Network Management
Internet-Draft
Intended status: Informational
Expires: 28 August 2026

Q. Wu
Huawei
C. Zhou
China Mobile
L. M. Contreras
Telefonica
S. Han
China Unicom
Y. Hong
Daejeon University
24 February 2026

Network Digital Twin and Agentic AI based Architecture for AI driven Network Operations
draft-wmz-nmrg-agent-ndt-arch-03

Abstract

A Network Digital Twin (NDT) provides a network emulation tool usable for different purposes such as scenario planning, impact analysis, and change management. Integrating a Network Digital Twin into network management together with Agentic AI, it allows the network management activities to take user intent or service requirements as input, automatically assess, model, and refine optimization strategies under realistic conditions but in a risk-free environment. Such environment that operates to meet these types of requirements is said to have AI driven Network Operations.

AI driven Network Operations brings together existing technologies such as Agentic AI and Network Digital Twin which may be seen as the use of a toolbox of existing components enhanced with a few new elements.

This document describes an architecture for AI driven network operations and shows how these components work together with network digital twin and Agentic AI capabilities. It provides a cookbook of existing technologies to satisfy the architecture and realize intent-based network management to meet the needs of the network service.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Network Management mailing list (nmrg@irtf.org), which is archived at https://mailarchive.ietf.org/arch/browse/nmrg.

Source for this draft and an issue tracker can be found at https://github.com/QiufangMa/Agent-architecture.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months

and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 28 August 2026.

Table of Contents

1. Introduction

The rapid expansion of network scale and the increasing demands on these networks necessitate of continuous network reconfiguration to better adapt to ever-changing service requirements.

Since network changes are directly related to service operations, any successful change needs to not only ensure that new services are provisioned smoothly, but also that existing services are not affected and that no problems are introduced with the new configurations. Network operators are, therefore, increasingly cautious about making network changes, given that they need to review the solution design as well as evaluate all change impacts, before making any change. Then, after the change, they need to perform dialing tests, monitor traffic, and manually check table entries.

The Network Digital Twin (NDT) [I-D.irtf-nmrg-network-digital-twin-arch] has been proposed as a mean to provide a network emulation tool for scenario planning, impact analysis, and change management. Agentic AI introduces disruptive paradigm to the network management and allow declarative intent interpretation, multi-step action, multi-agent coordination. Integrating a Network Digital Twin into network management together with Agentic AI, it allows network management activities to dynamically adapt to customer needs, network changes, as well as to automatically assess, model, and refine optimization strategies under realistic conditions but in a risk-free environment. An environment that operates to meet these types of requirements is said to have AI driven network operations.

AI Driven network operations provide the following capabilities to applications by coordinating the components that operate and manage the network:

* Service intent and service assurance work together to ensure that the network change or network optimization aligns with business goals and that the services provided meet the agreed-upon Service Level Agreements (SLAs).

* Provide network capacity planning and ensure that the network has sufficient capacity , resources, and infrastructure to meet current and future demands.

* Provide simulation on fault scenarios, formulate recovery plans, and verify whether the plans are applicable and effective so that the service will not be affected during disaster recovery drill.

* Support fault and risk detection and provide network health check and network risk check.

* Model the network configuration change and use a virtual topology model to test network changes and assess the effect of the network configuration changes on the network.

* Model the protocol operations and interactions among devices in the network and simulate specific networking protocols such as IS-IS, OSPF, BGP, SR, etc to understand how they perform under different conditions.

* Model traffic flow across the network, including traffic generation, flow control, routing, and congestion control and evaluate traffic’s impact on network performance.

* Support generation of rectification solutions for potential network risks and provide verification on the repair solution in seconds, including loop, address conflict, and security policy conflict.

This document describes an architecture for AI Driven network operations, showing how these components work together with network digital and AI capabilities. It provides a cookbook of existing technologies to satisfy the architecture and realize intent-based networking to meet the needs of applications.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The document uses the following definitions and acronyms defined in [I-D.irtf-nmrg-network-digital-twin-arch]:

* Network Digital Twin (NDT)

* Artificial Intelligence (AI)

The following acronyms are used throughout this document:

* Generative Artificial Intelligence (Gen-AI)

* Large Language Model (LLM)

* Retrieval-Augmented Generation (RAG)

* Agentic AI [I-D.hong-nmrg-agenticai-ps]

* Multiple Agent System (MAS)

* Remote Code Execution (RCE)

Besides, this document defines the following terminology:

Network AI Agent: Network AI Agent is an autonomous system or entity with awareness of its environment, capable of conducting analysis, making decisions, and executing actions with specific intent based on its knowledge representation to achieve a set of service goals [TMF-1251D]. In addition, it is able of planning the tasks and decompose the tasks into several sub-tasks and coordinate with Task agent for these sub-tasks.

Task AI Agent: Task AI Agent is responsible for coordinating with Network AI Agent in the multi-Agent System and executing specific task assigned by Network AI Agent.

3. Introduction of Concepts

3.1. Generative AI and Agentic AI

The integration of AI into network operations has marked a significant leap forward in the pursuit of network automation and intelligence, while generative AI further enhances the role of AI driven network operations and management. Generative AI is a subfield of AI that uses generative models such as Large Language Models (LLMs) to generate new and original content such as text, images, videos, or other forms of data with the capability to adapt and make decisions to achieve specific goals.

Agentic AI refers to the broader category of AI systems that exhibit "agency"—the ability to act independently and iteratively to perform tasks without constant human prompting [I-D.hong-nmrg-agenticai-ps]. In the context of network operations and management, Network AI agents are increasingly being designed to interact with physical world and act upon it based on tools [Google-Agents-Whitepaper] and perform network management tasks such as understanding user intent, generating network configurations, diagnosing and resolving network incidents [I-D.ietf-nmop-network-incident-yang]. Meanwhile, other SDOs also try to define terms related to Network AI agent in the context of network operations and management, e.g., TM Forum defines Autonomous Agent in [TMF-1251D] as one of AN (Autonomous Network) Terminologies.

