models at the edge, agents immediately ingest multimodal inputs and translate them to tokens. For instance, a traffic camera processes video locally and sends only parsed events to the agent. The computational overhead of diffusion models makes them unsuitable for execution on

resource-constrained edge devices. A more efficient solution for distributed Agentic AI is to represent observations as semantic tokens and offload the diffusion decoding process to the cloud or base station.

- Planning: distributed computing enables multi-agent swarms where specialised nodes (e.g., a "search agent" or a "coding agent") work simultaneously to decompose and evaluate sub-tasks. This can exponentially expand long-horizon planning capabilities and minimises the risk of the primary model drifting off course. It also supports dynamic routing, where an agent chooses a simple local model for easy tasks and scales to a frontier model for complex reasoning.
- Execution: Distributes the "decode" and "prefill" steps of inference to processing units located physically close to the APIs, databases, or users they interact with. This drastically reduces cumulative round-trip latency. It also ensures that compliance regulations are met by running actions inside secured, localised networks.
- Memory updating: Through distributed learning techniques such as federated learning, we employ collaborative and federated learning protocols to continually update the agent's memory banks and prompt configurations based on real-time feedback, without exposing raw user data. Agents securely learn from past task outcomes and human feedback, sharing these global improvements across the swarm while keeping sensitive data private.

This turns workload placement from a static, design-time assignment into a runtime, agentic decision. In a largely software-defined and increasingly AI-native architecture, a perception, control, or reasoning function is exposed as a schedulable service whose execution tier can be re-bound continuously as network conditions evolve on sub-second timescales. The placement agent must therefore be predictive rather than reactive: it must anticipate near-future access-segment behaviour from observable radio and context state. Furthermore, placement decisions should reduce unnecessary raw-data movement, protect sensitive processing close to the source, and interface with governance mechanisms. This helps avoid turning distributed intelligence into another source of operational complexity.

A production-grade distributed-intelligence layer needs certain qualities beyond simple offloading. It should be risk-aware, favouring the safer choice when a prediction is uncertain. It should be stable, avoiding constant switching between tiers, since every switch is a real reconfiguration that costs time and signalling, while still allowing fallback when a tier becomes unusable. Finally, it should be auditable, recording a clear reason for each decision so operators can trace and govern the system's behaviour.

Simple "always use the edge" rules break down exactly where traffic is heaviest, so predicting conditions per task is necessary for any latency-sensitive service, such as teleoperation, drone inspection, or mobile XR, to choose between architectural layers of distributed intelligence, dynamically.

Delivering, assuring, and leveraging distributed intelligence need lifecycle management. AI functions in mobile networks should be managed across the full lifecycle, including training, validation, deployment, continuous monitoring and retraining. For example, a model may be trained centrally, validated in a non-production or emulated environment, deployed at the edge, executed at the device, monitored by a management function, and retrained when drift or performance degradation is detected. The purpose is to keep AI functions deployable, monitorable and updatable across the system [6].

As computing is dynamically composed within this distributed intelligent communication framework, under explicit constraints, intent-based architecture must be well defined and designed, and the coordination among agents and entities becomes critical.

2.3. Multi-Agent Coordination Models

The transition from monolithic Artificial Intelligence (AI) systems toward agentic architectures represents one of the most significant shifts enabled by foundation models. Rather than relying on a single model to perform all reasoning, planning, optimisation, and execution tasks, agentic systems decompose complex objectives into smaller, more manageable subtasks, which are assigned to multiple specialised agents [7]. This paradigm is particularly relevant for future mobile communication systems, where network intelligence will need to operate across multiple domains, timescales, and layers, ranging from radio resource management and network orchestration to energy optimisation and service assurance.

2.3.1. Task Decomposition and Agent Specialisation

The primary motivation for adopting a multi-agent architecture is the *divide-and-conquer* principle. Complex network management problems are often naturally decomposable into smaller tasks that can be executed in parallel or semi-independently.

This decomposition can reduce overall execution time and improve responsiveness by allowing several activities to progress simultaneously. It also enables specialisation, since each agent can operate with task-specific models, tools, data, and domain knowledge, thereby reducing the likelihood of hallucinations and improving the reliability of its outputs.

A decomposed architecture enhances modularity by allowing individual agents to be upgraded, replaced, or maintained independently, avoiding the need to redesign the whole system. It also enables better fault isolation, restricting failures to specific components or subtasks instead of spreading across the entire framework. Additionally, the architecture can be scaled flexibly by dynamically adding or removing agents based on network load, service demand, or problem complexity.

Task decomposition can be implemented using various strategies, including functional, hierarchical, spatial, or domain-oriented approaches. In functional decomposition, agents are assigned specific roles like planning, monitoring, optimisation, validation, or execution. Hierarchical decomposition involves high-level orchestrator agents that oversee, coordinate, and compile the efforts of lower-level worker agents. Spatial or domain decomposition assigns agents to different geographical regions, administrative domains, network layers, operators, slices, or technological subsystems.

The effectiveness of decomposition depends strongly on minimising interdependencies among the resulting tasks. When subtasks are tightly coupled or require frequent information exchange, the resulting communication and synchronisation overhead may outweigh the benefits of parallel execution. Appropriate decomposition should therefore not only divide a complex problem into smaller components but also minimise dependencies and information exchanges between them.

2.3.2. Best Practices and Trade-offs in Agent Granularity

Determining the appropriate number of agents is one of the central architectural decisions in multi-agent systems. While increasing the number of agents generally enhances parallelism and specialisation, it also comes with several costs [8]. These include greater token consumption

due to additional context windows, communication overhead, and higher computational complexity. Furthermore, energy consumption can rise, along with synchronisation delays and larger memory requirements. Additionally, having more agents complicates the debugging and validation procedures.

Consequently, the performance gain obtained from adding agents exhibits diminishing returns beyond a certain scale. Excessively fine-grained decompositions may lead to situations in which agents spend more resources coordinating than on solving the original problem. In this regard, a useful design principle is to *maximise task independence while minimising coordination requirements*. This, in turn, suggests identifying the number of inherently independent tasks and allocating a corresponding number of agents to execute them.

To this aim, several best practices emerge:

- Create separate agents only when the subtasks can progress largely independently;
- Avoid splitting sequential workflows into multiple agents;
- Allocate specialised models only when domain expertise significantly improves performance;
- Dynamically scale the number of active agents according to workload and network conditions.

Future mobile networks may particularly benefit from adaptive agent population management, in which orchestration agents instantiate or terminate worker agents depending on traffic conditions, network slices, or service-level agreements.

Multi-agent coordination inherently involves two related but distinct questions. The first is architectural: how many agents there should be, how their roles should be specialised, how tasks should be decomposed, and how coordination should be organised. This is the focus of the present section. The second is technical: which communication protocols, semantic exchange mechanisms, and interoperability models enable agents to discover one another, expose capabilities, exchange context, negotiate workflows, and return verifiable artefacts. These mechanisms are treated later.

2.4. Policy-Driven Control

The transition to agentic AI-native mobile networks will be a shift to a highly dynamic policy framework and control logic that enables networks to autonomously choose, adapt, optimise, deliver, and self-improve. When AI agents in a communication system perceive, plan, decide, collaborate, and act, while reasoning, accessing, calling, retrieving, choosing, chaining, and sharing, the governance paradigm and currency transform and can no longer rely on pre-defined scripts or static frameworks, reviews, and audits. Policy-driven control therefore expands from managing predefined network and service behaviour to managing the decision-making and operational authority of autonomous agents. This is particularly important as agents become capable of interpreting intent, applying actions, continuously evolving through learning and adapting their behaviour to changing network conditions.

This paradigm shift requires policy-driven control within the cognitive system fabric that is sensitive to these independent actions and the changing behaviours, typically in multi-modal, multi-domain, collaborative, and distributed environments. It demands architectural design attributes and operational agility to accommodate this, where control carries policy showing up as a machine-readable and explicit set of rules and codes that can be enforced in runtime. As such, policy-driven control should handle the complete agentic lifecycle (from perception and reasoning to planning, collaboration, execution and learning), while taking into account agent goals and roles, identities and privileges, skills, context and memory, constraints, timing, and lifecycle state. As a result, policies become dynamic and context-aware rather than preconfigured. Their machine-readable form also allows policy decisions to be separated from execution, where one agent decides what is allowed and another may apply the decision.

The transformation to unleash the prospects of autonomous, cognitive, collaborative, and self-improving agentic systems will make possible the co-existence of two foundational design principles: To assure expected reliability, service quality, trust, resilience, efficiency, performance, accountability, and outcome, while embracing and enabling agency; orchestrating multi-agent powerful systems with autonomous, creative, and adaptive agentic AI planning, reasoning, intelligent choices, and decisions. The objective is not to excessively restrict agency, but to establish an operating environment within which autonomy can safely be applied. The architectural framework and the agency are interdependent. More specifically, the framework establishes integrity, trust, performance, resilience, and compliance requirements, while agency provides the flexibility to autonomously determine how these requirements can be effectively achieved. This is especially important as independent agents can decompose tasks, assign authorities, collaborate, and adapt their actions while adapting their behaviour to shared and potentially competing objectives.

Policy-driven control shall ensure alignment with goals, and with associated envelopes and boundaries in terms of identities, roles, privileges, context, and constraints. It observes and enforces at runtime what may be necessary to avoid or resolve vulnerabilities, risks, misalignments and drifts, misuse, degradations, competing or conflicting objectives, access or actions, error propagation, safety or data protection gaps. It facilitates runtime governance through observing and tracing, evaluations, validations, and compliance, efficient escalations, and necessary oversight triggers and interceptions. These controls may establish thresholds, boundaries and guardrails, evidence and validation requirements, zero-trust access principles, sandboxing, continuous observation and tracing, and mechanisms for escalation, interception, or human oversight. As a result, runtime execution can permit an agent action, constrain it, reject it, request additional validation, or escalate it when the action is beyond the agent's authorisation or expected capability.

3. Foundation AI Models for Mobile Communication System

3.1. Foundation Models for Telecommunication Systems

To materialise a cognitive network control capable of translating high-level business goals into optimised, localised network behaviours, the underlying intelligence must move beyond generic, offline machine learning models. General-purpose models lack the domain-specific vocabulary, structural understanding of cellular protocol stacks, and the deterministic telemetry interpretation required for telecom environments. Therefore, the network must natively embed domain-specialised Large Telco Models (LTMs). These models serve as the cognitive engine for distributed AI agents, allowing them to parse declarative intent templates, reason about multi-dimensional transport and compute constraints, and dynamically synthesise configuration changes across network layers.

Foundation models can support different modalities and functions within the mobile communication system. Large Language Models (LLMs) can provide language understanding and reasoning capabilities, while Large Multimodal Models (LMMs) can jointly process information from multiple modalities and support context-aware perception and communication. Other foundation-model architectures, such as diffusion models, can support the generation or content. Their applicability to mobile communication systems depends on the data, modality, task, and operational requirements of the respective use case.

For telecom-specific applications, these foundation models can be adapted using telecom-domain data and knowledge, resulting in domain-specialised models such as LTMs. Such models can incorporate domain-specific knowledge related to network protocols, topology, telemetry, services, and operational processes, enabling more specialised processing and reasoning within the mobile communication system. [9]

3.2. Collaborative Models: Compression, Distillation, Collaboration and On-Device Execution

Next generation networks are expected to enable the collaborative execution of AI models across heterogeneous network entities. Instead of executing an entire AI model on a single device or centrally located infrastructure, different AI model components, referred to as AI model partitions, may be distributed across end devices, edge nodes, and cloud data centres for collaborative execution. This model partitioning can contribute to effective resource utilisation, energy efficiency, scalability, privacy preservation, and overall improved performance, while enabling the deployment of increasingly complex AI models in resource-constrained environments.

Agentic AI Collaborative Model Execution

AI model partitioning requires defining the appropriate splitting points based on a variety of criteria (e.g., resource availability, service requirements, system constraints). More specifically, initial model layers could be executed locally on the end device level and their output is transmitted either to more capable end devices or more powerful edge or cloud nodes to continue the execution of the remaining part of the AI architecture. On the one hand, local execution does not require raw data transmissions and as such enhances data privacy, and decreases network resource utilisation. On the other hand, executing computationally intensive model partitions at the edge or cloud could reduce the computational burden imposed on resource-constrained devices. However, the transmission of intermediate model outputs may introduce additional communication overhead and network delays, especially when such outputs are high dimensional or when the available communication channel capacity is limited.

The dynamic splitting and assignment of model partitions introduces several challenges. Firstly, model partitioning and placement decisions should consider the heterogeneity of the involved devices in terms of computational capabilities, memory availability, energy resources, and supported hardware accelerators. Secondly, variations in network conditions may affect the feasibility of transferring intermediate model outputs within the required response time. Thirdly, the selected partitioning strategy should preserve data privacy, since intermediate representations may still expose sensitive information. Finally, frequent model migration, repartitioning, and overall coordination may impose additional communication and computational overhead, potentially reducing the benefits of collaborative execution.

Agentic AI could enhance collaborative model execution by dynamically deciding how an AI model architecture should be distributed across the underlying infrastructure. Agents could continuously monitor device capabilities, resource and energy availability, network conditions, task complexity, data sensitivity, and service requirements [10]. Based on this contextual information, they could determine the most appropriate model architecture, compression level, execution location, and splitting point. As a result, model execution would not rely on a predefined and static deployment configuration, but could be dynamically adapted according to changes in the operational and network environment. For example, an agent could select a compressed or distilled model for on-device execution when the task complexity is low or when stringent performance and privacy requirements are present. In cases where the local device cannot satisfy the required AI model performance, the agent could partition the model and offload selected computational tasks to a nearby edge node. More computationally demanding tasks, requiring advanced reasoning, could be assigned to cloud infrastructures.

Efficient Small and Large Model Collaboration

Model collaboration is the coordinated use of multiple specialised models to solve complex tasks that exceed the capability of any single model, enabling systems to combine complementary strengths and operate with greater accuracy, resilience, and adaptability. It is needed because modern problems, especially in telecom sector, are too multidimensional for one model to handle alone: data sources are distributed, decisions require cross-domain knowledge, and real-time environments demand both speed and precision. By integrating models through ensemble cooperation, pipeline workflows, or federated collaboration, organisations can achieve higher performance, reduce failure risks, and respond more effectively to dynamic conditions.

Model collaboration provides a natural paradigm for mobile communication systems: a small and fast model can be deployed on the device or at the edge for low-latency execution, while a larger and more capable foundation model can be deployed on the network side to provide stronger reasoning when needed. A key challenge, however, is that conventional collaboration often asks the larger model to perform open-ended generation. This not only increases inference latency and communication overhead, but also makes subsequent distillation

difficult, since the small model must imitate the larger model's complex generative distribution under its own capacity limits. As a research track, Select to Think (S2T) addresses this bottleneck by simplifying LLM assistance from free-form generation to discrete selection: at reasoning divergence points, the larger model only selects among the small model's top candidate tokens. This reformulates the distillation target into a lightweight selection problem over a local candidate set, enabling the small model to internalise the selection behaviour and achieve significant performance gains without inference-time external LLM calls [11].

Data-Local, Edge-Mediated Collaborative Adaptation

In addition to compression, distillation and model partitioning, collaborative models for mobile agentic AI should also support data-local collaborative adaptation. The key issue is not only where different parts of a model should run, but also what information can be safely shared across distributed devices, edge nodes and network domains. In many mobile scenarios, raw data cannot be freely pooled because of privacy, ownership, bandwidth, latency, energy or regulatory constraints. A practical approach is therefore to keep data close to its source and share selected knowledge instead, such as model updates, lightweight adapter parameters, embeddings, statistics, task-level summaries or other lightweight knowledge representations [12].

In this setting, edge and RAN-domain nodes are not only execution points. They can also act as collaboration coordinators. For example, an edge node may select suitable clients, group devices with similar radio, mobility or task context, adjust update frequency, filter low-value or high-cost updates, and decide whether an update should remain local, be aggregated at the edge, or be passed towards a cloud-level model. This is important because mobile data is often non-IID, dynamic and unevenly distributed across devices, cells, sites and operator domains. Mobility, intermittent connectivity and heterogeneous device capabilities also mean that not every device should participate in every collaborative update round [12], [13].

Federated and privacy-preserving collaboration should therefore be treated as a selective mode, not as the default solution for all tasks. For some tasks, local-only adaptation may be sufficient. For others, edge-level federation can support regional, cell-specific or site-specific adaptation. Cross-edge or cloud-level aggregation may be useful when broader generalisation is needed. The choice should depend on task complexity, latency target, energy budget, model size, data sensitivity, communication cost, data distribution and the expected gain from collaboration.

This provides a useful complement to hybrid execution and model partitioning. Model partitioning asks where different parts of a model should run across device, edge and cloud. Data-local collaborative adaptation asks what knowledge can be safely shared, when it should be shared, and how much communication cost is justified by the expected improvement. This helps collaborative AI remain practical for mobile agentic systems without assuming that all data, models or decisions must be centralised.

3.3. Model Lifecycle, Fine-Tuning, and Continuous Learning

Existing literature describes the lifecycle of foundation models using different stage definitions and levels of granularity [14], [15]. The lifecycle considered in the following follows the consolidated view presented in [16], which – as illustrated in Figure 3.1: Foundation model-based AI system lifecycle (taken from [16]) - groups the lifecycle into three overarching phases:

- Phase 1: Data Preparation (including data creation and data curation)
- Phase 2: Model Construction (including model training and model adaptation)
- Phase 3: System Operation (including management and deployment)

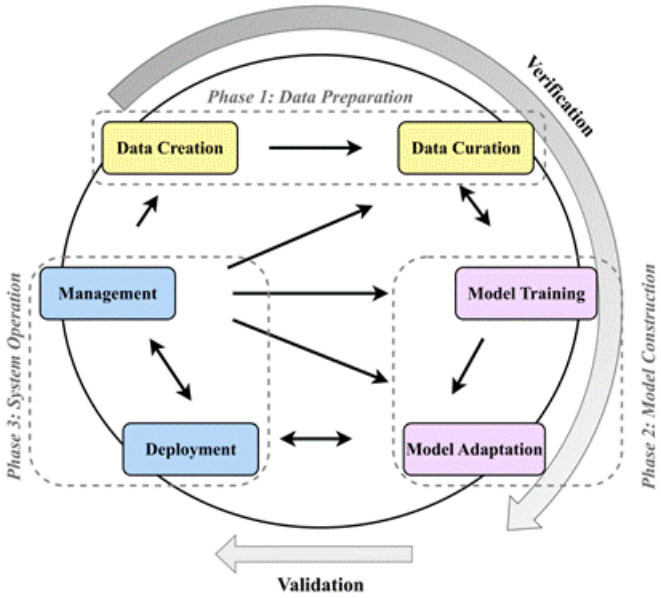

Figure 3.1. Foundation model-based AI system lifecycle [16]

The lifecycle is iterative, with feedback between the different stages and phases, while validation and verification provide cross-cutting activities throughout the lifecycle.

The lifecycle of foundation models in mobile communication systems introduces several challenges: Model updates must account for changing network conditions, evolving service requirements, and potential model drift while avoiding degradation of operational performance. Before updated models are introduced into the operating network, their behaviour and performance therefore need to be validated under representative network, device, and data conditions. At the same time, repeated fine-tuning, retraining, and model updates can introduce additional computational, energy, and communication overhead, particularly when models are adapted across distributed devices, edge nodes, and network domains. The complexity/dynamicity of mobile data, together with mobility and heterogeneous device capabilities, further complicates collaborative adaptation and update coordination. These aspects motivate lifecycle mechanisms that support controlled model updates, validation, deployment, monitoring, and, where necessary, further adaptation or rollback.

Model Adaptation / Fine Tuning

Fine-tuning enables a pre-trained foundation model to be adapted to specific domains, tasks, or operating contexts by further training it on targeted data. In mobile communication systems, this provides a mechanism for deriving domain-specialised models, such as Large Telco Models (LTMs), from general-purpose foundation models and adapting them to the specific characteristics of individual network domains, services, or deployment environments. Fine-tuning can be performed centrally or through data-local collaborative adaptation across device, edge, and cloud resources, depending on factors such as data sensitivity, available resources,

latency requirements, and the desired degree of specialisation. As network conditions and operational requirements evolve, repeated fine-tuning can support the continuous adaptation of deployed models.

Continuous Learning

Continuous learning enables foundation models to be updated over time based on newly available data, observed changes in the operating environment, and evolving service or network requirements. In mobile communication systems, this can support the continuous adaptation of domain-specialised models by incorporating new network observations and application-specific data. Continuous learning can involve repeated fine-tuning or retraining and may leverage the distributed intelligence capabilities of device, edge, and cloud resources. This enables models to remain aligned with changing network conditions while supporting the paper's vision of adaptive and continuously evolving AI capabilities across the 6G system. As mentioned above, repeated fine-tuning can support the continuous adaptation of deployed models, while rollback to a previously validated model version can be used when an update does not meet the required performance or operational conditions.

4. Key Technical Considerations for Agentic AI-Enabled Mobile Communication Systems

4.1. From Use Cases and Requirements to Intent

The operational core of the cognitive network control relies on transitioning from traditional network Key Performance Indicators (KPIs) to intent-oriented system performance. Legacy KPIs such as raw throughput, latency, reliability, and availability bounds remain foundational as enabling constraints, but they are no longer treated as end goals. Emerging cyber-physical services are defined strictly by whether they successfully achieve high-level operational objectives, shifting the primary focus to intent-oriented outcomes.

Table 4.1. Mapping of Intent-Oriented Outcomes, Perception Layers, and Transport Constraints

Level	What is Optimised	Example Metrics	Design Implication
Intent-Oriented	Outcome success	Task success probability, safety violations, control stability margins, mission completion time	Connects networking directly to application objectives
Information & Perception	Quality of situational knowledge	Localisation error, environment inference error, Age of Information (AoI), confidence/uncertainty	Couples sensing and learning directly with resource control
Network (Transport)	Delivery constraints	Latency/reliability bounds, throughput, spectral/energy efficiency	Remains essential as an enabling constraint

Every vertical engagement starts in the customer's own language. A car manufacturer asks for wireless control of the automated guided vehicle (AGV) fleet that never stalls the line; a mobile operator asks for no complaints about video quality during the evening peak; a municipality asks for a network that keeps the traffic-camera grid alive across the whole city without inflating the energy bill. None of these statements can be executed by a network. They are goals, not instructions: they name an outcome, leave the scope implicit, quantify nothing, and say nothing about what should happen when the goal cannot be met. Earlier discussion sets out why intent-oriented outcomes, rather than raw transport KPIs, are the right currency for such requests. What follows is the step immediately after that: how a goal expressed in the vertical's words is

turned into an object a management system can act upon, and what has to be added along the way.