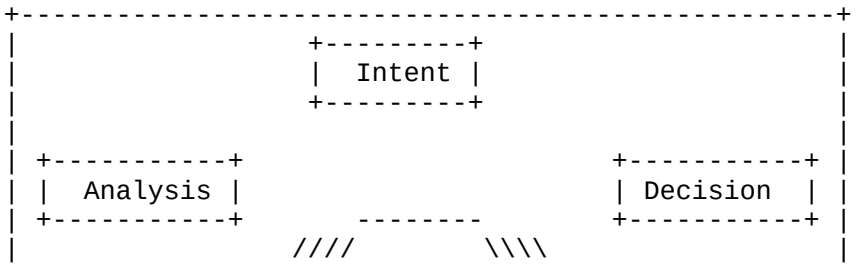
3.2. Network Digital Twin

The Network Digital Twin is a digital representation that is used in the context of network. The concept and architecture of the Network Digital Twin are specified in [I-D.irtf-nmrg-network-digital-twin-arch]. Three core functional components which includes Data Repository component, a Service Mapping Models component, and an NDT Management component are introduced to characterize the Network Digital Twin and its reference architecture.

The Network Digital Twin is widely recognized to be useful as an advanced platform for network emulation, serving as a tool for scenario planning, impact analysis, and change management. By delivering applications requests to the Network Digital Twin through standardized interfaces (see Section 9.4 of [I-D.irtf-nmrg-network-digital-twin-arch]), the Network Digital Twin exposes the various capabilities to network applications.

4. Characteristics of AI driven Network Operations

AIOPS was first defined by Gartner in 2016, combining "artificial intelligence" and "IT operations" to describe the application of AI and machine learning to enhance IT operations. However there is no unified definition for characteristic of "AI driven network operations" within the networking industry. Referring to the characteristics of AIOPS in IT field and the characteristics of networking itself, this document introduces six key elements (i.e., awareness, decision, analysis, execution, intent and knowledge) to characterize the AI driven network operation and its use, as shown in Figure 1. They together form a close-loop of network operation and management.



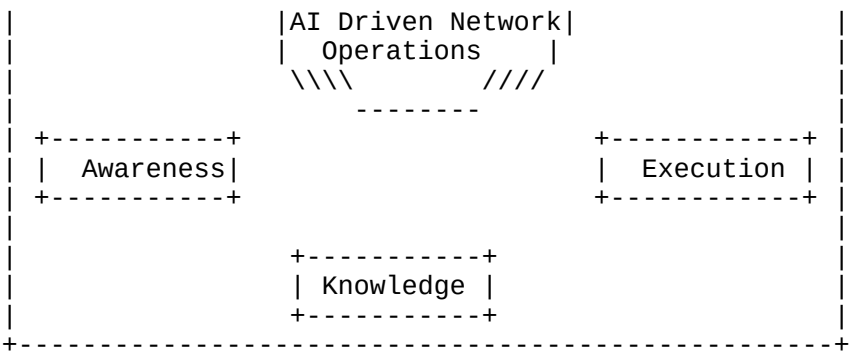


Figure 1: Six Key Elements to Characterize AI driven network operation

* Intent: Intent is defined as a set of operational goals and outcomes defined in a declarative manner without specifying how to achieve or implement them in [RFC9315]. The Network AI Agent must accurately interpret and understand the user's high-level business or operational objectives, this involves translating declarative requirements into specific network instructions, e.g., configurations.

* Knowledge: The Network AI agent relies on a knowledge base that includes network policies, historical data, expert experience, extra-system experience (updates to LLMs/their implied 'knowledge bases') and Manually or semi-manually entered knowledge, e.g., new equipment spec sheets,best practices in product manual. The knowledge is used to inform its analysis, decision-making, and execution processes. Over time, the Network AI agent can expand its knowledge through machine learning, incorporating new data and experiences to improve its performance. For example, it learns which configurations are optimal for specific scenarios or how to respond most effectively to particular types of network incidents [I-D.ietf-nmop-network-incident-yang].

* Analysis: The Network AI agent continuously analyzes vast amounts of network data from various sources, including network telemetry [RFC9232] and external feeds, and identify the gap between user intent and the existing network status. By integrating Network digital twin [I-D.irtf-nmrg-network-digital-twin-arch] with Network AI agent and leveraging machine learning and other data analytics techniques, it also identifies network fault, problem, incident, anomaly and perform data driven intelligent analysis such as service impact analysis, and so on. Their distinction is further discussed in [I-D.ietf-nmop-terminology].

* Decision: Based on the intent and network analysis, AI makes informed decisions. By integrating network digital twin [I-D.irtf-nmrg-network-digital-twin-arch] and AI, the intelligence decisions making can be realized. These decisions could involve dynamically adjusting network parameters, e.g., rerouting traffic to avoid congestion. The decision-making process is driven by predefined policies, real-time data analysis, and AI models (e.g., LLMs) that enable the Network AI agent to choose the best course of action to meet the specified intent. Network AI agent may also verify the correctness of the decision outcome by performing some network simulation or validation process.

* Awareness: Awareness is achieved through real-time monitoring and data collection. The Network AI agent maintains a comprehensive visibility of the network, enabling it to make

context-aware decisions. Network operators can also use the awareness understand the exact cause of specific network issues and achieve closed-loop decision-making.

* Execution: Once a decision is made, the Network AI agent executes the necessary actions to implement it. This could involve, e.g., sending configuration to network controllers or network devices through NETCONF/RESTCONF protocols. The execution is carried out in a controlled and precise manner to ensure that the network behaves as intended without causing disruptions. The Network AI agent also verifies that the executed actions have the desired effect and makes the proper adjustments if needed.

5. Architecture Design

5.1. Overall Architecture

Figure 2 provides the overall architecture for integrating Network Digital Twin and Network AI Agent System. The components and functional interfaces are discussed in Section 5.2 and Section 5.3, respectively. The use cases described in Section 9 show how different components are used selectively to provide different services. It is important to understand that the relationships and interfaces shown between components in this figure are illustrative of some of the common or likely interactions; however, this figure does not preclude other interfaces and relationships as necessary to realize specific functionality.

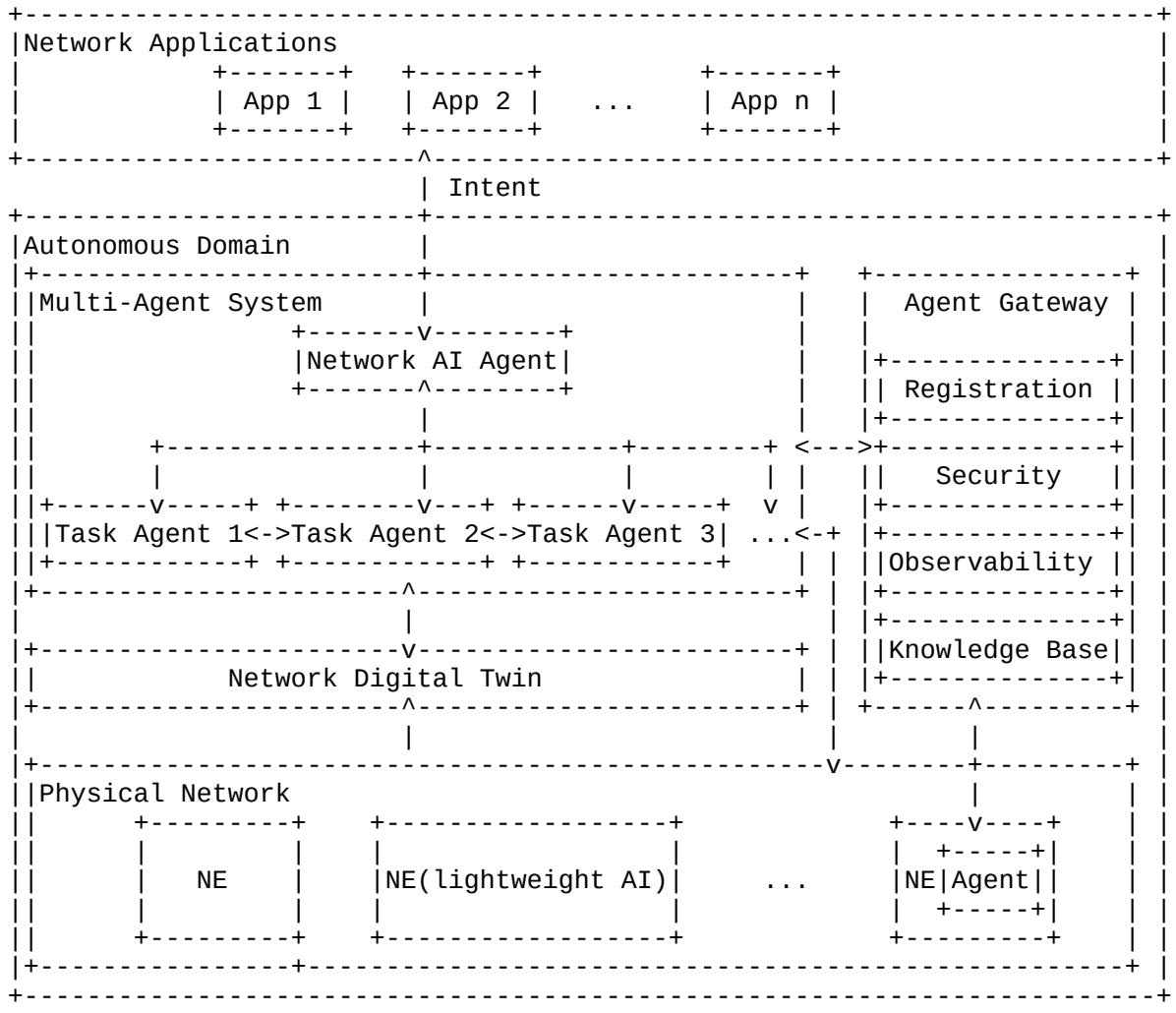


Figure 2: An Architecture for Integrating Network AI Agent with

5.2. Functional Components

This section describes the functional components shown as boxes in Figure 2. The interactions between those components, the functional interfaces, are described in Section 5.3.

5.2.1. Network Applications

Various network applications at the service level can effectively run over a AI driven Network operation platform to implement either conventional or innovative network operations, with low cost and less service impact on real networks.

A network application may be a software tool that a user uses to make requests to the network to set up specific services such as end-to-end connections or scheduled bandwidth reservations or NOC Application /Service AI Agent Application that is responsible for monitoring, managing, and maintaining the health, performance, and availability of complex networks.

Network applications make requests that need to be addressed by the AI driven network. Such requests are exchanged through a northbound intent interface (e.g., Restful API, Natural Language Programming Interface(NLPI),A2A, A2A-T), so that they can be applied by multi-agent system at the appropriate twin instance(s).

5.2.2. Autonomous Domain

An autonomous domain is a self-governing network that integrates NDT and AI driven capabilities to achieve autonomous network management. It comprises the following sub-components.

5.2.2.1. Multi-Agent System

Multi-Agent system acts as the smart brain of the Autonomous Domain, which is responsible for conducting AI-based analysis and making decisions regarding network management operations. It usually comprises a Network AI Agent and one or multiple task agents.

The Network AI Agent coordinates cross-task-agent collaboration, aligns tasks with user intent, and supervises the task execution of each task agent. And task agents are designed to perform specific functionalities, they could be scenario-oriented and classified according to the function they perform. Task Agents can adapt to new circumstances through access to evolving knowledge and reasoning, planning. It leverages the inference of LLM, the simulation of Network Digital Twin, and the contextual and domain-specific knowledge provided by Knowledge Base to accomplish specific network operation task. Some ongoing efforts (MCP [MCP], A2A [A2A]) in the industry may help with multi-agents coordination.

5.2.2.2. Agent Gateway

The Agent Gateway, which serves as a central management hub, provides essential services for the Multi-Agent System, including agent registration/discovery, authentication, observability, and knowledge base.

5.2.2.2.1. Registration

AI Agents need to first discover each other and understand their capabilities to collaborate. Agent Registration manages the process by which new agents could join the system, making them discoverable and available. It supports the unified registration of all AI agents

within the autonomous domain, including those residing on network devices. Each Agent instance submits its own metadata information including URI, supported authentication methods, and capabilities to the Agent Registry. And the consumer Agent (e.g., the Network AI Agent or task agent) could query or subscribe to the Agent Registry to find appropriate Agents for task execution.