Refining a high-level, vertical-specific intent into a machine-readable one is not a translation exercise. It closes four gaps, each addressed by adding a specific piece of structure (Figure 4.1):

- Scope. The business statement refers to “the plant”, “downtown”, “the city”. The refined intent must point to a named managed object and to the conditions that delimit it: a geographical polygon, a tracking area (TA) list, a public land mobile network (PLMN), a frequency layer, or a service type. Without this binding, the intent cannot be routed to a producer that owns the corresponding resources.
- Measurability. “Never stalls the line” must be expressed as one or more quantified targets. In the 3GPP intent-driven management framework [17], a target is a tuple of an attribute, a condition, and a value range, for example, an uplink latency below 5 ms. This is where domain knowledge becomes unavoidable: someone, or something, has to know that “the line does not stall” maps to a latency bound plus a per-device throughput floor, and that 5 ms is the right number for that particular control loop.
- Conditionality. Business goals are almost never unconditional; they hold in a context and under a load. Each target may therefore carry its own context list, again as attribute, condition, and value-range tuples, so that a requirement such as keeping the handover failure rate below 2 % when cell load exceeds 80 % can be stated without over-constraining the rest of the operating envelope. Contexts prevent well-meaning intent from pushing the network into a permanently over-provisioned state.
- Accountability. A machine-readable intent declares how it is judged. It carries a priority, so that competing intents can be arbitrated rather than silently overwritten; an observation period, so that “met” has a time base; and a fulfilment report through which the producer states whether the intent is fulfilled or not fulfilled and, in the latter case, why. This is what makes the intent a control loop rather than a configuration push: the refined intent is simultaneously the request and the acceptance criterion.

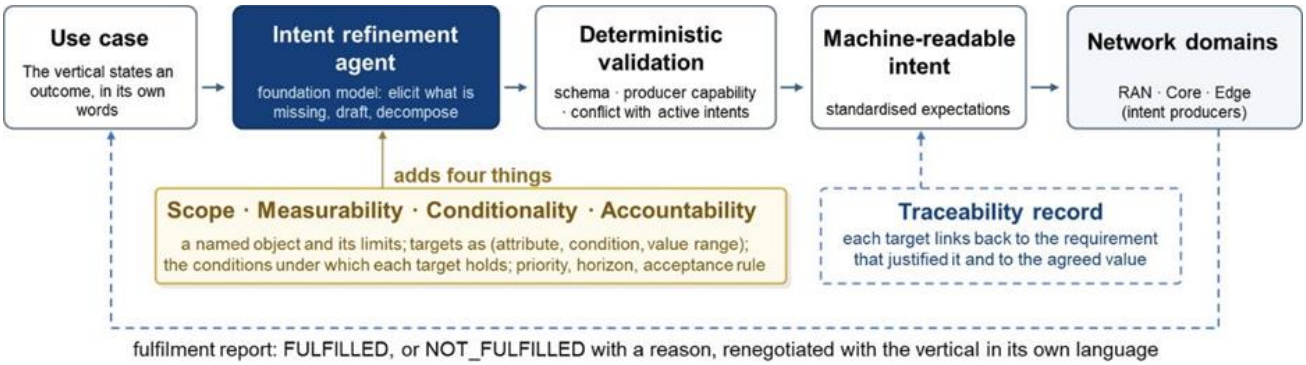

Figure 4.1. From a vertical use case to a machine-readable intent, and back. The refinement adds scope, measurability, conditionality and accountability; deterministic validation, not the model, decides what reaches the network; fulfilment reports close the loop and drive renegotiation in the vertical's own language

The resulting structure is consistent with the generic intent information model defined in 3GPP TS 28.312, in which an intent is decomposed into expectations associated with managed objects, contexts, and measurable targets. The fundamental structure can be illustrated by a small fragment of the industrial IoT case:

```
...
intentExpectations:
  - expectationId: "exp-agv-radio-service"
    expectationVerb: ENSURE
    expectationObject:
      objectType: Radio_Service
      objectInstance: "PrivateNetwork-PlantB-URLLC"
      objectContexts:
        - contextAttribute: ServiceType
          contextCondition: IS_EQUAL_TO
          contextValueRange: "URLLC"
    expectationTargets:
      - targetName: UlLatency
        targetCondition: IS_LESS_THAN
        targetValueRange: 5        # ms
...
```

Here, the high-level requirement that the AGV control service should not stall is bound to a specific radio-service object and expressed through an explicit, verifiable target—in this simplified fragment, uplink latency below 5 ms. The example is deliberately partial: a complete intent may include additional scope, contexts, targets, priority, and observation information. Detailed YAML representations are not reproduced here, since TS 28.312 specifies the corresponding information model and provides scenario-specific examples in Annex D. Table 4.2 instead focuses on the more relevant issue for the cognitive network control: how different vertical goals are refined into these formal constructs.

Table 4.2. The same refinement applied to three verticals

Aspect	Industrial IoT (AGV motion control)	Classic cellular (city-centre experience)	Smart city (sensing grid)

Who states the intent	Plant automation owner	Operator service-assurance team	Municipality, as an enterprise customer
Expectation object	Radio_Service, on a private network	RAN_SubNetwork	5GC_SubNetwork plus EdgeService_Support
Scope contexts	Coverage polygon; service type	Tracking areas; radio access technology; evening time window	Data network name; tracking areas served by one edge site
Dominant targets	Uplink and downlink latency; guaranteed per-AGV throughput	Share of poorly served users; weak-coverage ratio; RAN energy consumption	Registered subscribers and PDU sessions; per-camera throughput and latency
What good means	Worst case: the tail decides whether the line stops	Distribution: a small share of unhappy users, not a good average	Aggregate capacity, with a low duty cycle per device
Time horizon	Seconds, within a production shift	Minutes, within a recurring daily peak	Hours to days
Priority relative to other intents	Highest: production and safety	Medium: commercial experience	Lowest: yields to commercial traffic
Hardest part of the refinement	Turning a control-loop tolerance into a latency bound	Choosing a ratio rather than a mean, and bounding energy without starving the experience targets	Splitting one sentence into two expectations owned by two different producers

The industrial IoT case maps naturally onto the Radio Service expectation defined for delivery of a radio service (Clause 5.1.2 and Annex D.9 of TS 28.312), using service and geographical contexts together with latency and throughput targets. The classic cellular case can be expressed through the standardised Radio Network expectations for coverage and UE-throughput assurance and RAN energy saving (Clauses 5.1.4, 5.1.5 and 5.1.7; Annex D.3-D.5). The smart-city case is intrinsically cross-domain: its connectivity requirement maps onto the 5GC SubNetwork expectation, while local video analytics maps onto Edge Service Support (Clauses 5.1.8 and 5.1.3; Annex D.7 and D.2, respectively). The latter illustrates an important refinement operation: a single vertical goal may need to be decomposed into multiple expectations, each handled by a different network domain.

The relevant challenge for the cognitive network control is therefore not the encoding itself, but how the high-level vertical goal is transformed into these formal constructs. This refinement chain is precisely the kind of task for which foundation models are well suited: linguistic at the top, ambiguous during refinement, and formal at the bottom. A practical division of labour follows in which a foundation model performs elicitation and drafting, while deterministic

components perform validation and commitment. An intent-refinement agent can parse the vertical's request, detect missing information such as an unstated coverage area, time window, or numerical target, and request that information rather than inventing a plausible value. It can then propose a decomposition into standardised expectations and map the resulting goal onto the appropriate objects, contexts, and measurable targets.

Before commitment, however, deterministic mechanisms must validate the resulting intent against the information model, the producer's declared intent-handling capabilities, active policies, and other active intents. Unsupported targets or inconsistent constructs must be rejected, and conflicts must be resolved before the intent reaches the network. The model may assist in formulating the intent, but it does not determine which unvalidated payload is committed to the operational infrastructure.

Two design consequences follow. First, refinement must be traceable: every target should be linked back to the sentence, contract clause, or requirement identifier that justified it; otherwise, the operator cannot defend a fulfilment report to the customer. Second, refinement must be iterative. When a producer reports that an intent cannot be fulfilled, the agent's role is to support renegotiation in the vertical's own language rather than merely relaying an error code. For example, this may involve relaxing a target, narrowing a context, or modifying a priority. This is where the memory models of Section 4.2 become important, since renegotiation only makes sense against a record of what was requested, attempted, and previously agreed.

The 3GPP TS 28.312 allows the generic intent information model to support new scenarios by combining standardised expectation objects and targets, by extending it with vendor-defined elements, or, where necessary, by using vendor-defined expectations. The standardised information model therefore provides the formal vocabulary and validation framework, while AI-assisted refinement provides the semantic bridge from application goals to that vocabulary.

Once validated and committed, the machine-readable intent becomes an operational input to the cognitive network control. In this framing, information, perception, and transport metrics form a structured hierarchy that serves the overarching application goal. Network slices can therefore evolve into dynamic governance envelopes rather than remain static service templates: isolation and security boundaries are still enforced, while behaviour within those boundaries can be dynamically composed to meet the declared objective. The cognitive network control may instantiate specialised scheduling policies, predictive resource-control loops, or compute-placement modifications to fulfil the intent, while supervisory agents arbitrate competing intents, coordinate multi-domain resources, and enforce rollback mechanisms to preserve system stability. The complete chain, therefore, runs from an intent expressed in the vertical's own language, through formal refinement and deterministic validation, to the controlled composition of network behaviour.

4.2. Memory Models for Long-Horizon Network Optimisation

Agent memory refers to an AI agent's ability to store, organise, update, and recall information from past interactions and experiences. Rather than treating every interaction as an isolated task, a memory-enabled agent can build on previous conversations and experiences, retain useful information, and reuse relevant knowledge when making future decisions. Early systems such as Generative Agents demonstrated how stored experiences can be dynamically retrieved and synthesised into higher-level reflections to support subsequent planning and behaviour [18].

Agent memory can be broadly divided into several complementary components. *Working* memory maintains information that is immediately relevant to the agent's current reasoning process, such as active goals, recent observations, retrieved knowledge, and intermediate reasoning results. Long-term memory, in contrast, preserves information across interactions and can be further categorised into *episodic* memory, which stores previous experiences and interaction histories; *semantic* memory, which stores factual or generalised knowledge about the world and the agent itself; and *procedural* memory, which captures reusable behaviours, rules, or skills [19].

Figure 4.2 illustrates a concrete example of how such memory can be managed and utilised within an AI agent. In the Memory-R1 framework, incoming information is first processed to construct and continuously update a memory bank through operations such as adding, updating, or deleting memories; when answering a later query, the agent then selects relevant memories from this bank and uses them to support its reasoning and answer generation.

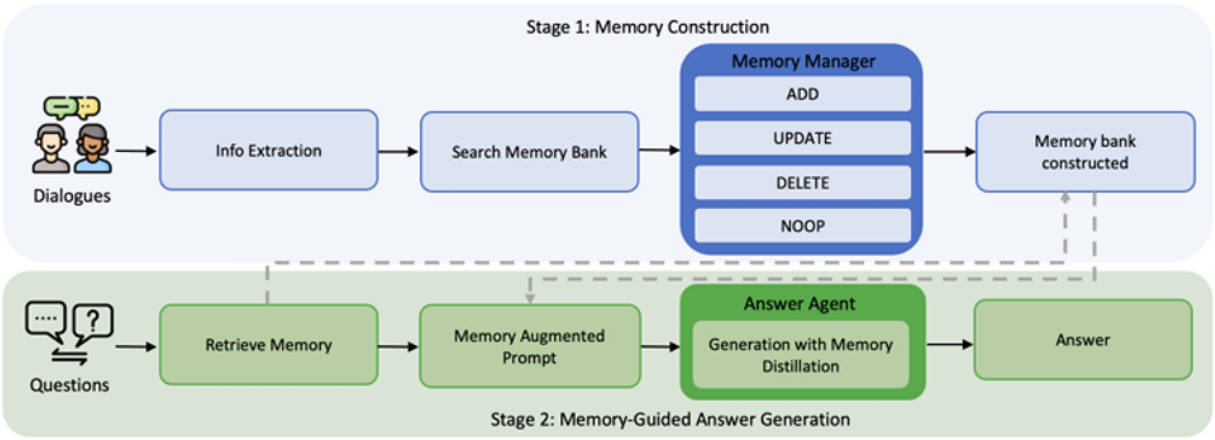

Figure 4.2. Overview of the Memory-R1 framework, illustrating the two main stages of agent memory: memory construction and memory-guided answer generation [20].

Building effective agent memory introduces several important challenges and trade-offs. Storing more information can improve coverage, but also increases computational cost and makes relevant information harder to identify. Aggressive compression improves efficiency but risks discarding useful details. Moreover, memories may become outdated, contradictory, redundant, or incorrect, requiring mechanisms for consolidation, updating, and forgetting. Recent work therefore increasingly treats memory management itself as a decision-making problem: for example, Memory-R1 and Memory-R2 trains agents to actively decide whether information should be added, updated, deleted, or left unchanged, as well as how relevant memories should be selected for answering future queries [20], [21].

Ultimately, agent memory is not simply about storing more data. It is about deciding what should be remembered, how memories should evolve, and when they should be retrieved and used to support reliable, adaptive, and personalised decision-making.

Long-horizon network optimisation

Long-horizon network optimisation is, at its core, a memory problem before it is a reasoning problem. It sits downstream of two other design layers already discussed in this contribution:

the intent that specifies what desired behaviour actually means for a given network (§4.1), and the policy-driven control that bounds which actions are admissible in pursuit of it (§2.4). Memory is what lets an agent hold both steady over a horizon long enough for them to matter. An agent that must keep a network converging on desired behaviour over days, weeks, or months cannot treat every decision as a fresh inference: it needs a persistent record of what the network looked like, what was tried, and what happened, structured so that record can be retrieved cheaply and acted on safely. Agentic AI research outside telecom has converged on a taxonomy of memory types that maps naturally onto this problem and adapting it to mobile communication systems is the purpose of this section.

As highlighted above, four memory types recur across agentic architectures, and each plays a distinct role in a network-management agent (Figure 4.3). Working memory is the agent's active context window, the KPIs, alarms, and in-flight task state it is reasoning over right now. It is bounded by the underlying model's context length and, following the operating-system-style paging view of context management introduced by MemGPT [22], has to be actively managed rather than simply grown; because every token held in it is billed and adds latency on each reasoning step, working memory is also the tightest coupling point between the memory architecture and the cost model developed in §7.1.

Episodic memory is the agent's log of past events and their outcomes: the last time a given congestion pattern occurred, which action was taken, and what the measured result was. The same long-term, scalable event memory that production agent-memory systems such as Mem0 are designed to maintain across sessions [23]. It grounds the agent's reasoning in this specific network's history rather than in generic patterns learned during training, which matters because no two deployed networks behave identically.

Semantic memory holds distilled, structured knowledge about the network itself: topology, node capabilities, active policies, contracted SLAs. This information is typically maintained as a knowledge graph or vector store and updated on a slower cadence than the episodic log, since topology and policy change far less often than traffic conditions; knowledge-graph representations of exactly this kind are already being explored for intent-based network management [24]. Procedural memory, finally, stores reusable playbooks for recurring tasks such as how to rebalance a given slice type, or how to walk back a failed configuration change, so the agent can execute a known good procedure instead of re-deriving a plan from first principles on every occurrence.

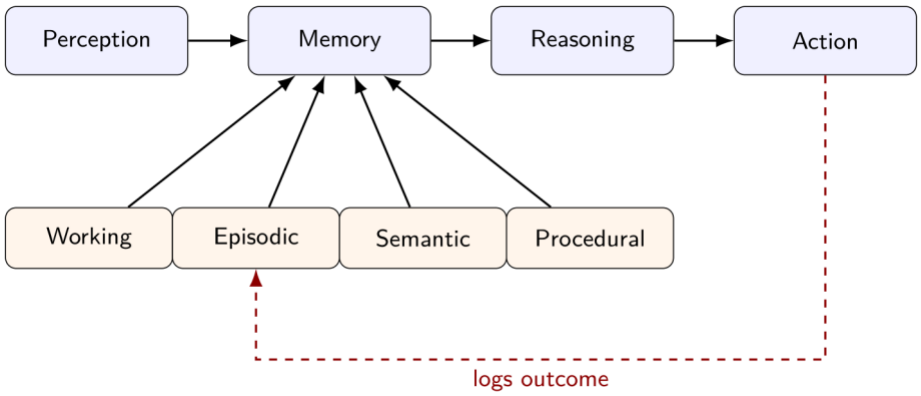

Figure 4.3. Memory types feeding into an agentic pipeline: perception, memory, reasoning, action. Each memory subtype supplies the reasoning stage; the action stage logs its outcome back into episodic memory, closing the loop.

What makes memory a genuinely hard problem in this setting, rather than a solved one borrowed wholesale from general-purpose agents, is the combination of non-stationarity and multiple coexisting timescales. Traffic patterns, topology, hardware inventory, and the RF environment all drift over the course of seasons and years, so a memory architecture needs an explicit consolidation and forgetting policy: recent observations should not be allowed to permanently outweigh older ones if the older pattern is a genuinely recurring seasonal effect, but neither should stale observations from a decommissioned cell site continue to bias current decisions indefinitely. This tension between catastrophic forgetting and stale-memory bias is a long-studied problem in continual learning [25] with no generic solution. It has to be tuned to the specific rate of change of each part of the network. Layered on top of this is the fact that a mobile network runs several control loops at once, at very different timescales: Functions closer to the radio layer such as dynamic radio resource management, load balancing and mobility control operate control loops in the order of 10s of milliseconds, functions around the management and orchestration layer such as policy management and AI/ML model training operate in multi-seconds or more, while traffic engineering, slice reconfiguration, and capacity planning extend that range out to minutes, days, and months (Figure 4.4). Each of these loops needs memory retrieval matched to its own decision horizon, so a single flat memory store - one undifferentiated log queried the same way regardless of the calling loop - does not fit the problem. A near-real-time control loop needs sub-second access to very recent episodic state; a capacity-planning agent needs slow-changing semantic and trend data and can tolerate retrieval latency that would be unacceptable a few layers down the stack.

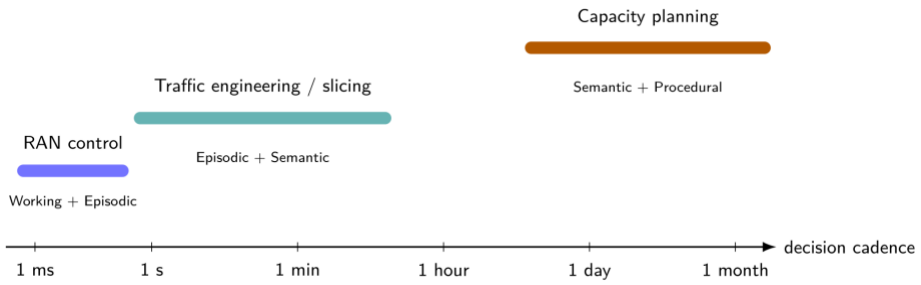

Figure 4.4. Memory access needs three coexisting network control loops, from millisecond-scale RAN control to month-scale capacity planning. Each loop draws on a different combination of memory types.

Memory also functions as a direct substitute for expensive reasoning, which is what ties this section back to the cost and latency constraints discussed elsewhere in this contribution. Retrieving a prior, sufficiently similar situation from episodic or semantic memory - in effect, retrieval-augmented generation applied to the network's own operational history rather than to an external corpus [26] - is markedly cheaper in terms of tokens and latency than asking the agent to re-derive an appropriate action from first principles on every invocation. This argues for treating memory-retrieval quality as a first-order lever on operating cost, not as a convenience layered on top of reasoning. The same argument extends to multi-agent settings: a shared semantic or episodic store, such as a common network knowledge graph accessible to every domain agent, lets agents coordinate without re-communicating their full context to one another on every interaction, which is both a token saving and a straightforward way to keep independently reasoning agents consistent with each other, connecting to the multi-

agent coordination models discussed in §2.3. However, in multi-tenant or multi-slice deployments, shared semantic and episodic stores require strict namespace isolation and cryptographic access boundaries so that one tenant's historical logs, SLA parameters, or incident records cannot leak into another slice's memory retrieval.

None of this is safe by default. An agent acting on outdated topology or policy memory can take an action that was correct for a network state that no longer exists. Therefore, memory needs explicit freshness and versioning, together with validation against live network state before it is allowed to drive an action. This requirement connects directly to the provenance and input-validation guardrails discussed in §5.4. And because compute in a mobile network is distributed across wildly heterogeneous tiers, from constrained on-device or near-edge nodes to cloud infrastructure with comparatively unlimited resources, where a given piece of memory physically lives is itself a design decision, not an afterthought. This decision has to be made jointly with the distributed-intelligence and collaborative-model placement questions raised in §2.2 and §3.2.

Concretely, we expect the memory layer for this class of agent to combine a vector store over embedded incident reports and operational logs - building on log-representation techniques already used for log-based anomaly detection [27]. This combination will serve episodic retrieval. Additionally, a knowledge graph over network topology and policy, serving semantic retrieval; and a small library of versioned playbooks, serving procedural memory. Access will be through a hybrid retrieval layer that routes each query to the store, and the timescale, appropriate to the calling control loop. Getting the consolidation policy, the freshness guarantees, and the placement of these stores right is, in our view, one of the more underrated technical prerequisites for agentic AI to be trusted with long-horizon network optimisation at all.

4.3. Reasoning Under Constraints (SLA, Policy, Cost, Risk)

Agentic decision-making in mobile communication systems is subject to multiple, potentially competing constraints: 1) Service-level agreements (SLAs) define the performance and service requirements that should be satisfied, such as latency, reliability, throughput, or task-specific objectives; 2) express desired operational objectives and rules for network behaviour, such as minimizing energy consumption, balancing resources, prioritizing certain services, or maintaining specific operational conditions; 3) Cost captures the resources required to execute a decision, including communication, computing, and energy consumption; 4) Risk reflects the uncertainty and potential consequences associated with a decision, including the possibility that an action may violate service requirements or lead to undesirable network behaviour. Effective agentic decision-making therefore requires to jointly consider these constraints and resolve their trade-offs when selecting an action.

Agentic decision-making can be supported by mathematical and algorithmic optimization methods that explicitly account for service constraints and resource limitations. The work in [28] combines LLM-based agents with a multi-objective optimizer to generate a set of Pareto-optimal SLA offers. Given global network resource budgets, stakeholder SLA constraints, and the number of participating stakeholders, the optimizer generates a Pareto set of non-dominated, SLA-feasible offers, i.e. a set of multiple optimal choices in which no other solution dominates them in presence of multiple objectives. The agents then negotiate over this set rather than over unconstrained configurations, while the agreed result is subsequently translated into network policies for enforcement

For telecom agents, this highlights the need for reasoning capabilities that can account for multiple, potentially competing constraints and objectives when assessing possible actions. Rather than considering each requirement in isolation, agents need to reason about their interactions and the potential trade-offs between them, taking into account the current network context and objectives.

4.4. Agent Communication and Semantic Exchange Mechanisms

Multi-agent coordination requires more than task decomposition and role assignment. Agents also need interoperable mechanisms to discover one another, expose capabilities, exchange requests, context and results, negotiate workflows, and return verifiable artefacts. In mobile communication systems, these mechanisms must also respect domain boundaries, authority levels, latency constraints, identity requirements, and auditability. This section examines agent communication protocols and semantic exchange mechanisms as technical enablers of distributed agentic operation.