Agent skills [Agent-skills], introduced by Anthropic, provides a new way for Agents to improve how they perform specific tasks through folds that include instructions, scripts, and resources that are only loaded when needed. Skills are folds containing a "skill.md" file, Registration component enables Agents to query directories for required skills.

[A2A] implements Agent Registration by providing the Agent Card mechanism to ensure Agents from different vendors can register and discover other Agents they need.

5.2.2.2.2. Security

The security component enforces trusted inter-Agent communication by verifying the identity of AI Agents and enforcing security policies throughout their interaction. It provides unified security functionalities for all AI Agents within the autonomous domain, including those residing on network devices. Some existing authentication methods such as OAuth 2.0, allow to issue each AI Agent its own authentication credentials to establish trusted communication.

Standardized protocols like TLS (Transport Level Security) could be leveraged to protect sensitive data exchanged between AI Agents.

It is also worth noting that once authenticated, authorization defines the specific tools and data an agent can access, which often using a Least Privilege access control method. It is also recommended to log every Agent decision and tooling call to maintain audit trail.

5.2.2.3. Observability

Observability component provides unified monitoring capabilities for all AI agents within the autonomous domain and enable network operators to gain deep insights into Agent behaviors. It collects logs, metrics and traces for each Agent and provides end-to-end visibility into task progress, failures, and network performance such as latency.

In addition, it can also make sure every action is governed by declarative policy, logged, and traceable for operational integrity, e.g., it can discern whether a human-in-the-loop approved an action or if the agent acted autonomously.

5.2.2.3.1. Knowledge Base

The Knowledge Base serves as a crucial repository of information within the architecture. It enables the injection of expert knowledge and chain of thoughts, provides the necessary knowledge and memory that helps Agents make more accurate and practive context-aware decisions. It also helps mitigate the hallucination problems that can arise in large-scale models, which enhances the accuracy of task execution. Additionally, the Knowledge Base plays a key role in providing the data needed for techniques like Retrieval-Augmented Generation (RAG), which further boosts the system’s ability to generate reliable and relevant outputs.

In case of coupling MCP [MCP] with the network management system, the

new knowledge also can be used to support modification of the currently operating automation Closed Loop, such as: - Choice of tools (data, analytics, algorithms/decision processes, closed loops) - Orchestration of tools

5.2.2.4. Network Digital Twin

A Network Digital Twin provides an enhanced and optimized solution in the face of increasing network and business types, scale, and complexity. It simulates the behavior, performance, and characteristics of the actual network, which could help in validation and testing scenarios, analyzing and predicting network behavior without affecting the real physical network.

As described in Section 7 of [I-D.irtf-nmrg-network-digital-twin-arch], the core functional components of an Network Digital Twin includes Data Repository, Service Mapping Models, and a Network Digital Twin Management component. The Network Digital Twin collects the real-time operational and instrumentation data from network through the appropriate real network-facing input interfaces, and it delivers NDT services through appropriate application-facing output interfaces, which is the interfaces to Network AI Agent(s) in Figure 2.

5.2.2.5. Physical Network

This is the actual hardware and infrastructure that makes up the network, which includes a set of network devices and wiring. In a physical network, Network Elements (NEs) with Lightweight AI [I-D.irtf-nmrg-ai-challenges] or AI Agent may also achieve some local close loop without relying on human intervention. It is also possible for Lightweight AI or AI Agent to coordinate with other AI Agent(s) to enhance the automation and efficiency of network operations. The Network Lightweight AI models could be trained, validated, deployed, and executed on Network Elements, and further refined (e.g., model re-training) through monitoring and continuous optimization based on feedback from LLM.

5.3. Functional Interfaces

This section describes the interfaces between functional components that might be externalized in an implementation allowing the components to be distributed across platforms. Where existing protocols might provide all or most of the necessary capabilities, they are noted.

As noted in Section 5.1, it is important to understand that the relationships and interfaces shown between components in Figure 2 are illustrative of some of the common or likely interactions; however, this figure and the descriptions in the subsections below do not preclude other interfaces and relationships as necessary to realize specific functionality. Thus, some of the interfaces described below might not be visible as specific relationships in Figure 2, but they can nevertheless exist.

5.3.1. Human in the Loop

The architecture allows human experts to monitor, guide, approve, or intervene in the AI driven network operations. Human may provide guidance and make critical decisions when necessary. By involving human in the process, the architecture can leverage their insights and experience, ensuring AI actions align with organizational goals.

Human in the loop is also helpful to provide a safeguard for complex or sensitive decisions, where human judgement is essential to avoid potential errors or ethical dilemmas.

This typically uses natural language as the primary mode of interaction, a chat platform that allows for conversational interaction with AI Agents can be leveraged. In some scenarios, operators may use structured format for strategy injection via workflows. Protocols like A2A [A2A], and RESTful API can be leveraged.

5.3.2. Application to Network AI Agent Interface

Intent based Network Management helps in delivering application requests to the AI Driven network operation platform and exposing the various platform capabilities to network applications.

Standardized protocols and interfaces facilitate smooth communication between applications and AI driven network operation platform and ensures different systems from various vendors can work together seamlessly. The interfaces between Network applications and Network AI Agent can adopt IG1453 Agent to Agent Protocol for Telecoms (A2A-T) [A2A-T] specified by TM Forum.

5.3.3. Network AI Agent to Task AI Agent Interface (Single Autonomous Domain)

This interface governs the coordination and task delegation within the Multi-Agent System of a single Autonomous Domain. The Network AI Agent, acting as the principal coordinator, uses this interface to decompose high-level goals into specific tasks and assign them to specialized Task Agents (e.g., for configuration generation or fault diagnosis). It facilitates communication for task assignment, progress monitoring, and result aggregation. This coordination can be implemented using protocols like [A2A-T].