Multi-agent systems require standardised communication interfaces to exchange requests, context, capabilities, and results. Several emerging protocols aim at enabling interoperable agent ecosystems [29], [30]. In the context of mobile communication systems, these protocols should not be treated as generic software interfaces only, but as mechanisms that determine where trust is established, how authority is delegated, how negotiations are traced, and which timescales of network operation they can support.

Three protocol families are currently emerging in this space, and Figure 4.5 compares them schematically: the Agent-to-Agent Protocol (A2A), the Agent Communication Protocol (ACP), and the Agent Network Protocol (ANP). They differ in the interaction pattern they assume and, above all, in the scope over which interoperability is sought. A2A is point-to-point and connects two agents that have discovered each other through explicit capability descriptions. ACP places a shared messaging bus between many agents, so that several of them can take part in the same negotiation or workflow. ANP extends the reach to the open Internet, where agents belonging to different organisations must first establish identity and trust before they can cooperate at all.

The three protocols correspond to progressively wider deployment scopes: a pair of collaborating agents, an organisation or enterprise ecosystem, and an open federation of platforms. Since each of them resolves a different problem, and each carries distinct advantages, limitations, and deployment implications for mobile networks, the three subsections that follow examine them individually before Table 4.3 summarises the comparison.

Agent-to-Agent Protocol (A2A)

The Agent-to-Agent (A2A) protocol [31] focuses on direct peer-to-peer interactions among autonomous agents. Under this approach, agents expose their capabilities through standardised descriptions, allowing other agents to discover them and dynamically invoke the services they provide.

In operational terms, an A2A exchange unfolds in the four steps shown in the left panel of Figure 4.5. First, each agent publishes a machine-readable description of what it can do, commonly referred to as an Agent Card, which other agents retrieve during registration and discovery. Second, a requesting agent delegates a task by issuing a structured request that carries a task

identifier, the role of the sender, the method invoked, and its parameters, typically transported over HTTPS using JSON-RPC 2.0. Third, the executing agent returns progress updates while the task is still running, so that long-running operations do not block the requester. Fourth, results are returned as explicit artefacts, such as reports, measurement sets, or configuration files, rather than as free-form text. A relevant property of this model is that execution remains opaque: the delegating agent observes the advertised capabilities and the returned artefacts, but not the internal tools, models, or state of the invoked agent. This separation is what allows two agents built by different vendors, and operated under different administrative authorities, to cooperate without exposing their internal logic to each other.

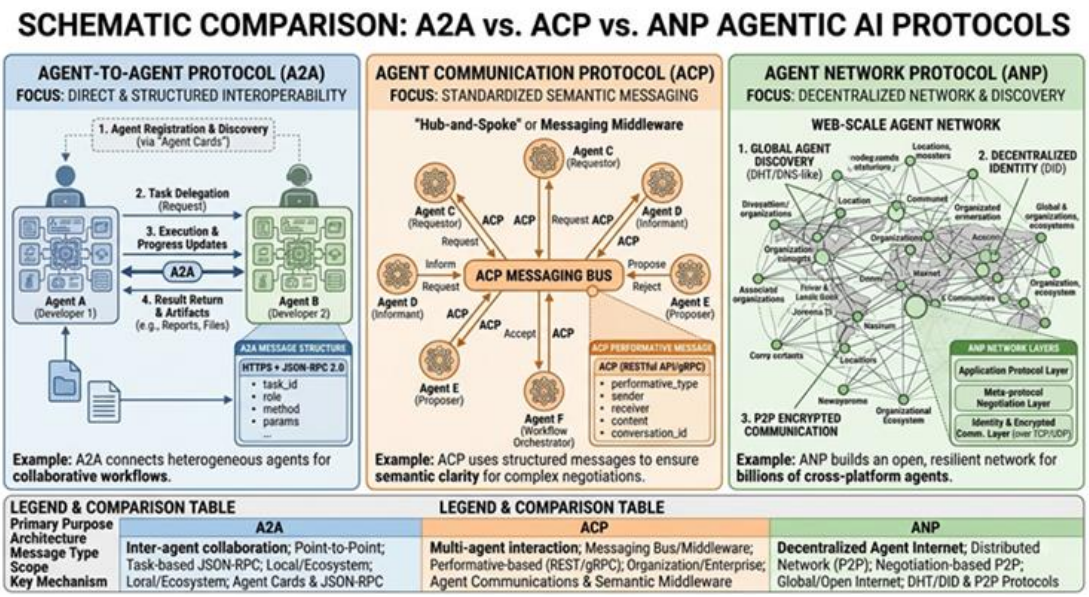

Figure 4.5. Schematic comparison of the A2A, ACP, and ANP agent communication protocols

A2A can facilitate interoperability across vendors and platforms and is well-suited to distributed execution. Its capability-based interaction model allows an agent to delegate a task to another agent based on the functions, tools, or expertise the latter makes available. This characteristic is particularly valuable for cross-domain orchestration, where a single optimisation process may involve multiple administrative or technological areas.

However, A2A interactions can cause considerable overhead due to agent discovery and capability negotiation. Direct collaboration between autonomous entities requires strict security measures, as identity, permissions, integrity, and trust need to be verified prior to sharing information or control. Keeping a consistent global network state is also challenging when multiple agents make decisions simultaneously. These issues become more prominent as the number of participating agents increases.

A2A appears particularly attractive for inter-domain mobile network orchestration. For example, it could enable collaboration among intelligence functions associated with the Radio Access Network (RAN), transport network, core network, edge platform, and service-management domain, while allowing each function to retain a degree of autonomy. A concrete example clarifies why a dedicated protocol is needed at this level. Consider a violation of the end-to-end latency budget of an ultra-reliable slice. A service-management agent detects the violation but

has no visibility of its cause, so it delegates one diagnostic task to a RAN-domain agent and another to a transport-domain agent, both identified through their Agent Cards. Each of them investigates within its own domain, using tools and data that the requester neither sees nor is authorised to access, and returns an artefact containing its findings. The shared task identifier allows the service-management agent to correlate the two answers and to attribute the fault before requesting a corrective action. Authority over each domain is never transferred; only the task and its result are.

The same example also exposes a structural limit. Because every collaboration is a direct relationship between two agents, the number of relationships to be established, secured, and maintained grows approximately with the square of the number of participating agents. A2A consequently fits the hierarchical structure discussed later in this section, in which a domain agent delegates to a bounded set of local agents and coordinates with a few peer domain agents, but it becomes unwieldy when many agents have to contribute to the same decision. This is precisely the situation addressed by the next protocol.

Agent Communication Protocol (ACP)

The Agent Communication Protocol (ACP) extends agent interactions by introducing structured messaging primitives and asynchronous exchanges. ACP typically supports richer interactions, including streaming responses, multimodal content, and workflow negotiation.

The structural difference with respect to A2A is visible in the central panel of Figure 4.5. Agents do not connect to one another directly, but attach to a shared messaging bus, or messaging middleware, which relays their exchanges. Hence, a sending agent does not need to know the address of the recipient, nor whether the recipient is currently active, since the bus decouples the participants both in space and in time. What travels on the bus is a structured message in which the communicative intent is declared explicitly through a performative type, such as request, inform, propose, accept, or reject, alongside the sender, the receiver, the content, and a conversation identifier. The conversation identifier is what turns a sequence of otherwise independent messages into a single traceable negotiation, possibly spanning several agents and several rounds. Agents assume well-defined roles within such a conversation, acting as requesters, informants, proposers, or workflow orchestrators, and these roles may change from one conversation to the next.

The main advantages include support for asynchronous communication, richer message semantics, improved support for heterogeneous agents, and efficient orchestration of long-running tasks. However, the disadvantages include increased protocol complexity, higher message overhead, and additional implementation requirements.

The practical value of declaring the performative type is that a receiving agent can decide how to handle a message before interpreting its payload: a proposal can be queued for evaluation, a rejection can close a negotiation branch, and an informative update can be logged without triggering any action at all. This makes the behaviour of the system inspectable, because the complete history of a negotiation can be reconstructed from the bus, and it reduces the number of relationships to be maintained from quadratic to linear in the number of agents, since each agent only has to be connected to the middleware.

These benefits are not free. The bus concentrates both traffic and trust: it can become a performance bottleneck and a single point of failure, and it implicitly assumes that all participants operate within one administrative domain that owns and governs the middleware. Every exchange also incurs the latency of one additional hop. Finally, explicit performatives guarantee that agents agree on the intent of a message, not on its meaning. If two domains define congestion or degradation differently, a perfectly well-formed proposal can still be

misinterpreted. Thus, a shared vocabulary for network state and actions remains a prerequisite and represents an open item for standardisation.

ACP may become highly relevant for orchestration scenarios involving edge-cloud hierarchies, where decisions are generated over multiple timescales and exchanged asynchronously.

In mobile networks, the exchanges that fit this model are those in which several parties must converge on a decision rather than execute a delegated instruction. Resource negotiation for network slicing is the clearest case: slice agents issue proposals for additional capacity, an infrastructure agent accepts or rejects them according to admission and priority rules, and the entire exchange remains bound to a single conversation identifier and is auditable afterwards. Energy-aware operation follows the same pattern, since decisions on sleep modes and traffic steering involve radio, transport, and service agents whose objectives conflict and whose reaction times differ by orders of magnitude. The asynchronous nature of the protocol makes this workable, as an agent reasoning over hourly traffic forecasts and an agent reacting within seconds can participate in the same conversation without blocking each other. The same messaging substrate can also carry the conflict-resolution and negotiation mechanisms discussed in Section 5.6.

Agent Network Protocol (ANP)

The Agent Network Protocol (ANP) is designed for open, decentralised ecosystems where agents may dynamically join and leave the system. ANP incorporates mechanisms for discovery, trust establishment, identity management, and secure collaboration.

As shown in Figure 4.5, ANP is organised into three layers, each addressing a question that neither A2A nor ACP addresses at web scale. The lowest layer handles identity and encrypted communication directly over TCP or UDP, and gives every agent a Decentralised IDentifier (DID), that is, a credential the agent can prove cryptographically without relying on a registry operated by any single party. Above it, a meta-protocol negotiation layer allows two agents that do not share a common interface to agree on which protocol and data schema to use before any application exchange begins. The application protocol layer then carries the interaction itself. Discovery is decentralised in the same spirit, with agents located through distributed, DNS-like directories rather than through a central catalogue, which allows the resulting network to span organisations, ecosystems, and platforms that have no prior relationship.

Advantages include decentralised operation, support for large ecosystems, secure identity management, and flexible agent discovery. However, ANP also has drawbacks, including increased signalling overhead, more complex trust management, longer coordination delays, and the need for additional security infrastructure.

The decisive advantage of this design is in governance rather than in engineering. Because identity is decentralised, no participant has to operate another's registry or issue credentials on its behalf, which is the condition that makes collaboration between competing organisations administratively feasible in the first place. The meta-protocol layer adds tolerance for heterogeneity, so that agents developed independently and at different times can still establish a working interface. The absence of a central component likewise removes the single point of failure inherent in a messaging bus.

The corresponding cost is concentrated at the beginning of every new interaction. Discovering a counterpart, verifying its identifier, and negotiating a protocol constitute a handshake that precedes any useful exchange, and its duration makes ANP unsuitable for short control loops. Managing the lifecycle of decentralised identifiers, and their revocation in particular, is operationally demanding, since a compromised agent must be excluded without a central authority able to act unilaterally. Dynamic protocol negotiation is moreover an attack surface in

itself, because an adversary able to influence the negotiation may steer two legitimate agents towards a weaker interface. These aspects connect the present discussion directly to the zero-trust architecture and the agent identity and attestation requirements addressed in Chapter 5.

ANP may be particularly useful for future federated mobile networks involving multiple operators, vertical industries, edge providers, and private networks.

Scenarios that justify such infrastructure are those that cross operator boundaries. Roaming agreements, network federation, and neutral-host deployments all involve agents from distinct administrative and legal entities, for which mutual authentication cannot rely on a shared credential authority. The same applies to verticals that temporarily attach a private network to a public one, to edge providers engaged for a single workload, and, more generally, to capability marketplaces in which an operator delegates a task to a third-party agent it has never interacted with before. A further benefit specific to this class of deployments is accountability: since every action is bound to a verifiable identifier, the resulting audit trail can support service-level agreement verification and dispute resolution between parties that do not trust by default. Consistent with the handshake cost noted above, these decisions are made on timescales of minutes to hours, not within the fast control loops of the radio interface.

From a deployment perspective, the three protocols address different boundaries rather than competing for the same one. A realistic agentic mobile network is likely to combine them as needed: ANP for interactions that cross operator or organisational boundaries; ACP within a single operator's perimeter when several agents must converge on a common decision; and A2A wherever one agent delegates a bounded task to a specific peer. The selection criterion is thus not the expressiveness of the protocol, but the scope of trust and the reaction time the interaction demands.

Overall, the three approaches can be viewed as complementary rather than mutually exclusive, as summarised in Figure 4.5 and in the following table.

Table 4.3. Main agent communication schemes

Protocol	Primary focus	Strengths	Limitations
A2A	Peer-to-peer task delegation	Simplicity and interoperability	Limited coordination
ACP	Structured collaboration workflows	Rich messaging and asynchronous execution	Higher complexity
ANP	Open decentralised ecosystems	Discovery and trust management	Significant signalling overhead

Beyond protocol-level interaction, agentic mobile networks also require efficient semantic information exchange. In this context, moving beyond conventional bit-level transmission towards meaning-centric communication enables efficient cross-modal information exchange. Token Communication (Tok-Com) supports this shift. However, deploying computation-intensive diffusion decoders across resource-constrained agents introduces a fundamental trade-off among reconstruction fidelity, energy consumption, and information

freshness, the latter quantified by the Age of Information (AoI). This trade-off is particularly critical in distributed robotic systems, where timely and semantically accurate information is essential for effective perception, coordination, and decision-making.

Agentic AI can revolutionise how Tok-Com will be used in 6G: This will enable a shift from Tok-Com as a point-to-point cross-modal mechanism for reconstructing images to Tok-Com as the communication substrate for distributed agentic AI. In that setting, semantic tokens are not merely used to reconstruct an image. They become shared, actionable state representations that multiple agents can use to perceive, reason, coordinate, and act. Tok-Com gives distributed agents a common semantic language. The fleet can then exchange tokens rather than raw sensor data. This is particularly important for agentic AI, because agents generally do not need to reproduce another agent's entire sensory observation. They need only the information necessary to make a better decision [32].

4.5. Digital Twin Driven Verification and Sandbox Technology

The execution of autonomous decision-making loops introduces a degree of operational uncertainty that standard telecommunications networks may not natively tolerate. To mitigate this, the cognitive network control integrates digital-twin operational models to serve as a real-time validation sandbox. Before an AI agent executes a synthesised control loop or alters a routing policy on the live physical infrastructure, the proposed behaviour is mirrored and executed within the digital twin. This virtual sandbox evaluates the policy's impact against historical telemetry, checking for potential multi-agent race conditions, safety envelope violations, or unintended SLA degradation, enabling safe validation and instant rollback.

Edge-Aware and Data-Local Digital Twin Sandboxing

Digital twin driven verification should not only check network-level performance indicators. For mobile agentic AI, the sandbox should also test how an AI model or agentic function behaves before it is exposed to the live network. This is especially important when intelligence is distributed across devices, edge nodes, RAN functions and cloud platforms. A model that performs well in a central test setting may behave differently under local traffic, mobility, radio, energy or device constraints.

A practical role of the sandbox is therefore to support pre-deployment validation under realistic edge conditions. This may include testing behaviour under changing traffic load, mobility patterns, radio conditions, device heterogeneity, partial connectivity and non-IID local data. The aim is not to build a perfect copy of the live network, but to expose major risks early, such as unstable recommendations, excessive signalling, poor performance under local data shift, or unexpected resource pressure at the edge.

For federated or privacy-preserving AI models, the sandbox should also support data-local validation. In many mobile scenarios, raw data cannot be freely centralised. Local domains may need to test models, updates or policies using local data, while sharing only validation summaries, performance indicators, error patterns or other non-sensitive outputs. This allows operators to assess whether a model is useful across different sites or edge domains without assuming full raw-data visibility.

The sandbox can also support staged deployment. An AI function may first be tested offline, then evaluated in an emulated or shadow mode, and only later activated in a limited operational scope. This staged approach can help operators compare expected and observed behaviour, monitor drift, and decide whether an update should be accepted, delayed, revised or reviewed further [6].

This complements the broader discussion on autonomous decision-making and safety boundaries. The focus here is not to define the agent's decision logic, but to provide a practical validation environment where distributed AI models can be tested under edge, mobility, privacy and lifecycle constraints before they are trusted in live mobile networks.

4.6. Telco-grade Behaviour in Telecom Infrastructure: Deterministic Operation under Probabilistic Intelligence

A key challenge in AI-native 6G is balancing AI's probabilistic decisions with the need for predictable telecom operations. Mobile networks already manage uncertainty from variable radio signals, traffic shifts, user movement, and interference through measurable targets like error rates and latency. With foundation and agentic AI, uncertainty changes: outputs can have larger, unpredictable outcome spaces and lack clear distributions for monitoring. The challenge isn't removing probabilistic behaviour but preventing it from unchecked influence, especially in critical functions like authentication, mobility, emergency access, charging, compliance, session continuity, slice isolation, and service assurance. In such areas, behaviour must be predictable, auditable, policy-compliant, and bounded by standard procedures. As automation expands, these properties grow more essential.

The core principle is to keep probabilistic AI within deterministic control boundaries. AI modules assess network conditions, forecast, optimise, rank, break down intents, or suggest actions. However, they should not directly change the network. Instead, AI proposals must undergo deterministic validation and actuation, ensuring probabilistic reasoning aids decision-making while execution remains governed by verifiable controls.

This separation is vital for telco-grade behaviour. An AI model might suggest adding resources to a network slice or moving workloads closer to users, but these conclusions must first be converted into structured, policy-compliant operations. The network should only execute authorised, validated actions checked against constraints, mapped to interfaces, and, if needed, paired with rollback procedures. The goal is to ensure the deployed behaviour is controlled and the decision process can be reconstructed, not necessarily to reproduce the reasoning itself.

A practical architecture separates four key responsibility layers. The AI reasoning layer generates predictions, recommendations, intent breakdowns, or possible control strategies. The policy and governance layer decides if an action is allowed and sets conditions, such as domain, slice, authority level, resource scope, and time frame. The deterministic validation layer ensures proposals meet operational constraints like resource and rate limits, SLA priorities, stability margins, security policies, compliance, and rollback conditions. The execution layer then converts approved proposals into standardised network procedures and logs the operational changes. This design keeps probabilistic reasoning apart from direct actuation, enabling AI to assist with interpretation, prediction, and optimisation where valuable.

Such an architecture also provides a basis for bounded autonomy. The authority of an AI agent should be defined in terms of the scope and consequences of the actions it is allowed to perform. Different agents may therefore operate with different levels of autonomy depending on the domain, service class, resource range, geographic area, time horizon, and the criticality of the affected function. A fast RAN-level agent, for example, may optimise scheduling or radio parameters within a tightly constrained envelope, whereas a higher-level orchestration agent may adjust compute placement or slice resource allocations over longer time scales. When several agents interact, a deterministic governance function is required to arbitrate competing objectives and prevent contradictory control loops. Actions can then be sequenced, prioritised, modified, or rejected according to explicit policies rather than being resolved through unconstrained agent negotiation.

The permissible autonomy level depends on two key factors: compliance impact and reversibility. Some network requirements are statistical, with an accepted failure rate, while others are invariants that must be safeguarded during operations, such as emergency communications or spectrum constraints. Actions affecting invariants need rigorous validation, unlike performance optimisations. Reversibility is crucial: actions with quick rollback options and known effects can

tolerate more autonomy than irreversible ones with large impacts. Therefore, autonomy isn't uniform; it varies based on task type and risk.

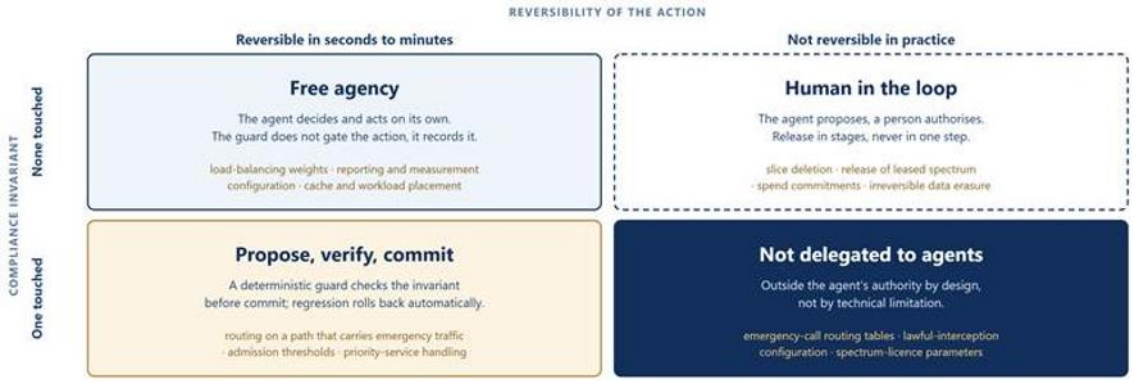

Figure 4.6. Governance regimes for agent actions based on reversibility and compliance impact.

As depicted in Figure 4.6, the degree of autonomy granted to an agent can be determined by considering two independent properties of the proposed action: its reversibility and its impact on compliance invariants. Reversibility indicates whether an action can be undone and, if so, how rapidly, whereas a compliance invariant is a requirement that must hold in every instance and therefore has no acceptable failure budget. Combining these dimensions defines four governance regimes with progressively different levels of agent authority. Importantly, restricting autonomy for irreversible, compliance-critical actions reflects a decision about where operational authority should reside, rather than a limitation on the agent's technical capability.

Digital twins and simulation environments support this model by evaluating strategies before live network deployment, assessing impacts on service continuity, isolation, resources, energy, stability, and compliance. However, digital twins aren't perfect predictors; they reduce uncertainty and identify potential issues, not eliminate them. Validation should blend simulation, explicit rules, monitoring, conservative operating limits, and fallback procedures.

Observability is essential because telco-grade behaviour needs verification before and after actions. In AI operations, networks should store enough info to trace decisions and their outcomes, including trigger events, network state, model inputs, versions, suggested actions, validation results, final actions, and observed effects. Differentiating deterministic execution from reconstructability is key: probabilistic reasoning may not produce identical results on reruns, especially after updates. Auditability relies on capturing the decision process at the moment, not recreating it later. In telco AI, explainability relates to the entire control process, not just the model.

Intent-driven operation should follow the same principle. Natural-language interaction can be useful for expressing high-level goals, but unconstrained natural language should not serve as a direct control interface to critical network functions. AI-generated or user-provided intents should be translated into structured, machine-verifiable representations that specify objectives, measurable targets, constraints, priorities, validity periods, authority levels, and rollback conditions. The AI can interpret and refine an intent probabilistically, while the network executes only the resulting validated control operations.

Finally, AI-native telecom systems require explicit fallback and degradation modes. If model confidence is insufficient, input data quality deteriorates, several agents disagree, validation fails, or the impact of a proposed action cannot be bounded with sufficient confidence, the system should revert to established deterministic procedures or restrict operation to a conservative parameter envelope. For critical functions, the appropriate outcome may be to reject autonomous action and require human approval. Autonomy should therefore be conditional and task-specific

rather than absolute. This aligns with the broader telecom approach to management autonomy, in which different stages of observation, analysis, decision, execution, and intent handling can involve different combinations of human and automated control.

The core design philosophy is that AI offers adaptability, prediction, optimisation, and composability, while telecom control systems ensure bounded behaviour, accountability, compliance, and safety. The goal is not to replace traditional telecom processes with probabilistic agents or turn a stochastic communication system into a deterministic one. Instead, the aim is to prevent AI's uncertain decisions from impacting critical control paths without oversight. AI can reason probabilistically and consider multiple options, but any network changes must follow predefined policies, be validated, observable, and reversible if needed. This separation allows AI-driven 6G networks to be more autonomous while ensuring reliability and safety.

4.7. Code Generation's Implication on Future Service Provisioning

Recent research suggest that generative AI will play an increasingly important role in future mobile networks, enabling higher levels of autonomy, flexibility, and service customisation. In particular, emerging research has demonstrated that AI agents can translate text-based service requests into executable code, allowing networks to generate customised processing functions on demand. These functions can inspect and interpret user-plane protocol data units and perform application-specific actions, thereby enabling a new class of dynamic, AI-driven connectivity services. A representative example of this direction is provided in [33], which investigates the accuracy and feasibility of generating such processing blocks using different model configurations, prompt designs, and code templates. Their findings indicate that, under suitable conditions, AI agents can reliably produce network-specific function blocks with the desired behaviour. This line of work highlights the potential of generative AI to introduce on-demand capability creation in 6G networks and beyond.

4.8. Agent Deployment, Resource Usage, and Energy Consumption

To ensure success for real-time services that rely on strict end-to-end deadlines across communication and computation, the system must carefully coordinate its resources. Technologies such as in-network computing (INC), split-AI workloads, and advanced MIMO systems should not be viewed as separate or standalone features. Instead, their effectiveness increases when they are managed as a unified, integrated system.

By managing these technologies through a unified framework, the network balances the immediate computational needs of localised AI agents with the energy footprints and transport constraints of the infrastructure. This co-design ensures that, for instance, initiating a localised, compute-intensive task or a highly directive radio beam does not cause systemic energy inefficiencies or violate local environmental or cost constraints.

The deployment of Agentic AI across distributed computing environments introduces new opportunities for intelligent resource management. Autonomous agents use complex AI model architectures that exploit real-time contextual information to make adaptive decisions to dynamically orchestrate computational and communication resources across heterogeneous devices, edge and cloud infrastructures as indicated earlier. This is particularly important in dynamic environments, where resource availability, network conditions, workload intensity, energy constraints, and service requirements may significantly vary over time.

However, the increased autonomy that Agentic AI introduces may also result in significant operational costs. A critical challenge of major importance is the trade-off between the intelligence provided by an agent and the overhead required for its operation. Specifically, agentic AI systems rely on continuous monitoring, planning, reasoning, communication, and coordination among multiple agents and infrastructure components. These processes may require significant energy, computational and communication resources, particularly when complex AI models are deployed or when agents exchange significant amounts of contextual information. On the one hand, advanced reasoning capabilities may improve the quality of resource management decisions. On the other hand, the energy and resource costs required to reach such decisions may overcome the resulting efficiency. As a result, agent operation should be adapted according to the complexity of the task and the conditions of the underlying environment.

Toward this direction, resource-aware Agentic AI deployment requires strategic design. Selective agent activation, adaptive reasoning depth, knowledge distillation, model quantisation, early exiting and layer skipping could reduce the introduced overhead [34], [35]. For example, tasks of limited complexity could be handled locally by smaller AI models or lightweight agents, avoiding unnecessary transmissions and remote computations. In contrast, tasks requiring advanced reasoning or access to broader contextual information could be assigned to more capable agents or computationally powerful edge and cloud infrastructures. Such decisions should occur only when the expected improvement in decision quality justifies the additional computational, communication, performance, and energy costs.

Agents could also support energy-efficient resource management by considering battery levels, energy availability, grid power dependence, carbon intensity, and infrastructure utilisation during the decision-making process. Based on this information, agents may delay non-critical tasks, migrate computation toward more energy-efficient infrastructure, or reduce the complexity of the AI models under energy constraints. In addition, the availability of green energy could drive the integration of renewable energy sources through energy-aware placement decisions and resource allocation.

Based on the above, autonomous resource management should be performed within operational constraints. Agents should respect predefined limits related to energy consumption, QoS, AI model performance, and privacy and security constraints. Their decisions should also be observable, explainable, and reversible when necessary. Without such measures, agentic decisions may result in system inefficiencies, excessive inter-agent communication, or unfair resource allocation across devices and services. In addition, multiple agents independently optimising their own objectives could lead to conflicting decisions and unstable management strategies. These are further elaborated in other sections. It becomes apparent that resource- and energy-aware agent deployment should be strongly considered when designing future Agentic AI infrastructures. The objective should not be limited to deploying increasingly capable agents but should also ensure that the performance improvements justify the costs. Agentic AI should therefore contribute not only to intelligent and autonomous operation, but also to the development of sustainable, efficient, and reliable distributed infrastructures.

4.9. Agentic AI & the Physical Layer — Supervisory, Self-Improving Configuration

Agentic AI can extend toward the physical layer at the level of supervisory configuration, adapting the semi-static configuration envelope within which deterministic PHY/MAC mechanisms continue to operate. Instead of relying on isolated, fixed physical-layer heuristic calibrations, this system uses supervisory, self-improving control loops operated by AI agents. These agents continuously monitor complex RF environments and adapt configurations such as CSI reporting, codebook restrictions, power-control parameters, bandwidth parts, numerology, and NTN-related configuration parameters. By viewing the physical layer as an adaptable, software-controlled

system guided by AI agents, the network maintains link stability, predictive coverage, and control margins even in environments with transient, fast-moving nodes or highly variable conditions.

4.9.1. Two timescales and a separation principle

Agentic operation, as described in this paper, denotes intent-driven, closed-loop control across the Radio Access Network (RAN), Core, Transport, and Operations, Administration, and Maintenance (OAM) at timescales of seconds and above, while the physical layer (PHY) operates several orders of magnitude below. In 5G New Radio (NR), a slot lasts $1\text{ms}/2^{\mu}$ for numerology μ, and Hybrid Automatic Repeat Request (HARQ) feedback and uplink grants are scheduled a few slots ahead [36]. At the edge and close to the radio layer, the one-way transport budget is on the order of milliseconds, and at the fronthaul on the order of 100 μs. Inference with a Large Language Model (LLM) requires tens of milliseconds to seconds, so the separation is of three to six orders of magnitude. Two further properties reinforce it. The PHY datapath is fixed-function: deployed systems expose no standardised, certified path for injecting agent logic at runtime. The radio is, moreover, type-approved, so certified transmit behaviour cannot depend on a probabilistic component.

The PHY nonetheless relies on a large semi-static configuration state, comprising the Radio Resource Control (RRC) parameters that govern channel-state reporting, codebooks, power control, HARQ behaviour, bandwidth parts and numerology. These are dimensioned at planning time and rarely revisited on a per-cell basis as the propagation environment evolves. When a building rises in the served area, shadowing and the angular distribution of scatterers change permanently, and the channel statistics, the codebook, the power targets and the HARQ operating point drift together, all four being coupled through the same propagation. The configuration degrades without raising any alarm. This state, rather than the runtime loop, is where agentic AI reaches the PHY.

As depicted in Figure 4.7, the architecture rests on a single design rule: the agentic layer acts as a slow-loop supervisor that shapes the configuration envelope within which the fast loop operates, while the fast loop, comprising the per-slot signal-processing datapath and the deterministic Medium Access Control (MAC) scheduler that drives it, remains classical, hardware-realised and untouched by the agent (Figure 4.7). This separation naturally leads to the question of which PHY parameters are suitable for agentic supervision: The agent sets the link-adaptation policy and the target Block Error Rate (BLER), not the per-slot Modulation and Coding Scheme (MCS); the active codebook and its restrictions, not the per-slot beam; the open-loop power-control parameters, not the per-slot power command. Envelope and fast loop remain coupled because an over-restricted envelope removes degrees of freedom the scheduler relies on, so the enforcement layer is required for stability rather than adopted as a precaution.

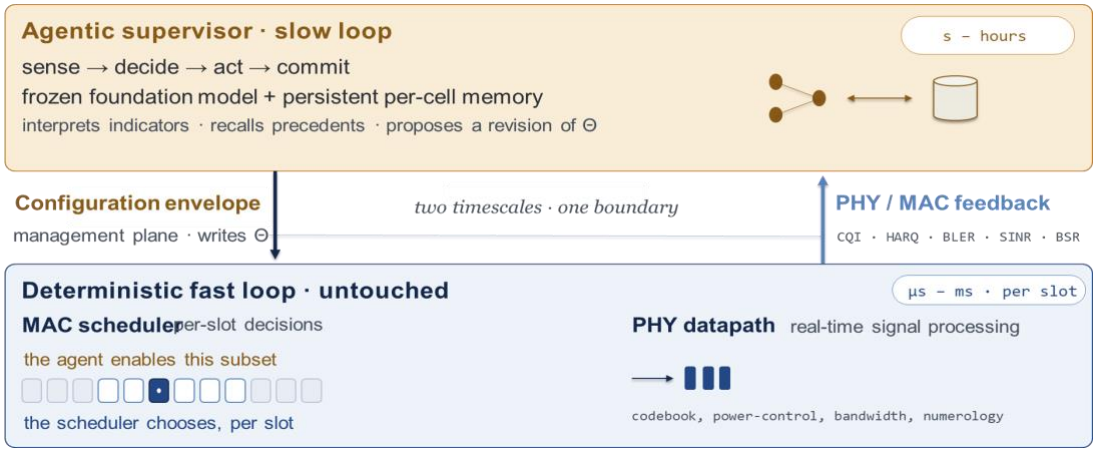

Figure 4.7. The two-timescale separation.

4.9.2. The configuration surface and the admissibility criterion

Each PHY-coupled resource admits a decomposition into a semi-static, slowly varying configured envelope and a runtime control part that tracks the channel and belongs to the scheduler and the datapath. The remit of the agent is the admissible column, taken as a single coupled set, which distinguishes the present approach from Self-Organising Network (SON) functions, each engineered for a single objective with hand-crafted feedback and no memory across episodes: an environmental change that invalidates one parameter invalidates its neighbours as well. Within the Channel State Information (CSI) row alone, codebook-family choices yield throughput differences of over 30%, and Rel-16 feedback compression reduces overhead by about 30% while maintaining equal performance [37].

Admissibility rests on a timescale condition: a resource is agentically controllable only if the agent's reaction time, sense–decide–act–commit plus reconfiguration, stays well below the stationarity time of the controlled quantity. CSI reporting reconfigures in milliseconds, and an agent cycle spans seconds to minutes, against an angular occupancy stable over hours to days; small-scale fading decorrelates in milliseconds and fails the condition by four orders of magnitude, remaining with the fast loop. The condition is necessary but not sufficient, which is one reason the guard also bounds reconfiguration cadence.

4.9.3. Telco-grade enforcement and standardisation implications

The output of the agent is probabilistic, whereas the network is certified, and the two are never connected directly. Every proposed reconfiguration passes a deterministic guard before commit. The guard performs admission control against regulatory limits, hardware capability and RRC validity; it limits reconfiguration cadence, which is the enforcement counterpart; it monitors the live network for regression and falls back automatically to the last-known-good configuration; and it may, prospectively, verify a proposal against a network digital twin (Section 4.5) before commit, the fallback path being independent of the availability of such a twin.

Per-cell agency raises a coordination question that the guard also settles. Neighbouring cells are coupled, a reshaped codebook modifies the interference footprint of its neighbours, and independent learning loops acting on coupled objectives may oscillate, which is the classical difficulty of distributed SON. Inter-cell conflicts are accordingly resolved deterministically at the enforcement tier rather than by negotiation among agents. This complements the multi-agent coordination models: negotiation suits separable objectives and timescales that permit deliberation, whereas deterministic arbitration suits physically coupled resources operating under certification constraints. The guard is also the policy enforcement point at which the zero-trust posture attaches to the PHY, its decision log serving as an audit trail. One asset absent from current zero-trust catalogues requires protection, namely the accumulated memory: crafted indicator patterns may poison consolidated experience while each individual commit remains inside the envelope admitted by the guard.

Three recommendations follow. The PHY and the air interface should be included in the scope of agentic 6G networks at the configuration level, under the two-timescale separation principle. The admissibility criterion should be adopted as the test of what agentic AI may control near the PHY, and the configuration admissibility descriptor should be carried into the network resource models of the management frameworks. Agent-observable measurement, probe windows included, should be treated as a standardisation object in its own right, since observation is the half of the loop that current specifications leave implicit. Open research items are consolidated in Section 8.

5. Trustworthiness and Governance

5.1. Self-improving supervision with a frozen model

Model-level improvements change weights and reopen certification. Agent-level improvement, by contrast, keeps the model frozen while accumulating capability around it through memory of past episodes, validated experience, and supervised reflection. In this approach, the agent does not generate arbitrary PHY parameters. It selects among a small set of validated configuration candidates, which are checked by a deterministic guard before deployment. The outcome is then consolidated into per-cell memory, allowing future decisions to benefit from previous experience without modifying the certified model artefact. Physical-layer AI models therefore enter the architecture as governed tools rather than replacements for physical-layer processing: the agent selects, configures, validates, monitors, and retires them, while runtime PHY/MAC operation remains deterministic and certification-compatible.

The loop is shown in Figure 5.1. A deterministic front-end reduces the indicator streams to features; the frozen model selects from a small, validated candidate set of configurations; the guard verifies and commits; and the outcome is consolidated into memory. The model neither ingests raw numeric streams nor synthesises free-form parameter values. One difficulty is created by the architecture itself: the agent configures its own sensors, so that a restriction it imposes censors the evidence that would justify lifting it, a pruned beam producing no further evidence that the obstruction has been removed. The mitigation is itself a configuration act, namely periodic probe windows, and belongs to any standardisation of agent-observable measurement.

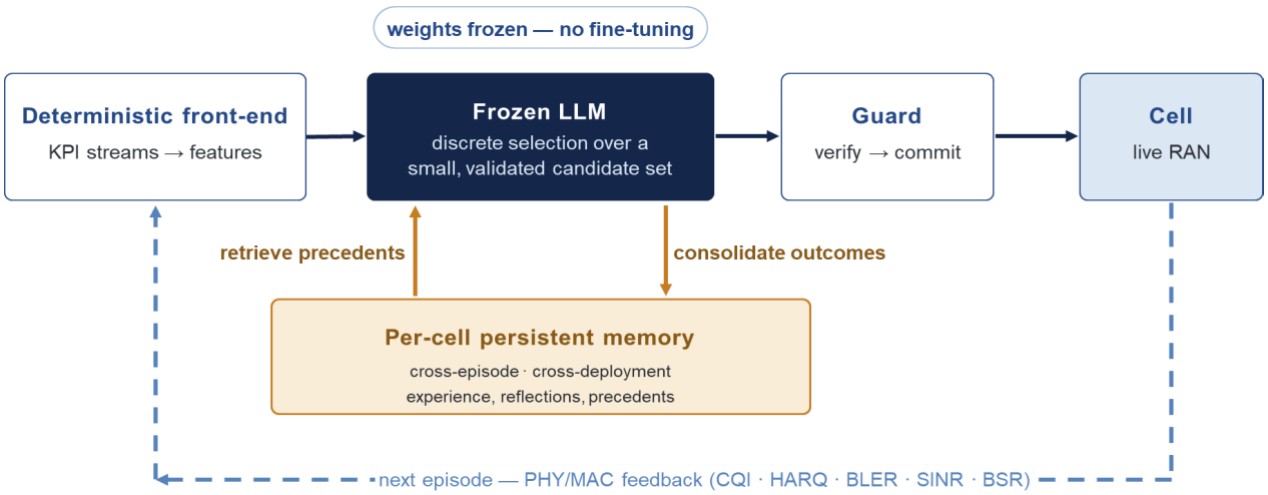

Figure 5.1. The self-improvement loop.

Physical-layer foundation models enter the architecture as governed tools rather than as replacements for physical-layer processing [38]. The agent selects, configures, validates and retires them through the same configuration surface, orchestrating the life-cycle management that 3GPP defines for air-interface models [39], while inference remains in the fast loop as a frozen, pre-compiled artefact.

5.2. Identity, Access, and Zero-Trust Architecture

The growing integration of autonomous AI agents into 6G mobile networks introduces security challenges that go beyond those addressed by conventional perimeter-based protection. These agents may independently make decisions, access network resources and external tools, interact with other agents, and influence critical network functions, creating risks such as impersonation, unauthorised access, model manipulation, and malicious agent collaboration. Addressing these risks requires a security model in which trust is not implicitly granted on the basis of network location or initial authentication. Zero-Trust Architecture (ZTA), founded on the principle of "never trust, always verify," provides an appropriate framework by continuously evaluating the identity, behaviour, context, and privileges of both human and autonomous entities. In Agentic AI-enabled 6G networks, this approach therefore combines secure identity management, least-privilege access, continuous trust evaluation, network isolation, and controlled agent collaboration to establish a resilient, dynamically adaptive security environment.

5.2.1. Zero-Trust Principles and Security Controls for Agentic AI

Translating Zero-Trust principles into Agentic AI mobile networks requires a combination of complementary security mechanisms that govern how autonomous agents are identified, authorised, monitored, isolated, and permitted to interact. Rather than relying on a single security control, Zero-Trust Architecture provides continuous, context-aware protection throughout an agent's operational lifecycle. This is particularly important in AI-native 6G environments, where agents may dynamically access network resources, collaborate with other agents, and influence critical network functions. Accordingly, the following mechanisms provide the core security controls for enforcing Zero Trust in such environments, covering secure agent identity and authentication, least-privilege operations, micro-segmentation, continuous trust evaluation, and secure agent collaboration.

Agent Identity and Authentication: In Agentic AI networks, AI agents act as independent entities capable of influencing network operations, such as optimising radio resources or addressing system faults. To ensure accountability and proper policy enforcement, each agent requires a secure, cryptographically verifiable digital identity. This is particularly important in AI-native 6G systems, where continuous authentication is essential, verifying agents not only at onboarding but throughout their activities. Supporting technologies such as Public Key Infrastructure (PKI), Decentralised Identity (DID), and hardware solutions such as Trusted Execution Environments are crucial. PKI allows agents to hold digital certificates, while DIDs facilitate cryptographic verification in decentralised networks. A core principle of zero-trust architecture is continuous authentication, which monitors agents for behavioural changes and compliance, enabling dynamic adjustments to access permissions. Additionally, verifying the authenticity of AI models, including their origins and training history, is vital to prevent manipulation. Future AI-native 6G designs will focus on secure lifecycle management for both agents and the models they utilise.

Least-Privilege Agent Operations: The principle of Least-Privilege Agent Operations is crucial to Zero-Trust Architecture (ZTA) in Agentic AI Mobile Networks. It holds that users, applications, and, importantly, autonomous AI agents should be granted only the minimum permissions required for their functions. As 6G networks advance towards distributed, multi-agent models, strict privilege management is vital for security and trustworthiness. In agentic systems that use Large Language Models (LLMs) and other advanced capabilities, limiting permissions minimises the attack surface and potential damage from breaches. A least-privilege framework includes role-based access control (RBAC), in which agents receive permissions based on their roles, and attribute-based access control (ABAC), which considers contextual factors such as agent identity and network status to enable dynamic security policies. Continuous monitoring and auditing of access requests and agent behaviour are essential for enforcing least-privilege principles. This helps identify unauthorised access attempts and trigger corrective actions. Ultimately, by

restricting agent permissions, enforcing contextual authorisation, and maintaining oversight, operators can reduce cybersecurity risks while benefiting from autonomous network functions.

Continuous Trust Evaluation: Continuous Trust Evaluation is a critical component of Zero-Trust Architecture (ZTA) for Agentic AI Mobile Networks. Unlike traditional security models that assume trust after authentication, AI-native mobile networks require ongoing trust assessment because of the dynamic interactions among autonomous agents. Trust must be continuously evaluated throughout the operational lifecycle of each agent, network function, and service, aligning with the Zero-Trust principle of "never trust, always verify," particularly in future 6G systems with high levels of automation. At the heart of Continuous Trust Evaluation is the dynamic trust score, which evolves in response to operational behaviour and contextual factors. Each agent, function, user, or device is assigned a trust level influenced by factors such as authentication history, policy compliance, resource usage, and communication patterns. Rather than relying solely on identity verification, networks assess entities based on their behaviour within authorised scopes. Consistent, normal behaviour results in high trust scores, while suspicious actions reduce trust. This approach also enables automated responses within a Zero-Trust framework, allowing systems to adjust authorisation policies based on trust scores without human intervention. Suspicious behaviour may trigger re-authentication, reduced privileges, or other protective measures, while trustworthy agents maintain access without disruption. This dynamic process balances security and operational efficiency in autonomous environments.

Micro-Segmentation: In Agentic AI Mobile Networks, micro-segmentation enhances traditional network partitioning by organising resources by operational function, trust level, data sensitivity, and agent responsibilities. This creates separate segments for network components such as RAN controllers and security systems, each functioning as an independent trust domain with defined access rights. This limits the impact of compromised agents; for instance, a traffic-forecasting AI can access only necessary data while remaining isolated from sensitive subscriber information. A key benefit of micro-segmentation is reducing lateral movement for attackers. If one component, such as a RAN optimisation agent, is compromised, it does not grant access to others, such as core databases, without additional authentication. This containment strategy effectively protects against complex cyberattacks in autonomous networks. Micro-segmentation also enables dynamic, risk-based security management, adjusting segment boundaries and access permissions based on real-time assessments and threat intelligence. Suspicious agents can be isolated during incidents, and emergency response agents can temporarily access additional resources as needed. However, managing policies across large-scale 6G networks with numerous AI agents is challenging and requires advanced orchestration and standardised trust frameworks. Excessive segmentation may introduce operational overhead if authorisation decisions become too frequent. Future research aims to address these challenges through AI-driven policy management and adaptive segmentation techniques, ensuring security without compromising efficiency.