5.3.4. Network AI Agent to Network AI Agent Interface (Cross Autonomous Domain)

This interface enables collaboration and information exchange between Network AI Agents residing in different Autonomous Domains. It is essential for scenarios requiring end-to-end service assurance or coordinated optimization across multi-domain networks. Through this interface, Network AI Agents can negotiate resource allocation, share summarized domain-specific insights (while preserving detail isolation for privacy and scalability), and coordinate actions to fulfill cross-domain objectives. Standardized protocols like A2A-T [A2A-T], designed for agent interoperability in telecommunication area, are candidate technologies for implementing this cross-domain interface, ensuring secure and reliable interaction between autonomous systems from different administrative domains.

5.3.5. Network AI Agent/Task AI Agent to Agent Gateway Interface

The interface between Multi-Agent System and Agent Gateway serves as the management bridge which encompasses a set of services designed to manage the lifecycle, security, and collaborative capabilities of the AI Agents.

Registration handles Agent onboarding, lifecycle tracking (e.g., heartbeat monitoring, status updates), and capability-based Agent discovery. Interfaces like RESTful APIs with structural schema for AI Agents metadata description could be leveraged. Protocols like A2A [A2A] Agent card mechanism may also be used to ensure interoperability among different Agent vendors. It is also worth noting that message queue mechanisms such as Kafka could also be a candidate interface for asynchronous communications for agent registration and discovery.

Authentication ensures trusted inter-Agent communication by verifying the identity of AI Agents and enforcing security policies throughout their interaction. Protocols like Transport Layer Security (TLS) could be leveraged for in-transit data Protection. While OAuth 2.0 and OpenID Connect are increasingly used to authenticate AI Agents.

The interface between AI Agent and Knowledge Base is specified in Section 5.3.7.

5.3.6. Network AI Agent to Network Digital Twin Interface

The interface between Multi-Agent System and Network Digital Twin are the application-facing interface as defined in [I-D.irtf-nmrg-network-digital-twin-arch]. Furthermore, the Model Context Protocol (MCP) [MCP] can be leveraged to standardize this interaction, enabling the NDT to expose its simulation and analysis capabilities as a set of discoverable "tools" that the AI Agent can dynamically invoke. This MCP-based approach facilitates seamless integration and richer contextual exchange between the Agent and the NDT.

5.3.7. Network AI Agent to Knowledge Base Interface

Knowledge Base service provides contextual data and insights to enhance the decision-making accuracy of the Multi-Agent System.

Interfaces such as Cypher or SPARQL with schema-defined data models (e.g., LPG or RDF for knowledge representation) allow efficient retrieval and updates. Other high-throughput interfaces such as gRPC or RESTful API can be the candidate for synchronous semantic search queries. For large-scale knowledge operations, asynchronous data message systems (e.g., Kafka) can also be employed for data ingestion and real-time knowledge synchronization across distributed Agents.

Additionally, the Model Context Protocol (MCP) [MCP] could also serve as a standardized interface for AI Agents to dynamically access and utilize a wide range of tools and data sources provided by the Knowledge Base. It enables the Knowledge Base to expose contextual information, expert rules, and external data as "tools" that Agents can invoke, significantly enhancing their reasoning and problem-solving capabilities.

5.3.8. Task AI Agent to Physical Network Interface

5.3.8.1. Data Collection

Data Collection interface is responsible for gathering data from the physical network through various different tools and methods (e.g., IPFIX [RFC7011], YANG-push [RFC8639],[RFC8641], BMP [RFC7854], and MCP [MCP]). It collects various types of network data including configuration data, operational data, network topology, routing data, logs, and trace on management plane, control plane, and forwarding plane as needed. The collected data is fed into the Network Digital Twin and Network AI Agent(s) to provide with up-to-date information about the current state of the physical network.

5.3.8.2. Configuration

Once network decisions are made and confirmed, the Multi-Agent System performs specific actions to the physical network, e.g., modify specific configuration on network controllers or network devices through protocols like NETCONF [RFC6241] , RESTCONF [RFC8040], MCP [MCP]. It is the component that makes the planned control and management changes a reality in the real physical network.

5.3.8.3. Lightweight AI and Large AI Model Collaboration Interface

Collaboration between small AI model and large AI model is also designed to be supported by this interface.

In the past, we only support AI and machine learning technologies at the network level, e.g., we can use collected various different network data to provide network analysis and generate network insight. With more intelligence introduced into the network element, more GPU/NPU resource can be allocated for AI inference, this make collaboration between large AI model and small AI model possible.

Large AI models can provide basic logical reasoning and generalized analytical decision-making capabilities While specialized small AI models can provide efficient problem-solving capabilities in specialized areas. The synergy between the two allows the AI agent to combine both multitasking generalization capabilities and domain expertise, thus minimizing the reliance on human intervention in the network management process.

On one hand, we can use accumulated field engineering expertise to train large AI model into one foundation model for fault management AI agent, On the other hand, we can deploy small AI model, leverage hardware resource or chipset resource in the intelligent network element to collect more fine granularity data or provide local processing for Collected data and summary report generation, Trend prediction, etc. When small AI model is outdated and unable to detect specific applications or security risk, these specific applications and security risk information can be collected by network analytics platform to retrain this small AI model and re-deploy it in the same network element when this small AI model has been trained to work correctly to detect applications or security risk.

With collaboration between large AI model and small AI model, we can allow Network AI Agent within the Network controller interact with network element and has more quick response to network change.

This collaboration, facilitated by APIs or agent communication protocols like A2A [A2A], combines the generalization power of large models with the efficiency and low-latency of specialized small models, leading to quicker and more context-aware responses to network change.

5.3.9. Feedback-driven Improvement Interface

The architecture should incorporate mechanism for continuous improvement based on feedback. This includes collecting data on AI decisions, network performance, and user feedback to identify areas for enhancement. By analyzing the feedback, the system can adapt and optimize its operations over time, leading to better performance and more accurate decision-making. For example, if a Network AI Agent fails to accurately identify the exact cause of a network incident, the relevant records can be submitted as negative samples to the LLM which provides inference services, this allows the LLM to be trained on these negative samples for optimization.