Secure Agent Collaboration: Agentic AI systems involve multiple specialised agents, such as those for traffic forecasting, radio optimisation, network slicing, and security, to manage complex operations, such as congestion management. Effective collaboration relies on efficient information exchange and coordination, but risks arise if agents are compromised. Implementing strict Zero-Trust principles is essential for secure collaboration. Mutual authentication is crucial; agents must verify each other's identities with secure credentials before sharing data, particularly in multi-vendor Open RAN contexts, to prevent impersonation by adversaries. Additionally, fine-grained authorisation controls must restrict access to network information and services in accordance with policies governing agent communication and data sharing. Trust-aware information sharing is vital, as the network's reliability depends on the trustworthiness of data received from peers. Scalable trust-management frameworks are necessary for dynamic agent interactions, especially in cross-domain collaborations dealing with interoperability challenges. Emerging threats underscore the need for advanced secure multi-agent systems and research into AI-native networks to support future autonomous telecommunications infrastructures.

5.2.2. Zero-Trust Architecture for Agentic AI Mobile Networks

As 6G networks evolve toward fully autonomous ecosystems, implicit trust between network entities is no longer sufficient. The proposed Zero-Trust Architecture integrates continuous verification with multi-agent intelligence, providing a secure foundation across the entire mobile network lifecycle.

As illustrated in Figure 5.2, the framework comprises six core layers:

- Identity and Authentication Layer: Establishes and manages verifiable digital identities for all users, devices, and AI agents using mechanisms like PKI, DIDs, and Trusted Execution Environments (TEEs).
- Trust Assessment Layer: Abandons static trust in favour of dynamic trust scores, which are continuously calculated using behavioural analytics, agent reputation, and threat intelligence.
- Policy Engine: Acts as the central decision-maker (utilising RBAC and ABAC) to evaluate access requests and enforce context-aware security rules dynamically.
- Agent Control Layer: Coordinates the AI agent ecosystem, providing secure task delegation, tool access control, and authenticated multi-agent orchestration under least-privilege principles.
- Network Service Layer: Hosts operational telecommunications components (e.g., Open RAN controllers, core networks, edge platforms) within isolated, micro-segmented trust domains.
- Monitoring and Auditing Layer: Ensures comprehensive visibility and accountability through SIEM systems and explainable AI (XAI) to detect anomalies and support incident response.

These layers are supported by cross-cutting Zero-Trust principles, including micro-segmentation, secure collaboration, and continuous evaluation. This ensures that security is embedded in every interaction rather than just at the network perimeter. Ultimately, the convergence of Agentic AI and Zero Trust provides the robust, scalable security posture required for the resilient and self-healing 6G mobile networks of the future.

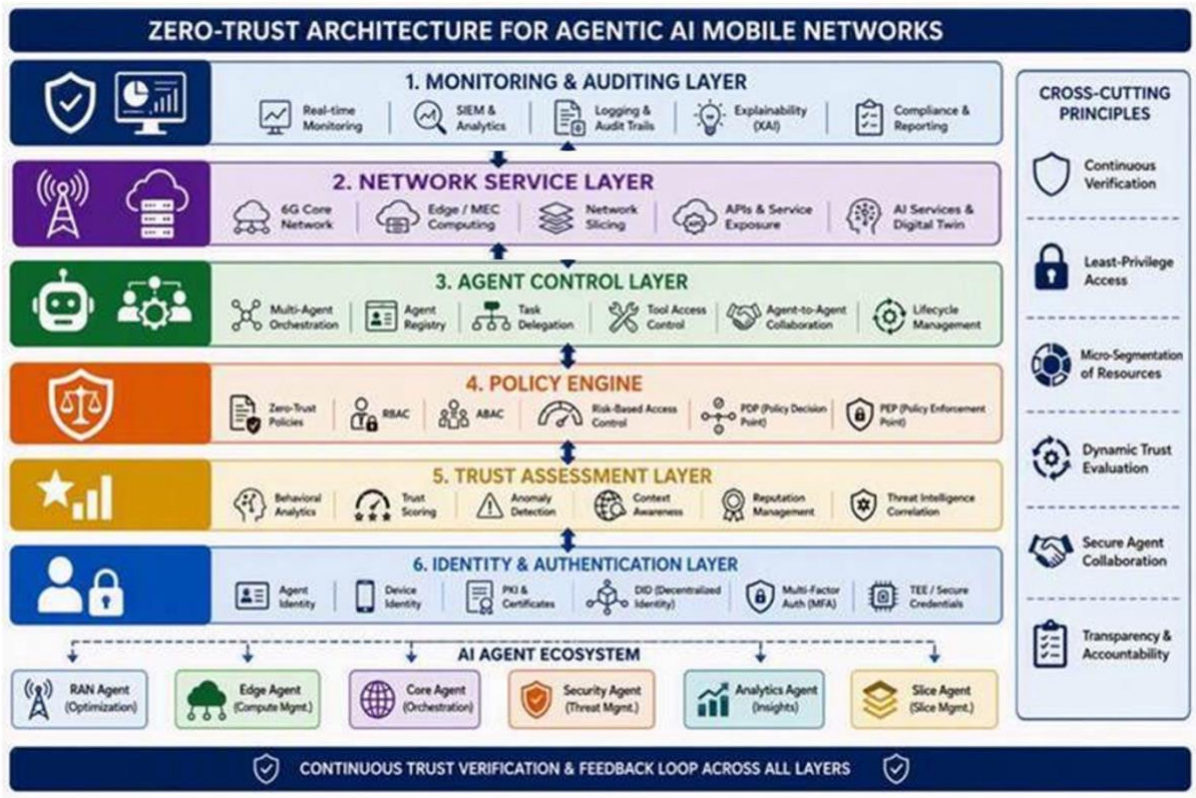

Figure 5.2. Zero-Trust Architecture for Agentic AI Mobile Networks.

5.3. Secure Model Supply Chain and Provenance

A Secure Model Supply Chain and Provenance Framework is the set of mechanisms, policies, and technologies that ensure every AI model, agent, dataset, prompt library, tool, and software component used in an Agentic AI Mobile Network can be trusted, verified, traced, audited, and protected throughout its entire lifecycle [40], [41], [42]. In essence, it extends software supply-chain security concepts to AI-native 6G networks, where autonomous agents continuously use, update, and exchange AI models [41], [42].

In traditional mobile networks, security is primarily focused on network functions and software components. However, in Agentic AI systems, AI models themselves become operational assets that directly influence network behaviour and decision-making [40], [43]. A compromised, poisoned, or unauthorised model could misallocate radio resources, degrade Quality of Service (QoS), manipulate network slicing policies, or trigger incorrect autonomous actions [43], [44]. Operators must therefore be able to verify where a model originated, how it was trained, who modified it, and whether it has maintained its integrity throughout deployment and operation.

A secure model lifecycle begins with trusted data acquisition and preprocessing. Since AI models learn from training data, poisoning attacks targeting datasets may introduce hidden biases or malicious behaviours into network operations [43], [45]. To mitigate such risks, dataset lineage records should preserve details about data sources, collection methodologies, preprocessing transformations, and validation procedures. Continuous monitoring of dataset quality and integrity can further reduce vulnerabilities associated with compromised or manipulated training data.

AI models and related components may be developed, trained, fine-tuned, validated, or distributed by different actors, including vendors, cloud providers, research organisations, third-

party AI suppliers, and network operators. This multi-party ecosystem increases the risk that model integrity or provenance may be compromised through poisoned datasets, malicious updates, manipulated weights, unauthorised fine-tuning, or insecure distribution channels. Therefore, supply-chain controls are required not only to verify the technical integrity of each model but also to establish accountability across the organisations and lifecycle stages involved in producing, approving, deploying, and updating AI components for autonomous network operations.

The resulting risks are amplified because agentic systems can autonomously execute actions across the network. The framework can therefore be understood as a flow of trust composed of the following architectural components, presented in the order in which trust is progressively established.

(i) Secure Dataset Lineage

Secure Dataset Lineage ensures the traceability, integrity, and accountability of datasets used throughout the AI model lifecycle. It maintains a verifiable record of dataset origins, collection and processing methods, transformations, and quality assessments from data acquisition to model training and retraining. Each dataset version is uniquely identified and cryptographically protected, enabling the detection of unauthorised modifications, integrity deviations, or compliance violations. By establishing a trusted lineage chain, the architecture allows network operators and autonomous agents to assess whether deployed models have been trained on approved, high-quality, and policy-compliant data sources. Furthermore, Secure Dataset Lineage supports auditability, reproducibility, and regulatory compliance by enabling stakeholders to trace model behaviour back to the underlying training data.

(ii) Cryptographic Model Signing

Every approved AI model should be digitally signed before deployment, e.g. by an authorised entity, such as the network operator, model developer, or trusted AI governance authority, using an appropriate key infrastructure. The generated signature is bound to the model's unique hash and associated metadata, allowing receiving agents, edge nodes, and network functions to verify that the model originates from a trusted source and has not been altered, corrupted, or replaced during storage or transmission. Combined with the Model Provenance Registry, Cryptographic Model Signing establishes a verifiable chain of trust across the AI supply chain

(iii) Model Provenance Registry

Every approved AI model should be registered in a trusted Model Provenance Registry that maintains its complete lifecycle history, lineage, integrity metadata, and deployment status. It records essential provenance information, including model origin, training datasets, validation status, responsible entity, approved use cases, and deployment history. By providing immutable traceability and verification capabilities, the registry enables autonomous network agents to verify model authenticity before execution. Furthermore, it facilitates secure model updates and rollback operations, ensuring that decisions made by AI-driven network functions can always be linked to a verified and accountable model version.

(iv) Agent Provenance

Not only models, but also AI agents must possess provenance information. Agent Provenance captures lifecycle information of autonomous AI agents, including their identities, assigned roles, underlying models, decision-making processes, accessed data sources, executed actions, and interactions with other agents [42]. This enables network operators to trace network decisions back to the specific agent and model version responsible, supporting incident investigation and accountability. Agent provenance should also support the verification that autonomous actions remain aligned with applicable security, operational, and governance requirements across cloud, edge, and network domains.

Figure 5.3 depicts the main components of the Secure Model Supply Chain and Provenance framework and the flow of trust.

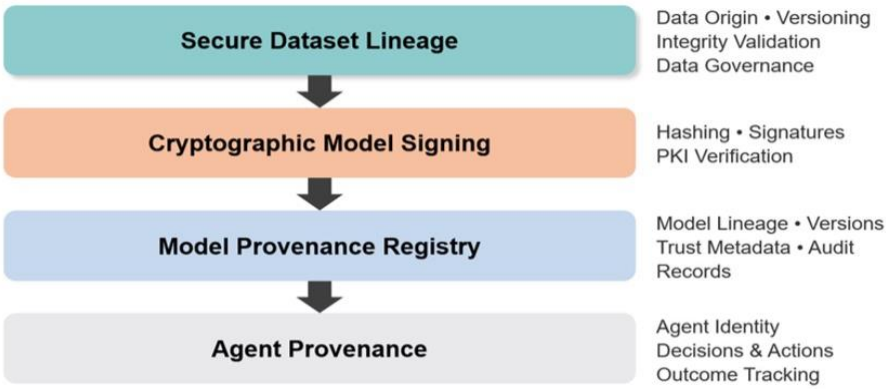

Figure 5.3. Secure Model Supply Chain and Provenance framework and flow of trust

In summary, Secure Model Supply Chain and Provenance mechanisms allow operators to introduce AI-driven autonomy without losing accountability, traceability, or operational control. By ensuring that datasets, models, agents, and runtime actions remain verifiable across their lifecycle, the framework strengthens trust, resilience, and compliance in Agentic AI mobile networks.

5.4. Adversarial Robustness and Model Hardening

The deployment of distributed intelligence across a heterogeneous network fabric introduces a broad attack surface vulnerable to prompt injection, model poisoning, and adversarial data manipulation. To maintain systemic resilience, security cannot be handled through incremental patches; it must be treated as an architectural design constraint embedded directly into the system fabric.

The critical deployment challenge is not whether AI agents will exist, but how their operational authority is explicitly bounded and how their runtime decisions are governed by continuous policy validation. The cognitive fabric implements automated guardrails that intercept agent actions in real time and validate synthesised behaviours against an immutable set of operational rules. This secure-by-construction framework isolates compromised or behaving erratically agents, ensuring that malicious inputs or corrupted model weights cannot translate into unsafe physical actions or cascading network failures.

Adversarial robustness for agentic AI in telecom is a larger problem than the evasion-attack literature it inherits from, because the object under attack is no longer a single classifier producing a label but a full perception–memory–reasoning–action pipeline, each stage of which is independently attackable (Figure 5.4). Classic adversarial machine learning treats a model as an isolated function and asks how a bounded input perturbation can flip its output; an agentic system is closer to a distributed program with several trust boundaries, and hardening means reasoning about all of them at once, not only the one attackers have historically targeted.

Two of the four stages are attacked mainly by corrupting what the agent is allowed to believe. At the perception boundary, data and telemetry poisoning at edge or RAN sources - a compromised counter feed, a spoofed KPI, a manipulated log line - can bias the agent's view of the network before any reasoning happens, the same threat class already documented against ML-based systems. One layer downstream, prompt injection targets the agent's context rather than its training data: untrusted telemetry, tickets, or logs ingested into the agent's working memory can carry instructions the agent then follows as if they came from its operator, a failure mode with no analogue in pre-agentic ML security and first documented systematically for LLM-integrated applications by Greshake et al. [46]. Because this attack rides in on data the agent is supposed to read anyway, it is largely invisible to input-validation checks designed around numeric or structural

anomalies, and it compounds directly with the memory-provenance problem raised in §4.2 — an agent that cannot tell a stale record from a hostile one is exposed on two fronts at once.

The remaining two stages are attacked by corrupting what the agent decides or reveals. Tool-call hijacking manipulates the agent's chosen action directly, inducing it to invoke the wrong tool or the right tool with the wrong parameters, rather than manipulating an output label as classic evasion attacks do. As a result, a detector tuned to catch out-of-distribution inputs will not detect it.

Evasion attacks against the ML-based intrusion- and anomaly-detection modules an agent relies on or coordinates with degrade its situational awareness without ever touching the agent itself. And because production agents are queried repeatedly against the same underlying model, that model is exposed to model-inversion and membership-inference attacks that can leak whether a specific subscriber's telemetry appeared in training or fine-tuning data [47]. A privacy exposure distinct from, and additive to, the zero-trust data-minimisation concerns raised in §5.2. Jailbreaking an agent's policy or safety instructions through crafted input sits across both stages: it corrupts the reasoning that selects an action, but its effect is only visible once that action is attempted.

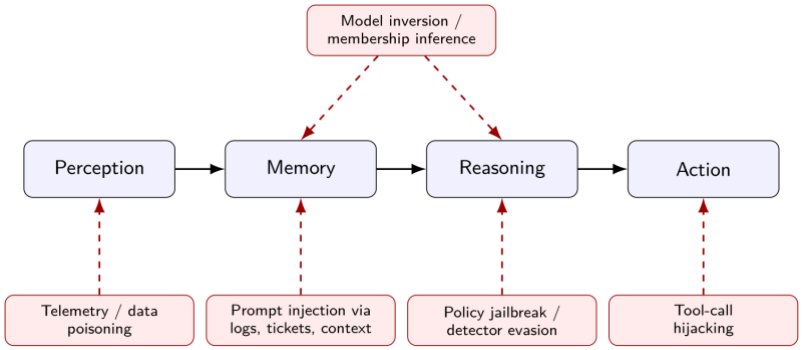

Figure 5.4. The perception–memory–reasoning–action pipeline shown earlier, now annotated with the attack vector aimed at each stage; model inversion and membership inference target the underlying model wherever it is invoked.

No single defence covers all four stages, so hardening has to be layered rather than concentrated at the model (Figure 5.4). At the data layer, cryptographic signing and attestation of model weights, prompts, and telemetry sources provide a way to reject inputs and artifacts that cannot prove their provenance across the decentralised and edge deployments a mobile network actually runs on [42]. At the context layer, input sanitisation and provenance/freshness checks on telemetry and logs entering the agent's context catch prompt injection and stale-memory attacks before they reach the reasoning stage, independently of anything the underlying model does. At the model layer, adversarial training, certified-robustness techniques such as randomised smoothing with a provable perturbation bound [48], and defensive distillation reduce the model's own susceptibility to crafted inputs. At the action layer, every proposed tool invocation must first pass strict API schema validation and least-privilege role-based access control (RBAC). Once structurally verified, an output/action validation gate, which is independent of the model that proposed the action, checks the proposed action against policy and the live network state before it is allowed to execute on infrastructure. If the action falls outside its authorised envelope, it escalates to a human operator, connecting to the human-in-the-loop mechanism discussed in §5.8.

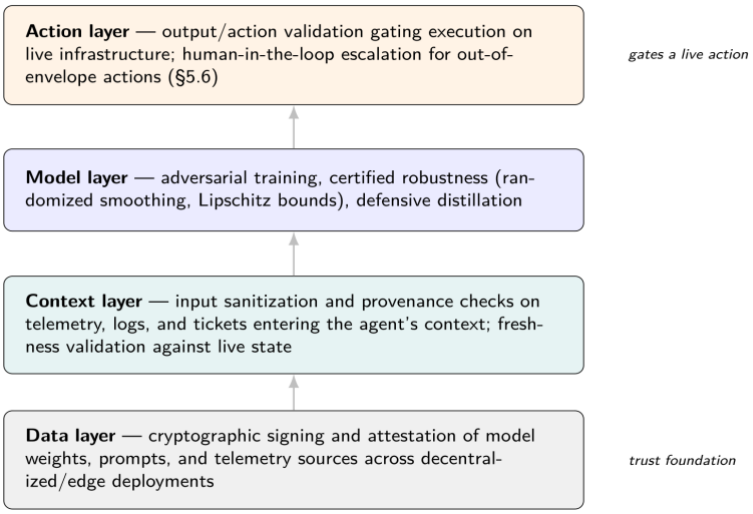

Figure 5.5. Defence-in-depth across the four attack layers: cryptographic attestation at the data layer, sanitisation and freshness checks at the context layer, certified-robustness techniques at the model layer, and an independent validation gate at the action layer.

None of this is free, and in a telco-grade control loop that matters as much as whether the defence works at all: robustness overhead has to be characterised against the same real-time budgets that bound the control loops themselves, from the functions near the radio layer with sub-second cycle up through slower traffic-engineering and capacity-planning loops. Certified-robustness techniques, in particular, are not a fixed-cost add-on. Randomised smoothing, for instance, requires multiple noisy inferences per decision to produce its certificate, directly multiplying the token and latency cost already treated as a first-order operating constraint in §7.1. So a technique that is unconditionally the most robust option on a benchmark may still be the wrong choice for a loop with a millisecond-scale deadline. Treating robustness overhead as a budgeted resource, quantified per technique against each control loop's real-time and cost envelope rather than assuming it is free, is, in our view, the more consequential technical contribution of this section, and remains open work.

Since these defences cannot be validated for free on an active network, red-teaming a strengthened agent pipeline requires a testing environment that mimics production without being part of it. This can be achieved with a digital-twin sandbox, increasingly suggested as a comprehensive test-bed for AI-enabled network security and closed-loop validation [49]. In such an environment, adversarial telemetry, injected prompts, and hijacked tool calls can be tested against the entire pipeline. Results from these tests can then inform retraining and policy updates before a hardened agent is integrated into a live control loop. This validation process should be directly linked to the digital-twin work described in §4.5, rather than established as a separate system.

5.5. Autonomous Decision-Making and Safety Boundaries

Mobile communication systems often require formal safety assurance, which can only be achieved through understanding the underlying system and its properties. When such formal assurance is not required, formal verification may be omitted. However, the critical question remains: when should systems no longer operate as defined in their design? Therefore, we propose physics-informed agentic AI models capable of fundamentally describing the physical properties of the communication system being modelled, where data enrichment is required only for learning behavioural aspects. Such models can describe fundamental behaviour and, through learning

from individual subsystems, deliver more precise models that can nevertheless be verified using formal verification methods.

Control Barrier Functions (CBFs) provide a mathematical framework for enforcing safety constraints in multi-agent systems by defining admissible state spaces that agents must not violate. In mobile communication networks, CBFs can encode hard constraints such as interference thresholds, latency bounds, and resource allocation limits, ensuring that autonomous agent decisions remain within safe operational envelopes. By integrating CBFs with physics-informed foundation models, the system can achieve verifiable safety guarantees while retaining the adaptability of data-driven learning – aligning with the telco-grade enforcement principles and the zero-trust governance architecture, outlined in this paper. This hybrid approach enables scalable multi-agent coordination where safety is mathematically guaranteed rather than statistically inferred.

For robotics, the algorithmic design of multi-robot coordination requires a systematic combination of model-based and data-driven methods to handle the inherent complexity of distributed decision-making under uncertainty. Model-based approaches, particularly those grounded in differential game theory, provide rigorous frameworks for analysing competitive and cooperative interactions among agents with conflicting objectives. Potential differential games extend this foundation by enabling agents to converge toward Nash equilibrium (a stable state in a game) while respecting coupled safety constraints encoded through CBFs. Data-driven learning-based methods complement these model-based techniques by adapting to unknown environmental dynamics and agent behaviours that are difficult to capture analytically. Together, this hybrid algorithmic framework ensures that multi-robot coordination in mobile communication networks achieves both computational tractability and formal safety guarantees, supporting the scalable deployment of agentic AI systems across heterogeneous network topologies.

5.6. Multi-Agent Conflict Resolution and Negotiation

The deployment of multiple agents and the description of their communication explicitly leads to potential *conflicts*. Such conflicts include, for example, access to resources, management of communication channels, and execution of various tasks across different robotic systems. For such mobile robotic systems, it is therefore critical that the resolution of such conflicts is first modelled and subsequently resolved.

The modelling of inherent conflicts arising from multi-agent system interactions can be approached in various ways. A novel approach in the literature involves considering the inputs of such systems and establishing a model that accounts for agent or subsystem inputs. This eliminates the need to measure the internal states of individual agents or subsystems at every step, thereby conserving resources and enabling more general model descriptions that do not require precise internal modelling. The goal should be to develop generative models that only require agent inputs to describe emerging conflicts; depending on the difficulty and complexity of the systems, various methods and models must be developed accordingly.

This input-based conflict modelling corresponds directly with the cognitive network control architecture from Section 2, where agents operate within bounded authority envelopes while maintaining system-wide observability. By abstracting internal agent states, this approach supports distributed intelligence across device-edge-cloud continuum (see Section 2.2) and complements memory architectures from Section 4.2, enabling conflict pattern retrieval without exposing sensitive reasoning processes.

The first step is the *detection* and *modelling* of conflicts. In the second step, these conflicts must be resolved. However, conflict resolution remains insufficiently explored in the literature. Some methods can resolve conflicts by having one agent yield and release resources to others. In robotic communication systems, prioritisation of communication channels (determining who can communicate with whom and when) is widely adopted. Such multi-agent systems employ negotiation mechanisms to resolve existing conflicts. This resolution can occur *centrally* or

decentrally. Centralised resolution offers the advantage of achieving better and more optimal solutions that truly reach the system optimum. Alternatively, decentralised methods enable individual agents to resolve emerging conflicts through interaction and mutual adaptation strategies. This approach has the advantage of not requiring a central communication unit, allowing mobile robotic systems to adapt more dynamically and decentrally. On the other hand, such methods do not always yield optimal solutions due to limited inter-agent communication.

The choice between centralised and decentralised conflict resolution must follow telco-grade enforcement principles discussed in this paper, where deterministic guardrails ensure safety when probabilistic negotiations fail. Hybrid architectures combining rule-based resolution for critical conflicts with ML-based negotiation offer a pragmatic path for 6G agentic networks. Ultimately, effective conflict resolution is foundational to the one6G vision of trustworthy and self-optimising systems balancing autonomy with governance.

5.7. Auditability, Explainability, and Decision Traceability

Telco-grade Auditability

Agentic AI mobile networks shift part of network operations from deterministic procedures to AI-assisted, policy-bounded decision-making. This does not remove the need for deterministic operational control; rather, it increases the need for complete auditability across the full decision chain. All intent translations, multi-agent coordination processes, policy evaluations, and synthesised control actions must therefore be recorded with sufficient traceability to reconstruct how decisions were made and how actions were executed.
This traceability supports governance by balancing autonomous execution with operational control through three complementary mechanisms:

a) Policy-Bounded authority: The network must ensure that all composable network behaviours remain within explicit policy boundaries, preventing unauthorised changes, uncontrolled optimisation, or unextected configuration drift.
b) Runtime validation and stability guardrails: The AI-native network control layer should incorporate runtime verification mechanisms to assess whether proposed actions remain compatible with stability, safety, SLA, and telecom-grade operational constrains before they are executed.
c) Operator override, deterministic fallback, and rollback: To prevent multi-agent race conditions or runaway autonomous loops, the governance layer should expose controlled interfaces that allow human operators to suspend agentic autonomy, trigger safe rollback, and return the system to a known deterministic operating mode.

With Agentic AI, auditability must cover not only the final action and its outcome, but also the process that led to that action. In a conventional network-control function, the execution path is predefined, and the audit record primarily captures the input, configuration command, and resulting system state. In an agentic system, an action may instead emerge from a sequence involving intent interpretation, memory retrieval, foundation-model inference, policy evaluation, agent negotiation, and runtime validation. Telco-grade auditability must therefore reconstruct the complete intent-to-outcome chain, rather than merely record the final command issued to the infrastructure [50].

Each material agentic decision should generate a structured decision-trace record with a unique correlation identifier that remains valid across RAN, Core, Transport, OAM, device, edge, and cloud domains.

Operational Explainability

Explainability should be role-specific, evidence-based, and adapted to the operational timescale [51]. A real-time network operator needs a concise explanation of the proposed action, its rationale, relevant constraints, and possible rollback options. Post-incident investigators require a more detailed reconstruction of the data, models, agents, policies, and system states involved in the decision [52]. Service owners may instead require an outcome-level explanation linking network behaviour to the relevant SLA or mission objective [53].

Explanations should be generated from verified decision artefacts, policy checks, constraint evaluations, and observed outcomes. They should not depend solely on unrestricted post-hoc narratives produced by the same model that generated the decision, because such narratives may not faithfully represent the actual basis of the action.

Decision traceability

Decision traceability provides secure, time-synchronised, and tamper-evident records of agentic decisions and their execution. It should also allow decisions to be correlated across domains when several independently operated agents contribute to a single network action. Without such correlation, an operator may be able to observe individual domain commands but remain unable to determine how their interaction produced the final outcome.

In multi-tenant, multi-slice, and multi-operator environments, auditability must also preserve confidentiality. Namespace isolation and selective disclosure mechanisms are required so that one tenant's prompts, historical records, SLA parameters, incidents, or optimisation strategies are not exposed to another party. Complete decision traceability therefore does not imply indiscriminate retention of all raw data. Trace generation should follow data-minimisation, purpose-limitation, access, and retention policies.

The traceability service should support root-cause analysis, post-event replay, responsibility allocation, and evidence export for assurance, audit, or regulatory assessment [54]. Where a safety-, security-, or service-critical action cannot be linked to a complete and verifiable decision trace, the cognitive control layer should fail closed, restrict the agent to advisory operation, or transfer control to a predefined deterministic mechanism. Auditability is therefore not merely a post-operation reporting function; it is a runtime condition for granting and retaining agentic authority.

5.8. Human-in-the-Loop and Override Mechanisms

Human oversight in agentic mobile communication systems should be designed as a risk-adaptive control function, rather than as a universal requirement for manual approval. Requiring operator confirmation for every action would eliminate much of the responsiveness, scalability, and economic value expected from agentic control. Conversely, unrestricted autonomy would be incompatible with telco-grade safety, security, accountability, and service continuity. Decision authority should therefore be allocated dynamically according to action criticality, uncertainty, reversibility, affected scope, available response time, and the operational maturity of the agent. Human-in-the-loop mechanisms combine automated learning and decision-making with human expertise, potentially improving model performance, accountability, and transparency, while also introducing implementation challenges related to data quality, human bias, user engagement, privacy, and the consistency of human feedback [55].

This could be implemented through three complementary oversight modes [56]:

a) Human-in-command governance: authorised persons define the operational intent, prohibited actions, authority envelope, escalation thresholds, applicable policies, and required safe states before the agent is deployed.

b) Human-on-the-loop supervision: agents execute routine and reversible actions within validated boundaries, while operators monitor system behaviour, receive prioritised notifications, and retain the ability to intervene.

c) Human-in-the-loop approval: explicit human authorisation is required before executing exceptional, high-impact, difficult-to-reverse, insufficiently evidenced, or out-of-envelope actions.

These modes may change during operation. The active oversight mode should be explicitly represented in the agent's authority profile and included in the decision trace. It may change during operation. For example, an agent may normally operate under human-on-the-loop supervision but automatically revert to human-in-the-loop approval following a security alert, significant data drift, repeated rollback, or loss of reliable situational awareness [57]. Escalation should be based on predefined, machine-detectable conditions such as safety, security, uncertainty, validation failure, or actions exceeding delegated authority

When human intervention is required, the human operator should receive a structured decision package, rather than an undifferentiated alarm, to support timely and informed intervention. This package should identify the current intent, affected resources and stakeholders, proposed action, supporting evidence and its freshness, expected benefits, uncertainty, applicable constraints, relevant alternatives, impact scope, reversibility, time remaining for intervention, and the recommended fallback. The interface should make clear which elements are observed facts, model estimates, policy requirements, and agent-generated recommendations. Manual override must be deterministic, authenticated, role-based, and logically independent of the agent or foundation model that is controlled. For high-impact or cross-domain interventions, dual authorisation may be required. Override and rollback actions should be coordinated across domains so that RAN, Core, Transport, OAM, edge, and service-management functions do not remain in inconsistent states. A partial rollback may be as hazardous as the original action and should therefore be treated as a transaction whose completion or failure is explicitly monitored.

Effective human oversight also depends on human factors, consequently, those human factors must be considered as part of the end-to-end system architecture. The human operator's workload, expertise, fatigue, directly affect the effectiveness of oversight.

Human approval should not be used to transfer all responsibility to an individual operator. Accountability remains distributed across the agent developer, foundation-model and tool providers, network operator, service provider, system integrator, and the organisation responsible for assigning decision authority.

To make these arrangements explicit, each agentic network function should be accompanied by an oversight-and-override profile defining its normal autonomy level, authorised supervisory roles, escalation conditions, and safe states [58].

6. Standardisation Considerations

As we have noted, there has been a remarkable move from simple AI workflows to defining and designing AI agents capable of cognition, autonomous actions, collaboration, and self-learning. This pace of paradigm change has necessitated standardisation across layers, from domain-specific architectures to governance, communication protocols, service awareness and management, collaborative computing, trust and security, safety, discovery and identity, and interoperability. The standards bodies working on this with somewhat distinct and complementary focus range from 3GPP, ETSI, IETF, ITU, and industry organizations (e.g. NGMN, GSMA, GTI) together with open-source communities (e.g. Linux Foundation), to ISO, IEEE, OASIS, and national or regional bodies.

6.1. 3GPP Standardisation Progress

Although the maturity of a cognitive composable fabric may go through phases, we believe a native-AI 6G shall, from the start, include the key principles of Agentic AI for mobile communications systems discussed so far in this paper. In particular, the initial 6G architecture should support autonomous and specialised agents, persistent, shared network-state representations, intent-based interaction, continuous learning, distributed decision-making, real-time digital-twin synchronisation, and controlled access to network actions. These capabilities should not be regarded as optional additions introduced only in later releases, but as foundational architectural principles that can progressively increase in sophistication, autonomy, and operational scope.

The transition towards a fully cognitive and composable fabric should therefore be understood as an evolutionary maturation process rather than as a delayed introduction of AI-native functionality. The underlying architectural foundations should be established from the beginning, while the authority delegated to agents, the complexity of their coordination mechanisms, and their capacity for self-learning and autonomous reconfiguration can evolve through successive stages. Such progression should rely on bounded authority, verifiable decision processes, standardised interfaces, continuous supervision, and deterministic fallback or rollback mechanisms.

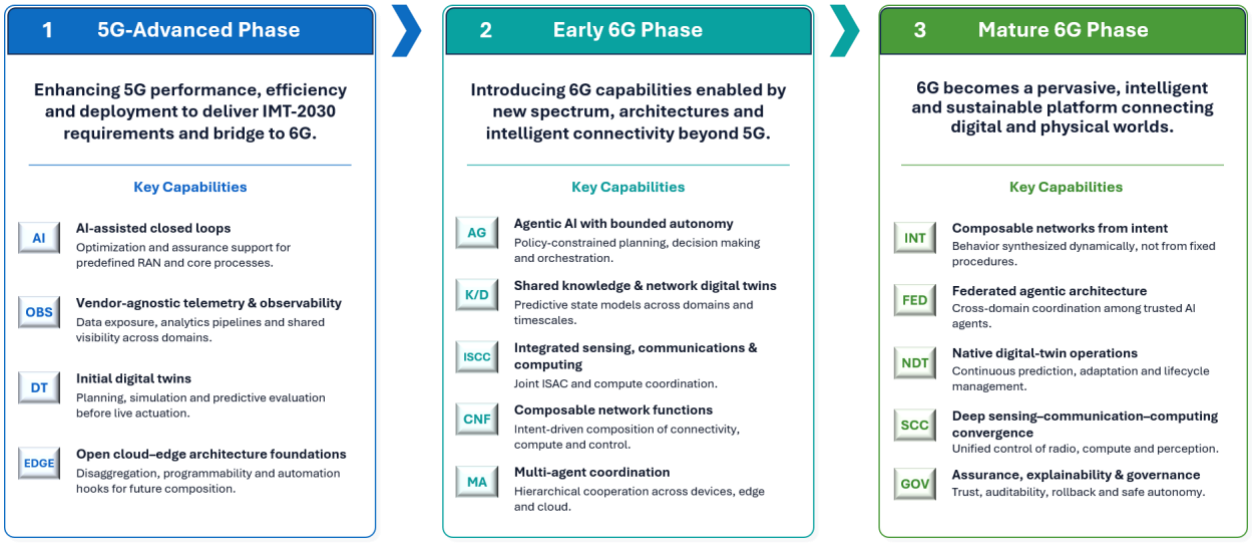

Figure 6.1. Evolutionary roadmap from 5G heuristics to 6G composable AI-native networks

The timeline in Figure 6.1 illustrates how recent industry developments and standardisation activities provide an evolutionary bridge towards this vision. Stages 1, associated with 5G-Advanced

and the completion of 3GPP Release 19, represents important advances in automated and data-driven network management. Capabilities based on the Network Data Analytics Function (NWDAF), for example, enable AI- and machine-learning-assisted functions related to network slicing, anomaly detection, mobility management, and service assurance. These mechanisms improve network observability and introduce increasingly automated forms of decision support and controlled actuation, while retaining deterministic procedures and rollback options. Stage 2 and 3 represent the stage at which these principles begin to be embedded more deeply and systematically across the architecture. Work associated with 3GPP Release 20 can contribute to the evolution of the 5G Service-Based Architecture into an AI-ready, progressively agentic framework capable of supporting persistent prediction, shared network state and knowledge models, continuous digital twin synchronisation, and increasingly autonomous closed-loop operation.

Subsequent stages should consequently reflect the progressive expansion of capabilities already present in the initial 6G architecture. This includes richer agent cooperation, broader cross-domain reasoning, more advanced self-learning mechanisms, dynamic composition of network functions and resources, and an increasing capacity to translate high-level intents into coordinated actions. The roadmap, therefore, describes growing maturity, autonomy, and composability rather than a transition from a non-AI-native architecture to an AI-native one.

At this stage, the 3GPP completed study on 6G use cases and service requirements [68] indicates how 6G will target an AI-native wireless network, with AI incorporated into the network architecture, radio resource management, physical layer design, network security and application enhancement from the start. The ongoing study on the protocol for AI in 6G [69] studies the protocol aspects for supporting and enabling the use of AI (e.g. AI Agent, intent) in 6G, while considering the architectural requirements defined in [70], and the security requirements defined in [73]. It covers candidate protocols and protocol design considerations, highlighted in this paper, to support AI (e.g. AI Agent, intent), and address the protocol aspects for enabling exposure, discovery, authorisation, among others. Therefore, the protocol design considerations are expected to include intent related and AI Agent related ones (e.g., agent to agent interaction, agent to tool interaction). The collaboration with respect to protocols with other bodies such as IETF is not limited to aspects and design considerations but also considers timelines.

6.2. ETSI, ITU-R/T, IETF, Others

As the industry moves beyond primitive closed loops (Composable Control and cognitive network control), as depicted in Figure 6.1, standardised capability exposure becomes critical to prevent proprietary vendor fragmentation. Later evolutionary stages introduce structured, intent-driven templates that enable heterogeneous multi-vendor agent swarms to dynamically coordinate decisions across domain boundaries.

This transition toward autonomous, intent-synthesised operations is being actively formalised across expanding ecosystem bodies.

ETSI

ETSI GR ENI 051, 055, 56, and 60 [29], [30], [59], [72] analyse how AI-agent–based core networks (AI-Core) can support future 6G systems by enabling intent-driven, autonomous, and highly adaptive network behaviour. The report [51] explains the motivation for AI-Core—mainly the increasing complexity of 6G scenarios, the need for flexible orchestration of diverse network capabilities, and the emergence of intelligent devices such as AI phones and robots. It defines AI-Core as a multi-agent system where each agent performs specialised tasks (intent interpretation, planning, resource management, analytics, optimisation) and collaborates to create customised, on-demand network configurations. The document highlights advantages such as flexibility, autonomous operation, and continuous self-improvement through reflection, while also noting challenges including safety risks, semantic ambiguity in intent translation, and domain-specific limitations of LLM-based agents. The report [55] provides extensive use cases across three domains:

(1) Consumer type of use case (e.g. smart life assistants, collaborative robots, AI phones) (2) Enterprise / B2B type of use case (e.g. smart city traffic monitoring, construction sites, gaming acceleration, energy distribution) (3) Telecom operators themselves focused use case (e.g. autonomous network operation and management, disaster response, time-sensitive networking, signalling optimisation, user-experience enhancement). For each use case, the report outlines potential requirements such as real-time context acquisition, distributed inference, dynamic resource allocation, QoS/QoE assurance, and multi-agent coordination. It concludes that AI-Core can transform operators from pure connectivity providers into service integrators, enabling new business models where networks dynamically assemble third-party functions and tools to deliver customised services.

ITU

ITU is providing a comprehensive and multi-faceted mandate and platform, driven by the goal that AI plays an important role in accelerating progress towards achieving the UN Sustainable Development Goals. The ITU-R recommendations on framework and requirements for IMT-2030 (6G) prominently includes ubiquitous intelligence, and integration of AI and communication.

ITU's own initiatives on AI target building a common understanding of the capabilities of AI technologies, and facilitating trusted, safe, equitable, and inclusive development of AI technologies and availability. This is led by the action-oriented UN platform AI for Good, including pillars on solutions and knowledge, skills and capacity, as well as standards and policy to support policy framework and recommendations, and innovation. It has a collaborative forum, for instance, with ISO and IEC, to align technical requirements, promote interoperability, and develop global standards for content authenticity.

A significant area of related focus and contributions is the work of ITU-T AI Focus Groups. These are specialised expert groups driving pre-standardisation investigation for emerging AI applications, addressing needs as they emerge, and when not covered within an existing Study Group.

The ITU-T Focus Group on Artificial Intelligence Native for Telecommunication Networks (FG AINN) is part of Study Group 13 on Future Networks. It focuses on exploring and defining the fundamental changes needed in network architecture to seize the potential of AI, while identifying the requirements, challenges, and opportunities:

Understanding how AI can be embedded in the network design, moving beyond traditional models to create networks that are intelligent by nature; exploring novel applications enabled by AI-native networks, such as autonomous systems, smart cities, and real-time adaptive services; reviewing current technologies and standards to identify what is lacking in the transition towards AI-native networks; and establishing partnerships with other standard development organisations (SDOs) and industry/research groups to ensure a unified approach to AI-native networking.

FG-AINN has a number of deliverables, including a recent comprehensive work on standardisation gap analysis [71].

Other ITU-T active focus groups established recently include, for instance,

- Focus Group on Trust and Identity for Humans and Agentic AI
- Focus Group on AI for Smart Sustainable Cities and Communities
- Focus Group on Embodied Artificial Intelligence for Multimedia Technologies

The focus group on Embodied AI is part of ITU-T Study Group 21, which does have a broad range of focus on topics such as AI models, agents, distributed AI, and federated learning. On Embodied AI (EAI), it outlines the unified system framework, covering technical requirements for environmental perception and autonomous task execution capabilities. The concept and features are shown in Figure 6.2, highlighting the cognition, collaboration and self-learning loop. Cognition is defined as the collective ability to form an internal model of the world through physical *perception*, *decision-making*, and *human-machine interaction*.

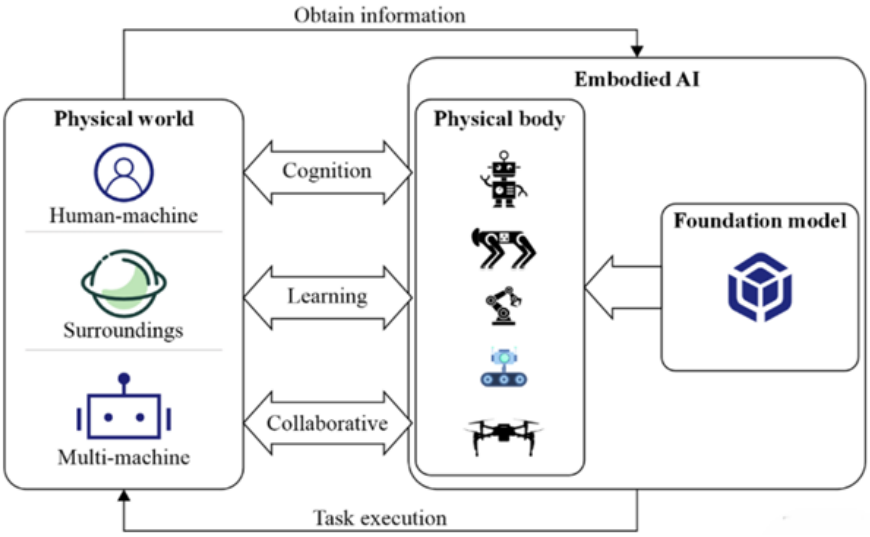

Figure 6.2 – Conceptual diagram and key features of embodied AI (ITU-T F.748.66)

IETF

The IETF works on a broad range of networking technologies. It publishes its technical documentation as RFCs (a traditional acronym for Requests for Comments) which describe the Internet's technical foundations.

IETF has an extensive repository of Internet-Drafts (I-D), cited as "work-in-progress". These are written by individual authors to distil and communicate an idea, need, way forward, or solution. Many I-Ds progress within the IETF and with enough interest and energy are adopted to be formally worked on by the group (Link: Active IETF working groups). In current Internet-Drafts repository there are many on the topic of Agentic AI and AI agents, from identity, authentication, discovery, and agent communication protocols, to use cases, and architectural, intent-based networking, orchestration, and management principles (Link: Active Internet-Drafts). These may include extensions to standards that have existed for years, e.g. identity management protocol for AI agents and agentic applications.

Another highly effective and relevant track at IETF involves the Birds of a Feather sessions (BOFs), which are basically initial discussions on a topic of interest to gather a sense within the community, even without intending to form a working group, though some do lead to forming one. The purpose nonetheless is to ensure a charter with milestones, support, and tight enough scope can be created for potential standardisation.

A BOF on Agent Communication has recently been formed with first session at IETF 126, building on earlier discussions on framework, use cases, and requirements. Its page also links to a rich list of related Internet-Drafts and RFCs.

There is an ongoing effort to respond to the pace of Agentic AI paradigm changes, as we notice in creation of new focus groups (as in ITU Study Groups), large number of Internet-Drafts (work-in-progress) on variety of essential topics or creation of BOF sessions (both as in IETF), or Introductory Guides (as in TMF). TMF IGs outline foundational and high-level concepts, architectures, models and attributes, e.g. context and principles for autonomous networks. For example, the objective of IG1444 is to specify the key elements of intelligent agents for the autonomous operation of computing networks, including architecture, functional requirements, and interaction mechanisms. IG1421A introduces an Agentic-AI service-management framework that uses a multi-agent system to express, translate, negotiate, and operationalise intent across the product, service, and resource layers. IG1463 defines the technical security considerations for Agentic AI in autonomous networks, including those related to reasoning, memory, tool use, authentication, human interaction, and multi-agent protocols (MCP, ACP, A2A, ANP).

While this range of activities to standardise different aspects or layers of Agentic AI agents or systems exists, there is a crucial need to ensure timely and unified standardisation of Agentic AI for mobile communication systems with the attributes expected for 6G from start.

7. Operational and Economic Implications

7.1. Token-Based Economic Models for 6G

Shifting to an Agentic AI-native architecture significantly alters the foundational infrastructure, transforming the traditional static base station and edge node setup into a scalable, cloud-enabled micro-data centre. Because this cognitive system runs natively on powerful processing platforms capable of handling signal processing, real-time radio resource management, and intelligent reasoning, it establishes a multi-layered resource framework. To monetise this distributed intelligence, it is necessary to move beyond outdated volume-based flat-rate billing models, which are not suitable for a network driven mainly by autonomous AI agents with bounded autonomy.

To address the challenge of service settlement in the upcoming 6G system, a dynamic, token-based economic framework is proposed. This framework shifts from traditional billing based solely on raw data volume to a more nuanced approach that uses multi-dimensional execution parameters. Transactions are now evaluated dynamically based on the precise number of computing cycles consumed, the levels of inference confidence achieved, and the carbon footprint limits associated with specific tasks.