This interface is implemented through a combination of system interfaces that collect, process, and apply feedback. Operational feedback—including the outcomes of AI decisions, network state metrics—is collected as structured data via system logging streams (e.g., in JSON format) and message queues (e.g., Kafka). This data is then consumed by analytics components and machine learning platforms through APIs (e.g., RESTful, gRPC) to refine AI models, for instance, by using failure records as negative samples for fine-tuning. Subsequently, optimized models and updated knowledge are deployed back into the runtime system via model serving and

configuration management interfaces, closing the improvement loop.

5.3.10. Network Element AI Agent and Network AI Agent Collaboration Interface

Network devices collect information from multiple dimensions, including flow information, configuration, events, alarms, logs, dynamic topology and routes, and device status (including CPU, memory, and hardware health). With large amount of data collected to the domain controller for analysis and processing, the data accuracy is very limited and therefore it is hard to determine the service impact within 1 minute. In addition, it usually require multiple step interaction, complex task management with various different data types or data sources.

To address those challenges, the network AI Agent can delegate massive data analysis and processing to distributed AI Agent in each network element, e.g., a) only allow distributed AI Agent export processed analytic data to help establish global view of network observability. b) or export key network fault information for Network AI Agent for further investigation the root cause of the problem.

For the former case, routing protocol specific fault data such as BGP Status Changed, OSPF Neighbor state changes, IS-IS Adjacency Changed data or hardware related fault data such as Optical fail, Physical Port down can be collected and using pre-trained LLM model with expert experience to match fault pattern and invoke corresponding routing protocol troubleshooting MCP tools and finally root cause. In addition, it allows network maintenance engineer using nature language interface to look up troubleshooting information or it allows Network AI Agent or Task Agent at the network element using MCP interface to invoke tools from MCP server within the network element.

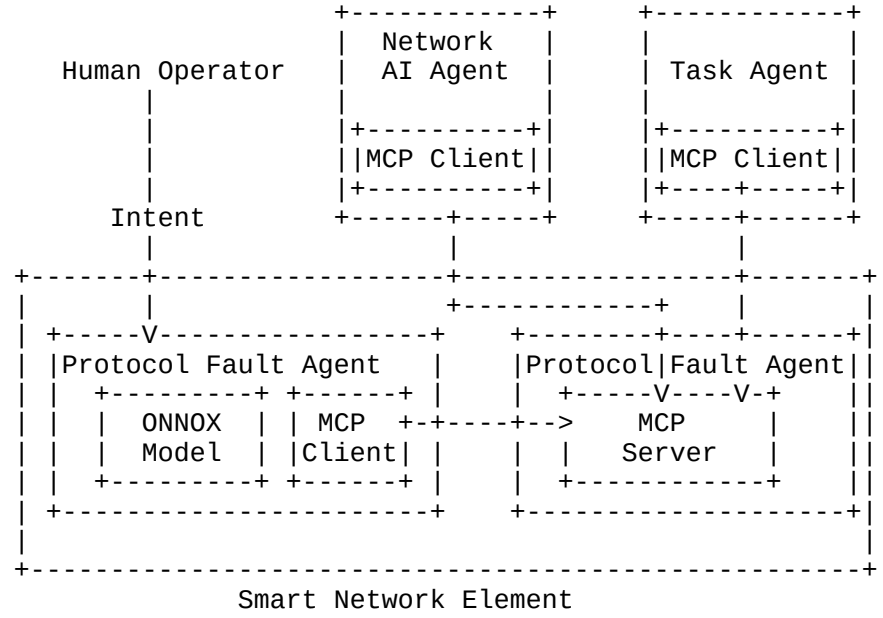


Figure 3: Network Element AI Agent and Network AI Agent Collaboration Usage Example

6. AI Driven Network Operations: Relationship Between Characteristics and Functional Components

The architecture in Figure 2 provides a concrete implementation framework to realize the six key characteristics of AI-driven network operations described in Section 4. Each characteristic is directly supported by specific functional components within the Autonomous

Domain. The following clarifies how the architecture operationalizes these characteristics:

* Intent: The Network Applications Layer conveys a high-level user intent via northbound interfaces. The Network AI Agent interprets this intent and translates it into actionable network operation tasks to each task Agent.

* Knowledge: The Knowledge Base in Agent Gateway serves as the central repository for domain-specific knowledge, expert rules, and historical data. It provides the necessary context and long/short memory to support accurate decision-making by task Agents.

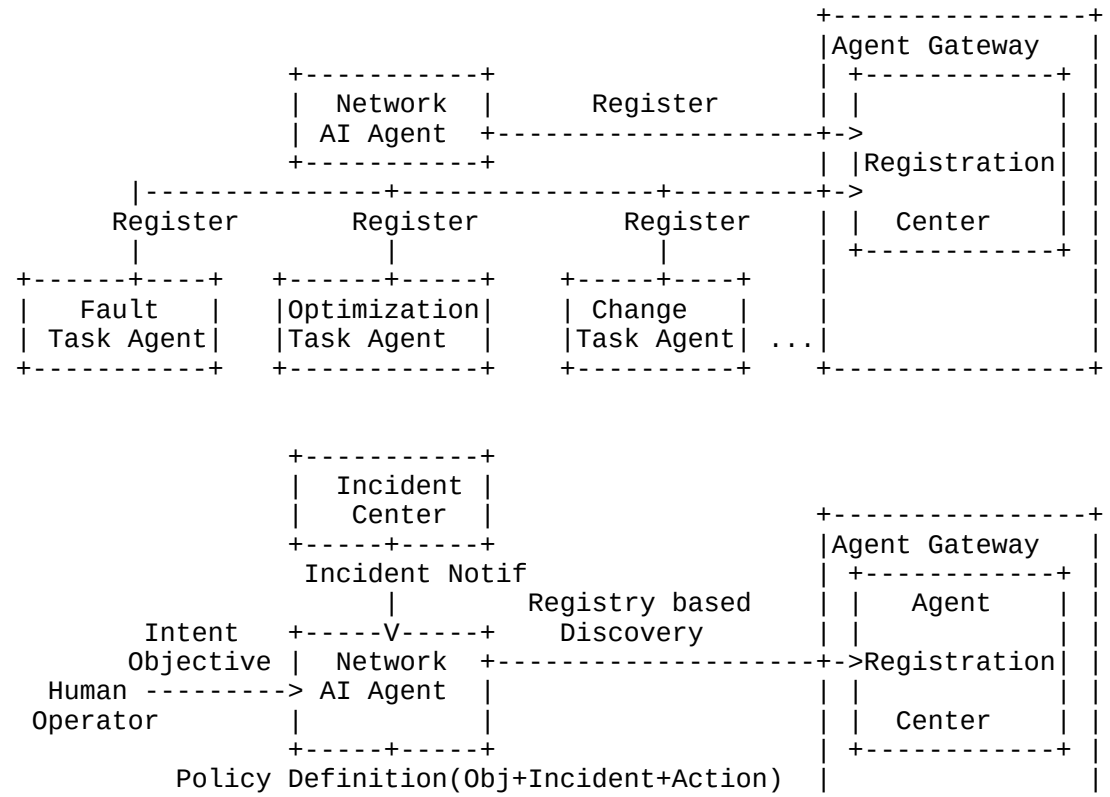
* Analysis: The AI Agent in Multi-Agent System performs intelligent analysis using data and tools. It leverages the Network Digital Twin to simulate and validate scenarios, enabling data-driven insights and gap analysis between intent and current network state.

* Decision: The AI Agent in Multi-Agent System makes informed decisions based on its analysis results. It utilizes the Network Digital Twin for risk-free validation before finalizing decisions. The decision may be sent to human operators for confirmation before actions are taken.

* Awareness: The AI Agent in Multi-Agent System gathers data from the Physical Network, it may also fetch data from the Network Digital Twin which maintains a dynamic, virtual representation of Physical Network. Together, they provide comprehensive network visibility and context-aware awareness.

* Execution: : The AI Agent in Multi-Agent System implements validated decisions by applying configurations or control actions to the Physical Network via southbound interfaces such as NETCONF, RESTCONF, or Model Context Protocol [MCP].

7. AI Agent Registration and Team formation



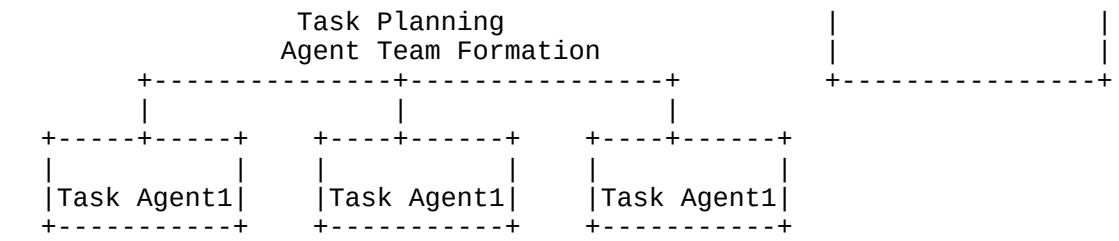


Figure 4: AI Agent Registration and Team formation Usage Example

The Agent gateway at the network level provides agent registration for both embedded AI agent in each network element and network AI agent and associated task agents. The following steps are performed to provide Event driven AI Agent Team formation within the Agentic AI network management architecture:

Step 1: Human Operator pre-provision user intent which comprises objective, incident list and corresponding action list.

Step 2: Network AI Agent receieves user intent and generate corresponding policies which comprise objective, incident list and corresponding action list. In addition, Network AI Agent subscribe corresponding incidents from incident center.

Step3: Network AI agent generate task planning based on objective and then discover matched task agent lists based on planned task from registration center within the agent gateway.

Step4: Upon receiving incident from the incident center, network AI agent assign the tasks to task agents corresponding to specific incident received.

Step 5: Network AI Agent distribute task to corresponding task agents and complete task agent team formation.

8. Agent to Agent Communication Security

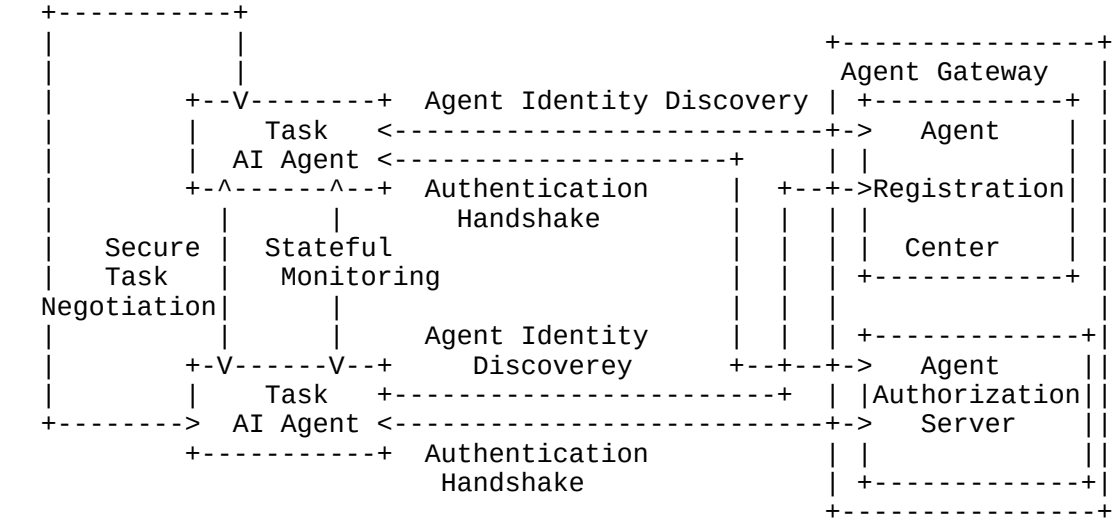


Figure 5: Agent to Agent Communication Security Usage Example

The following steps are performed to provide Agent to Agent Communication Security within the Agentic AI network management architecture:

Step 1. Discovery via Agent Card: The workflow begins when a client agent requests the Agent Card from the remote agent’s /.well-

known/agent.json endpoint. This JSON file acts as a secure manifest, declaring the agent's identity, capabilities, and required security schemes.