In addition, resource allocations and multi-agent negotiations are facilitated through instantaneous smart contract settlements. These processes occur on a millisecond basis, leveraging automated token exchanges and programmatic smart contracts for efficiency.

Furthermore, to optimise infrastructure usage during periods of low radio traffic, operators can lease idle edge processing capacity to hyperscalers or local enterprises. This "Token-as-a-Service" model enables external AI inference workloads to be executed, thereby maximising hardware utilisation and enhancing the return on capital expenditures.

As also introduced earlier, “token” here denotes a unit of agent computation. It is the same currency used in the discussion of memory retrieval as a token-cost lever and in the discussion of certified-robustness overhead. Recent work formalising tokens as a factor of production, a medium of internal exchange, and a unit of account for agent systems gives this reading a name grounded outside telecom [60], and it is the one on which the economic model in this section is built. An agentic AI system consumes tokens at every internal step it takes towards a goal: self-reasoning, tool execution, abandoned or failed paths towards an action, and inter-agent or agent-coordinator communication all carry a token price. The gap between the effective token counts a correct result actually requires and the token count the system actually spends is largely due to reasoning overhead, retries, and coordination chatter. That gap is what this section’s economic model seeks to price, budget, and ultimately reduce.

The most direct application is cost-per-task or cost-per-intent accounting: mapping a network operation, such as a slice reconfiguration, anomaly resolution, or capacity-planning recommendation, to an expected token budget, much like how cloud FinOps (Finance plus DevOps) maps a workload to an expected compute bill. Figure 7.1 illustrates this for a slice-reconfiguration intent handled by a small pipeline of domain agents. A RAN agent reasons over the current cell load, a Transport agent checks backhaul headroom, a Core agent validates policy against the SLA, and a coordinating agent reconciles their proposals into a single action. Each agent’s token spent, including its own reasoning, tool calls, and the tokens spent describing its proposal to the coordinator, becomes a line item in a per-domain chargeback ledger. The sum is the total cost of the intent. The shared-memory design choice becomes a direct cost lever here, not merely an architectural convenience, since a shared semantic memory of the kind described in Section 4.2 allows agents to draw on a common network knowledge graph rather than re-

communicating their full context in every round. This per-intent figure facilitates internal chargeback between domains, such as RAN, Core, Transport, and OAM, or among tenants in a multi-tenant network. Telecom already addresses a similar accounting issue in the network-slice-broker literature regarding leasing and costing shared infrastructure resources across tenants [61], now expanded to include agent token consumption instead of only compute or bandwidth.

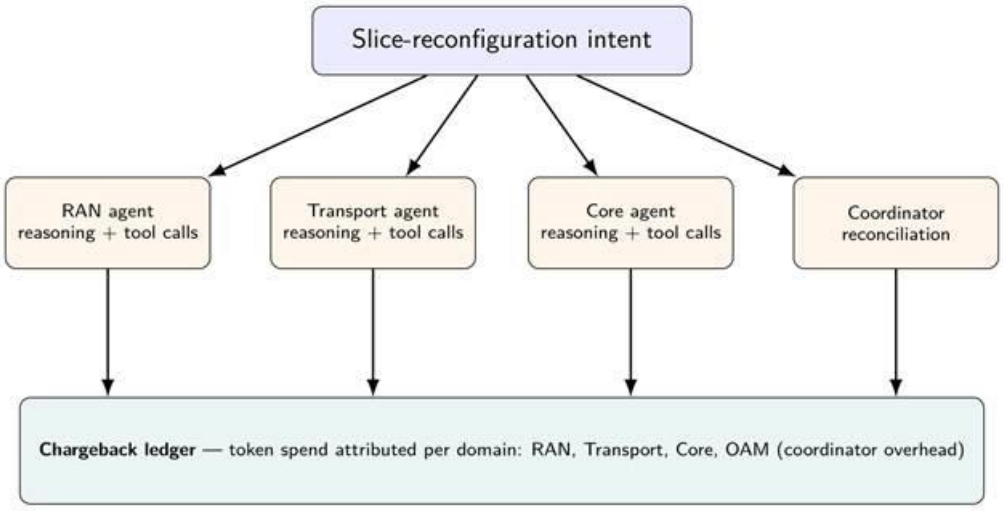

Figure 7.1. Token flow and chargeback for a slice-reconfiguration intent.

Cost-per-task accounting is only operationally useful if it is enforced, which argues for token-budget SLAs: a contracted maximum token, and therefore cost, envelope for a given agentic task, with defined escalation when that envelope is exceeded. The objective is to avoid a situation in which the agent silently runs over budget or degrades task quality to remain within it. Recent work on budget-aware multi-agent systems shows that this is not merely a policy statement: it can be built into the orchestration layer itself, selecting which agents and models participate in a task under an explicit budget constraint, rather than leaving spending as an emergent property of how agents happen to reason [62]. In a network operations setting, exceeding a token budget SLA is a natural trigger for the human-in-the-loop escalation discussed earlier. A task that needs materially more reasoning than budgeted is itself a signal: either the situation is more complex than expected, or the agent is stuck in an unproductive loop. In both cases, handing the decision to an operator may be preferable to letting the cost grow unbounded.

None of these matters if agentic AI is not the right tool for the operation in the first place. The token-economic model must therefore include the build-versus-buy-agentic decision as a first step, not as an afterthought. Figure 7.2 provides a conceptual cost-versus-complexity comparison between non-agentic automation and agentic AI. Non-agentic automation is usually cheaper for low-complexity, well-scripted tasks, because a rule-based system or classical control loop can execute predefined behaviour without spending tokens on planning, negotiation, or reasoning. However, its cost increases when each new case requires new scripted logic. Agentic AI carries a higher cost floor but may scale more gently for tasks that genuinely require dynamic planning, tool use, negotiation, and judgment under incomplete information. The economic crossover, therefore, occurs only when the flexibility of agentic AI offsets its reasoning and coordination overhead. Agentic AI should be considered economically justified where the task cannot be scripted in advance at a fraction of the cost [63]. Telecom already has early evidence on both sides of this trade-off: a hierarchical, agent-native architecture validated in a 5G Core environment quoted reduced mean time to repair by 86% relative to a static automation baseline [64]. This result would justify agentic AI's token cost for that class of fault-recovery task. For narrower, well-scripted operations, however, the same agentic pipeline would likely spend tokens re-deriving what a rule-based system already provides at lower cost. The key economic question is therefore not what agentic AI costs in isolation, but what it costs relative to the next-best alternative for a specific network function.

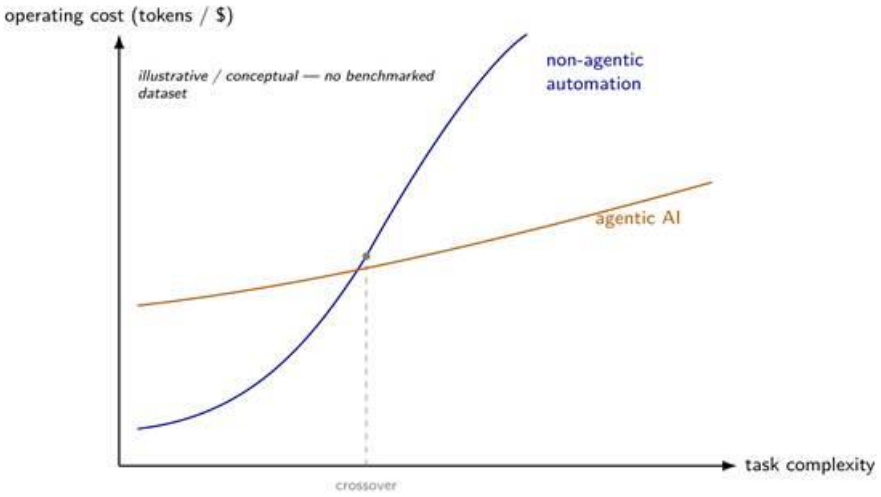

Figure 7.2. Conceptual cost-vs-complexity comparison between non-agentic automation and agentic AI.

7.2. Impact on OPEX, Workforce, and Processes

The operational efficiency of agentic autonomous systems is well articulated. Alternatively said, automation driven by real-time ubiquitous agentic intelligence is intuitively and explicitly about reduction and optimisation of operational cost and processes. Furthermore, the impact on workforce and proactive strategies to address or shape that cannot be overlooked. There are products testimonials or claims today on how, for instance, agentic RAN management targets a significant reduction in OPEX. Implicit in the operational efficiency, is the impact on workforce, and on range of processes that would otherwise be done manually or with conventional levels of automation. This is about impact and not necessarily an exact or equal replacement, as the prospects of an agentic AI-native 6G are expected to be more than a direct cost saving, as outlined in this paper.

Not all designs, systems, or sets of capable AI agents, however, will succeed, say, in transforming an enterprise business. Success is not limited to skilful design of high-performance systems. What has been discussed so far and has pointed to intuitive operational efficiency through cognitive automation, also introduces new paradigms and potentially new complexities in transition. This paradigm shift demands architectural and operational design imperatives and strategies, to balance the expected significant gains with clear operational efficiency, while providing scalability, modularity and versatility, interoperability, and simplified solutions.

The impact on workforce has multiple dimensions, from how roles may be replaced to how they may be reshaped. There is also a great deal of optimism on how this transformation can potentially boost economy and create opportunities, how AI and robotics may replace tasks rather than occupations, augment humans in difficult conditions, in novel tasks, or in response to shortages. No matter what and how, the impact cannot be overlooked and must be proactively considered by all stakeholders in programs and strategies. In this essential and remarkable journey, there is a growing emphasis, e.g. by ITU, regional and domestic authorities, and industry/research alliances, on responsible, equitable, inclusive, and collaborative AI, on data as responsibility, and on shared digital economy and social well-being and prosperity. There will indeed be a potentially significant impact on OPEX, workforce, and processes in need of proactive strategies and governance, within a collaborative and healthy global ecosystem.

7.3. Vendor Interoperability and Multi-Domain Orchestration

Achieving the full economic promise of a 6G cognitive network control depends on establishing seamless multi-vendor interoperability across highly heterogeneous infrastructure layers. Because behaviour is synthesised on demand rather than selected from static, pre-engineered procedures, the network must rely on open, standardised interfaces for capability exposure abstraction. Without strict compliance with unified data methodologies, such as those spearheaded by the AI-RAN Alliance [65], the ecosystem risks severe fragmentation.

Standardised interfaces are the only mechanism that enables distributed, multi-vendor agent swarms to safely coordinate cross-domain behaviours, exchange predictive models, and resolve runtime resource conflicts without creating vendor lock-in or compromising the underlying system stability.

Multi-agent communication and coordination was discussed earlier. Furthermore, in the context of task decomposition strategies, in general, and hierarchical decomposition, in particular, high-level orchestrator agents that oversee, coordinate, and compile the work of low-level execution and service agents were highlighted. Interoperable multi-domain orchestration has the delicate task of dynamic autonomous orchestration of synchronized data and semantics, workloads and service continuity, and controls and policies, across operational environments and boundaries, in a multi-vendor and multi-domain context [12], [66], [67].

7.4. Business Models for AI-Native Networks

Business Models for AI Native Networks (SLA 2.0 and Sensing-as-a-Service)

Monetising the transition to a 6G cognitive fabric remains a central strategic question for the telecommunications industry. Operators face a severe CapEx paradox: building an intelligent, distributed system fabric requires major upfront investment in edge compute, advanced RAN infrastructure, sensing-capable radio systems, automation platforms, and trustworthy data pipelines, while consumer willingness to pay higher premiums simply for faster data has largely stagnated since the 5G era. As a result, consumer 6G is likely to remain anchored to flat-rate or capacity-based data plans, whereas the real monetisation potential of the cognitive fabric will emerge primarily in high-value B2B markets, including enterprise networks, industrial campuses, transport hubs, ports, utilities, healthcare environments, and private network deployments.

The key commercial shift is from selling connectivity as a commodity to selling governed cyber-physical capabilities. In this model, connectivity, sensing, computing, AI inference, prediction, and control support are not billed separately as isolated technical resources but are bundled into services that deliver measurable business value. This transition enables several potential business models.

SLA 2.0: From Selling Bandwidth to Selling Outcomes

The traditional model of charging for network slices defined by static lower-layer KPIs, such as latency, throughput, availability, or packet loss, is expected to evolve toward outcome-based agreements. Instead of paying for a slice with a fixed 10 ms latency target, an enterprise may pay for a guaranteed business outcome, such as a 99.999% robotic task success rate, a bounded control stability margin, a maximum number of safety violations per hour, or a guaranteed level of information freshness in a factory floor. In this model, network KPIs remain essential, but they become contractual constraints supporting a higher-level operational objective. As slices evolve into dynamic governance envelopes, operators can charge premium tiers for the policy-bounded intelligence that proactively adjusts radio resources, compute placement, sensing updates, and control-loop parameters to prevent application-level failures during critical missions.

Sensing-as-a-Service enabled by ISAC

By embedding Integrated Sensing and Communication natively into the wireless access substrate, 6G networks can become distributed perception systems. This enables operators to monetise direct observability of the physical environment via APIs, dashboards, or data services. Potential offerings include high-resolution localisation, object detection and classification, movement tracking, blockage detection, spatial dynamics, traffic-flow mapping, and dynamic environment maps. For example, an airport authority could subscribe to an ISAC-based service for unauthorised drone detection and airside monitoring; a port operator could use it for asset tracking and safety-zone supervision; and a municipality could consume real-time traffic and crowd-flow information without deploying a dense network of conventional cameras. This model creates a new revenue stream from the network's sensing capability, provided that privacy, data ownership, accuracy, and regulatory constraints are properly governed.

AI-RAN infrastructure sharing and GPU-as-a-Service

AI-native RAN architectures may also change the economics of the base station. If radio signal processing, RAN intelligence, and AI workloads run on cloud-ready, accelerator-based platforms, the base station can increasingly behave as a distributed micro-data centre. During peak radio traffic, GPU or accelerator resources are used for radio processing, beamforming, inference, and network optimisation. During off-peak periods, part of the same edge compute capacity could be leased to hyperscalers, application providers, or local enterprises for AI inference, local model execution, or other latency-sensitive workloads. This enables models such as GPU-as-a-Service, Edge-AI-as-a-Service, or Token-as-a-Service, improving hardware utilisation and helping operators recover infrastructure investment beyond classical connectivity revenue.

In-Network Computing and Split-AI Service Models

As applications increasingly rely on real-time inference and closed-loop control, operators may monetise the coordination of communication and computing as a service. In Split-AI scenarios, different parts of an AI pipeline may run on the device, at the edge, in the RAN domain, or in a regional or cloud compute pool. The commercial value lies in guaranteeing that the combination of transport latency, compute execution time, privacy constraints, and reliability requirements meets the application deadline. Enterprises could therefore pay for managed AI execution paths, dynamic compute placement, inference confidence within a time budget, or resilience-aware execution of distributed AI workloads. This model is particularly attractive for robotics, industrial automation, XR, healthcare, and autonomous mobility, where performance depends on the joint behaviour of connectivity and computing rather than on either resource alone.

The Agentic Economy and BSS Reinvention

A more advanced business model emerges when the users of the cognitive fabric are no longer only humans or enterprise IT systems, but autonomous AI agents with bounded authority. For example, a drone fleet agent, a robotics orchestration agent, or a manufacturing control agent may dynamically request compute placement, sensing updates, reliability levels, localisation accuracy, or carbon-budget constraints. This requires a profound reinvention of Business Support Systems (BSS). Billing becomes multi-dimensional and potentially near-real-time, based not only on data volume but also on compute cycles, inference confidence, sensing quality, reliability assurance, deadline compliance, energy consumption, and carbon limits. Machine-to-machine negotiation, automated policy enforcement, auditable records, and smart-contract-like settlement mechanisms may be necessary to support these highly dynamic service relationships.

Capability exposure and API-based monetisation

Across previous models, a common enabler is the exposure of network capabilities via programmable, secure APIs. Enterprises may not buy ``a network'' in the traditional sense, but instead consume capabilities such as positioning, sensing, QoS control, compute placement, digital-twin synchronisation, anomaly prediction, or policy-bounded automation. This API-based model allows operators to package 6G functions as modular enterprise services and enables third parties to build applications on top of the network fabric. It also aligns with the broader shift from connectivity provisioning to capability platforms.

Overall, the most credible monetisation path for 6G in the mid-term is not mass-market consumer data, but B2B services where the cognitive fabric directly improves operational performance. Enterprise and non-public networks are therefore the primary proving ground for these models, because they provide controlled environments, measurable outcomes, clear responsibility boundaries, and immediate value from automation, sensing, and compute governance. The strategic challenge for operators is to translate these capabilities into trustworthy, auditable, and commercially understandable offers: outcome SLAs, sensing APIs, shared AI-RAN infrastructure, managed Split-AI execution, and agentic service platforms.

8. Discussion and Future Research Directions

8.1. Discussion

This paper stresses that agentic AI and foundation models should not be seen just as extra optimisation tools for mobile networks. Rather, it points to a larger transition: from networks that mainly ensure connectivity through predefined methods to AI-native mobile communication systems. These advanced systems can grasp intent, maintain context, reason within limits, coordinate specialised agents, and adapt their behaviour safely across communication, computing, sensing, and data resources.

This shift does not imply that all network functions should become fully autonomous without oversight or interception, nor that probabilistic AI should directly manage critical infrastructure. A key takeaway from the paper is that agentic intelligence is the native foundation enabling 6G, while it needs to work alongside telco-grade control systems. Rapid, deterministic, and safety-critical processes, especially near the PHY/MAC layer, should continue to be managed by known and enhanced mechanisms and control-theoretic principles. Agentic AI is cross-domain and policy-aware cognitive foundation that can reason over long periods, suggest adaptations, coordinate resources, and assist in decisions that are too complex for static rules alone.

Another key point is the allocation of intelligence. In 6G, agentic AI should not be confined to a single location, such as the cloud or the edge. Instead, intelligence needs to be spread across the device–edge–cloud continuum based on factors such as latency, energy, privacy, computing resources, data locality, network conditions, and service needs. This principle applies not only to model execution but also to perception, memory, planning, verification, and ongoing learning. Consequently, a multi-agent system emerges in which local, domain-level, and global agents collaborate while operating at different authority levels and timescales.

A third point is that intent-driven operation changes the role of network management. The network is no longer asked only to optimise throughput, latency, or reliability in isolation. It is increasingly asked to support higher-level outcomes such as task success, control stability, safety, information freshness, energy efficiency, and service assurance. Traditional KPIs remain essential, but they become constraints and evidence serving the final objective. This creates the metric translation need: vertical requirements must be refined into measurable targets, validated policies, admissible actions, and fulfilment reports.

The paper further contends that composable network behaviour pivots not only on intelligence but also crucially on effective governance. While agents can break down tasks, access memories, utilise tools, negotiate with others, or suggest actions, the live network must carry out only actions that are properly authorised, validated, traceable, and, if needed, reversible. Achieving this requires policy-based authority, deterministic validation processes, secure model supply chains, verified agent identities, comprehensive audit trails, transparency, decision tracking, human oversight, and fallback options. Thus, trust isn't an external feature but a fundamental runtime condition that enables agentic authority.

Finally, the operational and economic implications are significant. Agentic AI may reduce OPEX, improve resource efficiency, and enable new B2B services. Still, it also introduces costs: token consumption, model lifecycle management, distributed compute, validation overhead, security hardening, and human oversight. The value of agentic AI is therefore in how it empowers the 6G network and how it enables each use case. It is particularly justified where tasks require dynamic planning, cross-domain reasoning, uncertainty management, negotiation, or adaptation beyond what conventional automation can provide.

8.2. Path Toward Agentic 6G Networks

The development of agentic 6G networks is expected to progress in innovative phases, as it fully matures, rather than through disruptive changes. Existing 5G and 5G-Advanced systems already provide essential foundations, including cloud-native network functions, service-based architecture framework, network slicing, data analytics, automation path, and initial AI/ML-supported management. These features can advance toward enhanced observability, shared states, intent-based communication, controlled actuation, digital-twin verification, and coordination among multiple agents.

The initial step is not to hand over complete control to agents but to develop foundational architectural elements that ensure their safe and interoperable operation. These elements encompass machine-readable intent representations, capability interfaces, shared context and memory models, agent identity and attestation, policy enforcement points, decision-trace records, model provenance registries, and standardised methods for rollback and human oversight.

As these foundations develop, agentic intelligence grows from advisory and diagnostic functions to the ability to carry out actions. During this stage, agents might suggest actions, clarify intentions, coordinate tasks, optimise resource allocation, help manage energy, or activate approved control routines. Their authority should still be constrained by considerations of domain, timescale, reversibility, risk, and compliance.

This advances to enable a wider range of composable network behaviours, where intent-oriented objectives dynamically combine communication, sensing, computing, and AI functions. Nonetheless, this should still be viewed as governed autonomy rather than unrestricted independence. The network must stay auditable, adhere to policies, and remain compatible with deterministic telecom operations. In this context, a mature 6G system does not have agents replacing standards, protocols, and control mechanisms, but rather, agents functioning within them, with autonomy, while being intent driven and policy driven.

Enterprise and non-public networks are expected to be the initial practical settings for these capabilities. They offer clearer goals, controlled deployment areas, measurable business results, and stronger incentives for automation, sensing, edge intelligence, and outcome-based service models. Public networks might implement similar capabilities more slowly, particularly where interoperability, liability, regulation, and multi-vendor assurance are more complicated.

8.3. Open Research Topics

Several research questions remain open before agentic AI and foundation models can become trusted components of future mobile communication systems.

First, the translation from application goals to network actions remains unresolved. More work is needed on intent refinement, intent-oriented KPIs, policy-aware reasoning, and the mapping between vertical-level requirements and enforceable network behaviour.

Second, multi-agent coordination requires stronger foundations. Open questions include how many agents to deploy, how their roles should be specialised, how conflicts should be detected and resolved, how shared state should be maintained, and how to prevent race conditions across timescales.

Third, the integration of communication and computing requires new optimisation methods. Split-AI, in-network computing, edge inference, and distributed model execution require joint control of transport latency, compute execution time, energy consumption, privacy, and reliability.

Fourth, ISAC introduces both opportunities and overhead. Treating the network as a perception system requires efficient sensing signal processing, fusion of network and environmental context, and mechanisms to control sensing cost, power consumption, and spectrum usage.

Fifth, telco-grade safety and accountability remain central. Research is needed on deterministic validation of AI-proposed actions, formal safety envelopes, runtime verification, auditability, explainability, secure model supply chains, adversarial robustness, and human override.

Sixth, the economics of agentic AI-native networks remain to be further explored. Token-cost accounting, agentic service OPEX, infrastructure CAPEX, outcome-based SLAs, sensing-as-a-service, and agent-to-agent service negotiation require technical, operational, and commercial validation.

Finally, standardisation is essential. Common interoperable interfaces, shared data models, identity mechanisms, capability exposure, and governance primitives are essential to bring one global standard for agentic 6G mobile communication system.

9. Conclusions

Agentic AI and foundation models introduce a credible path for 6G networks to evolve beyond connectivity-centric operation. Their main contribution is not simply better automation, but a new operational model in which the network can understand intent, use memory and prediction, coordinate specialised agents, and adapt communication, computing, sensing, and AI resources toward higher-level objectives such as enabling ubiquitous connected intelligence and automated industries, and meeting Sustainable Development Goals.

This paper advocates a clear distinction between probabilistic reasoning and deterministic network control in the evolution of AI. While AI agents can interpret data, make predictions, propose actions, negotiate, and optimise, intelligently and creatively, any operational change must be validated against policy before it is applied, and must be traceable, auditable and, where necessary, reversible. Such strict boundaries are essential for integrating agentic AI with telco-grade infrastructure.

The resulting 6G architecture should therefore combine adaptive intelligence with strong governance. It should distribute intelligence across devices, edge nodes, RAN domains, core functions, cloud platforms, and management systems, for collaborative computing based on needs, resources, and conditions, while preserving interoperability, accountability, and operational control. Foundation models and Large Telco Models may provide cross-domain reasoning and semantic understanding. Still, they must be integrated through lifecycle management, model provenance, secure deployment, runtime monitoring, and fallback mechanisms.

The most likely scenario is that mobile communication systems will continue to advance capabilities such as observability, analytics, intent handling, digital twins, AI-assisted automation, and policy-driven control. Over time, these elements enable advanced multi-agent coordination and flexible network behaviours. Still, mature agentic 6G should not be seen as a fully autonomous network lacking human or deterministic oversight. Instead, it should be viewed as a governed, AI-native mobile communication system in which autonomy is granted considering factors such as risk, reversibility, compliance effects, accountability, and operational evidence. A concrete step for the community is to standardise a classification of agent actions by reversibility and by compliance impact, and the content of the decision record that accompanies each action.

The open challenge is therefore not only to continue making networks more intelligent, but to make intelligence operationally trustworthy. If this balance is achieved, agentic AI can help future mobile systems improve performance, reduce operational cost, support new service capabilities, and enable more adaptive, efficient, and sustainable 6G networks.

Acknowledgements & Endorsements

Chief Editor

Javan Erfanian (Veezhen)

Contributing Team

Narcis Cardona (iTEAM/UPV)
Giampaolo Cuozzo (WiLab/CNIT)
Lina Magoula (National & Kapodistrian University of Athens - NKUA)
Nikolas Koursioumpas (National & Kapodistrian University of Athens - NKUA)
Periklis Chatzimisios (International Hellenic University)
Eirini Kanaki (Z-RED)
Balint Varga (Karlsruhe Institute of Technology – KIT)
Mehmet Basaran (Turkcell)
Rui Aguiar (Universidade de Aveiro)
Mario Antunes (Universidade de Aveiro)
Dehao Wu (Bournemouth University)
Mohammad Shikh-Bahaei (King's College London)
George T. Karetsos (University of Thessaly)
Marouan Mizmizi (Politecnico di Milano)
R. R. Venkatesha Prasad (VP) (TU Delft)
Xueli An (Huawei)
Eirini Liotou (Harokopio University of Athens)
Susanna Schwarzmann (Huawei)

Abbreviations

3GPP	3rd Generation Partnership Project
5G	Fifth-Generation Mobile Communication System
6G	Sixth-Generation Mobile Communication System
A2A	Agent-to-Agent Protocol
ABAC	Attribute-Based Access Control
ACP	Agent Communication Protocol
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
AI-Core	AI-Agent-Based Core Network
AI-RAN	Artificial Intelligence for Radio Access Networks
AINN	Artificial Intelligence Native for Telecommunication Networks
ANP	Agent Network Protocol
AoI	Age of Information
API	Application Programming Interface
B2B	Business-to-Business
BLER	Block Error Rate
BOF	Birds of a Feather
BSS	Business Support System
CapEx	Capital Expenditure
CBF	Control Barrier Function
CPS	Cyber-Physical System
CSI	Channel State Information
DID	Decentralized Identifier / Decentralized Identity
DNS	Domain Name System
EAI	Embodied Artificial Intelligence
Edge-AI	Edge Artificial Intelligence
ENI	Experiential Networked Intelligence
ETSI	European Telecommunications Standards Institute
FG-AINN	Focus Group on Artificial Intelligence Native for Telecommunication Networks
FinOps	Financial Operations
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HARQ	Hybrid Automatic Repeat Request
HTTPS	Hypertext Transfer Protocol Secure
I-D	Internet-Draft
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IG	Introductory Guide
IID	Independent and Identically Distributed
IMT	International Mobile Telecommunications

INC	In-Network Computing
IoT	Internet of Things
IP	Internet Protocol
ISAC	Integrated Sensing and Communication
ISO	International Organization for Standardization
ITU	International Telecommunication Union
ITU-R	ITU Radiocommunication Sector
ITU-T	ITU Telecommunication Standardization Sector
JSON-RPC	JSON Remote Procedure Call
KPI	Key Performance Indicator
LLM	Large Language Model
LMM	Large Multimodal Model
LTM	Large Telco Model
MAC	Medium Access Control
MCP	Model Context Protocol
MCS	Modulation and Coding Scheme
MIMO	Multiple-Input Multiple-Output
ML	Machine Learning
NFV	Network Function Virtualization
non-IID	non-Independent and Identically Distributed
NR	New Radio
NTN	Non-Terrestrial Network
NWDAF	Network Data Analytics Function
OAM	Operations, Administration and Maintenance
OASIS	Organization for the Advancement of Structured Information Standards (OASIS Open)
OPEX	Operating Expenditure
PDU	Protocol Data Unit
PHY	Physical Layer
PKI	Public Key Infrastructure
PLMN	Public Land Mobile Network
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RBAC	Role-Based Access Control
RF	Radio Frequency
RFC	Request for Comments
RIC	RAN Intelligent Controller
RRC	Radio Resource Control
S2T	Select to Think
SBA	Service-Based Architecture
SDN	Software-Defined Networking
SDO	Standard Development Organization
SIEM	Security Information and Event Management
SLA	Service-Level Agreement
SON	Self-Organizing Network
Split-AI	Split Artificial Intelligence

TA	Tracking Area
TCP	Transmission Control Protocol
TEE	Trusted Execution Environment
TMF	TM Forum
Tok-Com	Token Communication
TR	Technical Report
TS	Technical Specification
UDP	User Datagram Protocol
UN	United Nations
URLLC	Ultra-Reliable Low-Latency Communication
XAI	Explainable Artificial Intelligence
XR	Extended Reality
ZTA	Zero-Trust Architecture

References

[1] S. Schwarzmann, R. Trivisonno, S. Lange, T. Erkilic Civelek, D. Corujo, R. Guerzoni, T. Zinner, and T. Mahmoodi, “An Intelligent User Plane to Support In-Network Computing in 6G Networks,” in Proc. IEEE International Conference on Communications (ICC), 2023, pp. 1100–1105, doi: 10.1109/ICC45041.2023.10279652.

[2] S. Schwarzmann et al., “Native Support of AI Applications in 6G Mobile Networks Via an Intelligent User Plane,” in Proc. IEEE Wireless Communications and Networking Conference (WCNC), 2024, pp. 1–6, doi: 10.1109/WCNC57260.2024.10570691.

[3] M. G. Spina et al., “In-Network Computing and Split-AI in 6G: Enablers and Proof-of-Concept Studies,” in Proc. 3rd International Conference on 6G Networking (6GNet), 2024, pp. 135–143, doi: 10.1109/6GNet63182.2024.10765651.

[4] ITU-R, Recommendation ITU-R M.2160-0: Framework and Overall Objectives of the Future Development of IMT for 2030 and Beyond, 2023.

[5] NGMN Alliance, Network Architecture Evolution Towards 6G, Version 1.0, Feb. 2025.

[6] NGMN Alliance, Cloud-Native Next Chapter – Agentic AI-Based Operating Models, Version 1.0, Mar. 2026.

[7] Khanh-Tung Tran et al., “Multi-Agent Collaboration Mechanisms: A Survey of LLMs,” arXiv:2501.06322, 2025. [Online]. Available: https://arxiv.org/abs/2501.06322

[8] Yubin Kim et al., “Towards a Science of Scaling Agent Systems,” arXiv:2512.08296, 2026. [Online]. Available: https://arxiv.org/abs/2512.08296

[9] GR ENI 060 - V4.1.1 - Experiential Networked Intelligence (ENI); Study on Model Training for Network AI Agents

[10] J. Liu et al., “Edge-Cloud Collaborative Computing on Distributed Intelligence and Model Optimization: A Survey,” IEEE Communications Surveys & Tutorials, vol. 28, pp. 5049–5080, 2026, doi: 10.1109/COMST.2026.3669216.

[11] W. Ye, Y. Zhang, X. An, G. Carle, and Y. Ma, “Select to Think: Unlocking SLM Potential with Local Sufficiency,” in Proc. 43rd International Conference on Machine Learning (ICML), 2026.

[12] SNS JU Technology Board, AI/ML as a Key Enabler of 6G Networks, 2025.

[13] Y. Yang et al., “Task-Oriented 6G Native-AI Network Architecture,” IEEE Network, 2023.

[14] B. Li, P. Qi, B. Liu, S. Di, J. Liu, J. Pei, J. Yi, and B. Zhou, “Trustworthy AI: From Principles to Practices,” ACM Computing Surveys, vol. 55, no. 9, Art. no. 177, 2023, doi: 10.1145/3555803.

[15] R. Bommasani et al., “On the Opportunities and Risks of Foundation Models,” Stanford Center for Research on Foundation Models, 2021, arXiv:2108.07258.

[16] P. Song, A. Ojo, and E. Curry, “Trustworthy Requirements for Foundation Models—A Comprehensive Survey and Roadmap,” Engineering Applications of Artificial Intelligence, vol. 163, Art. no. 113111, 2026, doi: 10.1016/j.engappai.2025.113111.

[17] 3GPP, TS 28.312, “Management and Orchestration; Intent Driven Management Services for Mobile Networks,” Version 19.5.0, Release 19, 2026.

[18] J. S. Park, J. O’Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, “Generative Agents: Interactive Simulacra of Human Behaviour,” in Proc. 36th Annual ACM Symposium on User Interface Software and Technology (UIST), 2023, Art. no. 2, pp. 1–22, doi: 10.1145/3586183.3606763.

[19] T. R. Sumers, S. Yao, K. Narasimhan, and T. L. Griffiths, “Cognitive Architectures for Language Agents,” Transactions on Machine Learning Research, 2024.

[20] S. Yan, X. Yang, Z. Huang, E. Nie, Z. Ding, Z. Li, X. Ma, J. Bi, K. Kersting, J. Z. Pan, H. Schütze, V. Tresp, and Y. Ma, “Memory-R1: Enhancing Large Language Model Agents to Manage and Utilize Memories via Reinforcement Learning,” in Proc. 64th Annual Meeting of the Association for Computational Linguistics (ACL), 2026, pp. 12805–12825, doi: 10.18653/v1/2026.acl-long.583.

[21] S. Yan, A. Bahloul, E. Nie, S. Schwarzmann, R. Trivisonno, V. Tresp, and Y. Ma, “Memory-R2: Fair Credit Assignment for Long-Horizon Memory-Augmented LLM Agents,” arXiv:2605.21768, 2026.

[22] C. Packer, S. Wooders, K. Lin, V. Fang, S. G. Patil, I. Stoica, and J. E. Gonzalez, “MemGPT: Towards LLMs as Operating Systems,” arXiv:2310.08560, 2023.

[23] P. Chhikara, D. Khant, S. Aryan, T. Singh, and D. Yadav, “Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory,” arXiv:2504.19413, 2025.

[24] K. Mehmood, K. Kralevska, and D. Palma, “Knowledge Graph Embedding in Intent-Based Networking,” arXiv:2405.07850, 2024.

[25] G. M. van de Ven, N. Soures, and D. Kudithipudi, “Continual Learning and Catastrophic Forgetting,” arXiv:2403.05175, 2024.

[26] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, “Retrieval-Augmented Generation for Large Language Models: A Survey,” arXiv:2312.10997, 2023.

[27] X. Wu, H. Li, and F. Khomh, “On the Effectiveness of Log Representation for Log-Based Anomaly Detection,” arXiv:2308.08736, 2023.

[28] I. Chatzistefanidis, N. Nikaein, A. Leone, A. Maatouk, L. Tassiulas, R. Morabito, I. Pitsiorlas, and M. Kountouris, “AGORAN: An Agentic Open Marketplace for 6G RAN Automation,” Computer Networks, vol. 275, Art. no. 111927, 2026, doi: 10.1016/j.comnet.2025.111927.

[29] ETSI, Experiential Networked Intelligence (ENI); Study on AI Agents Based Next-Generation Network Slicing, ETSI GR ENI 051 V4.1.1, Feb. 2025. [Online]. Available: https://www.etsi.org/deliver/etsi_gr/ENI/001_099/051/04.01.01_60/gr_ENI051v040101p.pdf

[30] ETSI, Experiential Networked Intelligence (ENI); Study on Multi-Agent Frameworks for Next-Generation Core Networks, ETSI GR ENI 056 V4.1.1, Oct. 2025. [Online]. Available: https://docbox.etsi.org/ISG/ENI/Open/ENI056%20(Release%204.1)/gr_eni056v040101p.docx

[31] R. Surapaneni, M. Jha, M. Vakoc, and T. Segal, “Announcing the Agent2Agent Protocol (A2A): A New Era of Agent Interoperability,” Google Developers Blog, Apr. 9, 2025. [Online]. Available: https://developers.googleblog.com/a2a-a-new-era-of-agent-interoperability/

[32] Z. Liu, Z. Yang, Z. Zhang, Q. Qi, and M. Shikh-Bahaei, “Multi-Agent Reinforcement Learning for Diffusion-Based Token Communications in Robotic Networks,” to appear in IEEE Transactions on Cognitive Communications and Networking, 2026, doi: 10.1109/TCCN.2026.3712126.

[33] X. Ma, O. Ayan, Y. Ma, and X. An, “Customized User Plane Processing via Code Generating AI Agents for Next Generation Mobile Networks,” in Proc. IEEE International Conference on Communications (ICC), 2026, pp. 1–6.

[34] W. Xu et al., “Deploying Foundation Model Powered Agent Services: A Survey,” IEEE Communications Surveys & Tutorials, vol. 28, pp. 1483–1519, 2026, doi: 10.1109/COMST.2025.3580745.

[35] S. Liu, W. Wu, X. Xu, T. Li, B. Pang, B. Guo, and Z. Yu, “Adaptive and Resource-Efficient Agentic AI Systems for Mobile and Embedded Devices: A Survey,” arXiv:2510.00078, 2025.

[36] 3GPP, TS 38.211, “NR; Physical Channels and Modulation”; TS 38.213, “NR; Physical Layer Procedures for Control”; and TS 38.214, “NR; Physical Layer Procedures for Data.”

[37] “Overhead Reduction of NR Type II CSI for NR Release 16,” in Proc. 23rd International ITG Workshop on Smart Antennas (WSA), 2019.

[38] “Large Wireless Model (LWM),” arXiv:2411.08872, 2024; and “WiFo-CF: Wireless Foundation Model for CSI Feedback,” arXiv:2508.04068, 2025.

[39] 3GPP, TR 38.843, Study on Artificial Intelligence (AI)/Machine Learning (ML) for NR Air Interface, Release 18; RP-234039, Release 19 Work Item on AI/ML for the NR Air Interface; and NR_AIML_air_Ph2, Release 20 Work Item on Two-Sided CSI Compression.

[40] E. Tabassi, Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1, National Institute of Standards and Technology, 2023, doi: 10.6028/NIST.AI.100-1.

[41] LF AI & Data Generative AI Commons, The Model Openness Framework (MOF) Specification, Dec. 2024.

[42] C. Cesarano and M. Monperrus, “The Grand Software Supply Chain of AI Systems,” arXiv:2604.27781, 2026.

[43] European Union Agency for Cybersecurity (ENISA), AI Cybersecurity Challenges: Threat Landscape for Artificial Intelligence, Dec. 2020, doi: 10.2824/238222.

[44] OWASP Foundation, OWASP Top 10 for LLM Applications 2025, 2025.

[45] M. Goldblum et al., “Dataset Security for Machine Learning: Data Poisoning, Backdoor Attacks, and Defenses,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 2, pp. 1563–1580, 2023, doi: 10.1109/TPAMI.2022.3162397.

[46] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, “Not What You’ve Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection,” arXiv:2302.12173, 2023.

[47] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, “Membership Inference Attacks on Machine Learning: A Survey,” arXiv:2103.07853, 2021.

[48] J. Cohen, E. Rosenfeld, and J. Z. Kolter, “Certified Adversarial Robustness via Randomized Smoothing,” arXiv:1902.02918, 2019.

[49] B. Hussain, M. Bilal, T. Li, H. Pervaiz, X. Tang, Q. Du, F. Ahmad, M. Azhar, and J. Zhang, “AI-Native Closed-Loop Security for 6G-Enabled Cyber-Physical Systems: From Edge Detection to Network-Wide Mitigation,” arXiv:2606.08173, 2026.

[50] I. D. Raji, A. Smart, R. N. White, M. Mitchell, T. Gebru, B. Hutchinson, J. Smith-Loud, D. Theron, and P. Barnes, “Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing,” in Proc. 2020 Conference on Fairness, Accountability, and Transparency (FAT* ’20), Barcelona, Spain, 2020, pp. 33–44, doi: 10.1145/3351095.3372873.

[51] S. Wang, M. A. Qureshi, L. Miralles-Pechuán, T. Huynh-The, T. R. Gadekallu, and M. Liyanage, “Explainable AI for 6G Use Cases: Technical Aspects and Research Challenges,” IEEE Open Journal of the Communications Society, vol. 5, pp. 2490–2540, 2024, doi: 10.1109/OJCOMS.2024.3386872.

[52] E. Kanaki, A. G. Hessami, P. Gonçalves, and P. Chatzimisios, “Ethical and Privacy Issues for AI in 6G Empowering Robotics,” IEEE Communications Standards Magazine, vol. 10, no. 2, pp. 296–304, Jun. 2026, doi: 10.1109/MCOMSTD.2026.3658562.

[53] Y. Rong, T. Leemann, T.-T. Nguyen, L. Fiedler, P. Qian, V. Unhelkar, T. Seidel, G. Kasneci, and E. Kasneci, “Towards Human-Centered Explainable AI: A Survey of User Studies for Model Explanations,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 4, pp. 2104–2122, Apr. 2024, doi: 10.1109/TPAMI.2023.3331846.

[54] N. Díaz-Rodríguez, J. Del Ser, M. Coeckelbergh, M. López de Prado, E. Herrera-Viedma, and F. Herrera, “Connecting the Dots in Trustworthy Artificial Intelligence: From AI Principles, Ethics, and Key Requirements to Responsible AI Systems and Regulation,” Information Fusion, vol. 99, Art. no. 101896, 2023, doi: 10.1016/j.inffus.2023.101896.

[55] S. Kumar, S. Datta, V. Singh, D. Datta, S. K. Singh, and R. Sharma, “Applications, Challenges, and Future Directions of Human-in-the-Loop Learning,” IEEE Access, vol. 12, pp. 75735–75760, 2024, doi: 10.1109/ACCESS.2024.3401547.

[56] V. Chamola, V. Hassija, A. R. Sulthana, D. Ghosh, D. Dhingra, and B. Sikdar, “A Review of Trustworthy and Explainable Artificial Intelligence (XAI),” IEEE Access, vol. 11, pp. 78994–79015, 2023, doi: 10.1109/ACCESS.2023.3294569.

[57] L. Methnani, A. Aler Tubella, V. Dignum, and A. Theodorou, “Let Me Take Over: Variable Autonomy for Meaningful Human Control,” Frontiers in Artificial Intelligence, vol. 4, Art. no. 737072, 2021, doi: 10.3389/frai.2021.737072.

[58] E. Papagiannidis, P. Mikalef, and K. Conboy, “Responsible Artificial Intelligence Governance: A Review and Research Framework,” Journal of Strategic Information Systems, vol. 34, no. 2, Art. no. 101885, Jun. 2025, doi: 10.1016/j.jsis.2024.101885.

[59] ETSI, Experiential Networked Intelligence (ENI) ISG, “Use Cases and Requirements for AI Agents Based Core Network,” ETSI GR ENI 055 V4.1.1, Oct. 2025.

[60] Y. Chen, J. Chen, C. He, Y. Li, Y. Ji, Y. Wu, D. Yang, L. Diao, L. Shou, H. Zhang, H. Li, and G. Chen, “Token Economics for LLM Agents: A Dual-View Study from Computing and Economics,” arXiv:2605.09104, 2026.

[61] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, “From Network Sharing to Multi-Tenancy: The 5G Network Slice Broker,” IEEE Communications Magazine, vol. 54, no. 7, pp. 32–39, 2016.

[62] L. Yang, J. Luo, X. Liu, Y. Lou, and Z. Chen, “BAMAS: Structuring Budget-Aware Multi-Agent Systems,” arXiv:2511.21572, 2025.

[63] R. Sapkota, K. I. Roumeliotis, and M. Karkee, “AI Agents vs. Agentic AI: A Conceptual Taxonomy, Applications and Challenges,” arXiv:2505.10468, 2025.

[64] B. Wu, S. Wang, Y. Liu, Y.-Q. Zhang, J. Sifakis, and Y. Ouyang, “From Automated to Autonomous: Hierarchical Agent-Native Network Architecture (HANA),” arXiv:2605.20608, 2026.

[65] AI-RAN Alliance Task Group Data-for-AI, “Data Frameworks and Methodologies for AI Model Training and Inference in RAN Optimization,” AI-RAN Alliance Technical Report, 2025.

[66] Y. Tong, A. Sun, A. Wang, Y. Liu, and Y. Xie, “Network AI Agent Use Cases and Requirements in 6G,” Internet-Draft draft-tong-network-agent-use-cases-in-6g-00, Work in Progress, Nov. 2025.

[67] A. Adimulam, R. Gupta, and S. Kumar, “The Orchestration of Multi-Agent Systems: Architectures, Protocols, and Enterprise Adoption,” arXiv:2601.13671, 2026.

[68] 3GPP, TR 22.870, “Study on 6G Use Cases and Service Requirements.”

[69] 3GPP, TR 29.832, “Study on the Protocol for Artificial Intelligence in 6G.”

[70] 3GPP, TR 23.801-01, “Study on Architecture for 6G System; Stage 2.”

[71] ITU-T Focus Group on AI Native for Telecommunication Networks (FG-AINN), “Standardization Gap Analysis of the FG-AINN,” FG-AINN-O-024, May 2026. https://www.itu.int/en/ITU-T/focusgroups/ainn/Documents/Standardization_Gap_Analysis.pdf

[72] Experiential Networked Intelligence (ENI), “Study on Model Training for Network AI Agents”, ETSI GR ENI 060 V4.1.1, 2026-08.

[73] 3GPP, TR 33.801-01, “Study on Security for the 6G system,” Release 20. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=4449