Step 2. Authentication Handshake: Before any task is sent, the client must fulfill the authentication requirements listed in the Agent Card. This typically involves an OAuth 2.0 flow where the client obtains a JSON Web Token (JWT) to prove its identity and permissions.

Step 3. Secure Task Initiation: :Communication is established over HTTPS/TLS. The client sends a tasks/send request using JSON-RPC 2.0. The server validates the token and authorizes the specific task based on the client's role.

Step 4. Stateful Monitoring & Feedback: :The task moves through a strictly defined lifecycle (submitted → working → completed). Security is maintained throughout as updates are streamed via Server-Sent Events (SSE) or webhooks, each tied to the unique, authorized Task ID.

Step 5. Artifact Delivery: Final results (Artifacts) are delivered only after the task reaches a completed state. These are structured objects (text, files, or data) returned to the verified requester, ensuring data integrity and preventing unauthorized access to output.

9. AI Driven Network Operations: A collection of Use Cases

Network AI Agent could help in the following phases which are usually mentioned in network management:

* Network Planning and Design: includes the understanding of user intent, generation of solutions, and simulation for decision-making.

* Service Deployment: includes the construction of the physical network, as well as intent understanding, pre-deployment simulation, automated configuration, post-deployment validation, and other capabilities to enhance the efficiency and accuracy of network configuration for service deployment.

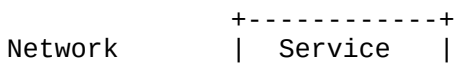
* Network Monitoring and Troubleshooting: includes intent monitoring, issues identification, solution generation, evaluation and decision-making, solution implementation, and service validation.

* Network Change and Optimization: involves the design, evaluation, decision-making, implementation, and validation of network configuration changes or optimizations to improve network operation efficiency.

In all phases and use cases, after the Agent performs specific action, it always continuously monitors the network by data collection. Based on the result of network running analysis and user explicit feedback, it may adjust and optimize the management strategy if necessary.

9.1. Multi-Agent Collaboration on Network Configuration Change

Network configuration changes are needed in scenarios such as optimizing network or service performance, provisioning new network services, or resolving network incidents/faults.



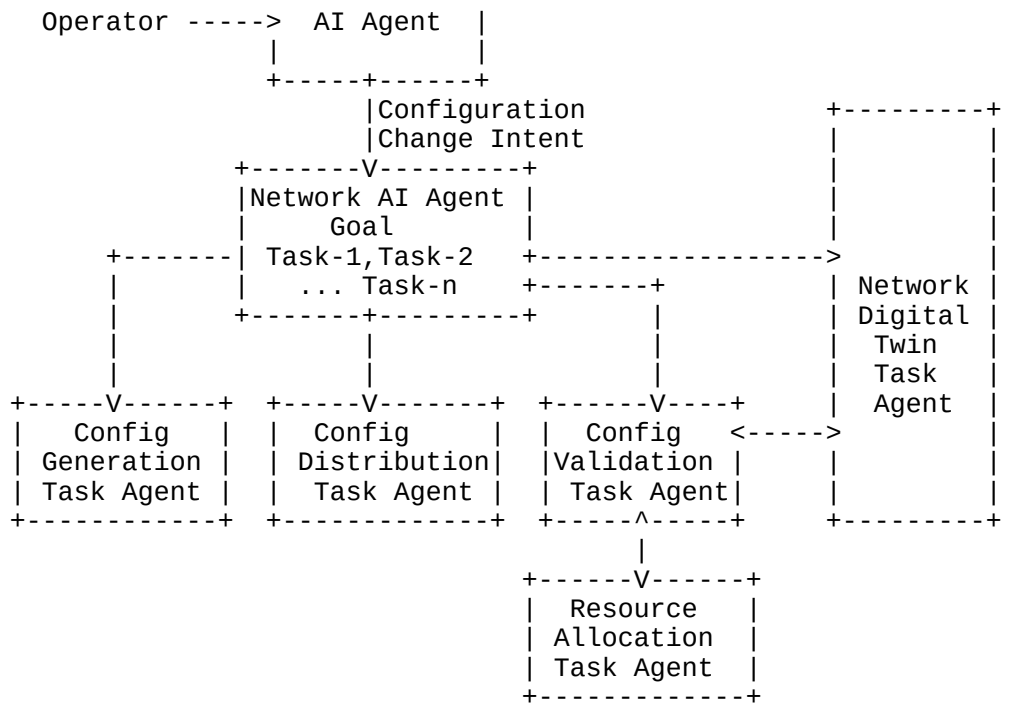


Figure 6: Intent Based Network Configuration Change Usage Example

Network configuration change leveraging Network AI Agent and Network Digital Twin may experience the following typical steps:

Step 1: The network operator inputs the intent of network configuration change into the Network AI Agent using natural language. The network operator may simply explain the objectives and requirements of the changes.

Step 2: Network AI Agent first verifies the identity of the user requesting the change and checks the user’s permissions to make certain types of network changes against predefined rules or policies. It then understands and parses the initial intent of the request, by leveraging the powerful knowledge and reasoning capabilities of LLM and decompose the tasks into configuration generation task, configuration distribution task, configuration validation task and assign to corresponding task agents. Configuration generation Task Agent first generates initial suggestions for specific network configuration update, which may include multiple possible network configuration change plans if possible.

Step 3: Network AI Agent further communicates with the Configuration Validation task agent and Network Digital Twin task agent to validate the suggested configuration change, including the syntax and semantics of the configuration, verification of effected application and resources. The network digital Twin task agent may generate a report indicating the validation result, and suggested configuration fix when the validation fails after network simulation leveraging the current physical network operational state.

Step 4: Network AI Agent may generate a configuration change plan and submit to the network operator for approval. Based on the feedback from the operator, Network AI Agent then further decides whether to optimize the change plan or deliver the plan to the Configuration Distribution task agent to conduct the physical network configuration change. The configuration distribution task agent may further communicate with resource allocation task agent to obtain network resource (e.g.,vlan, IP subnet) allocated by

resource allocation task agent.

9.2. Multi-Agent Collaboration on Network Troubleshooting

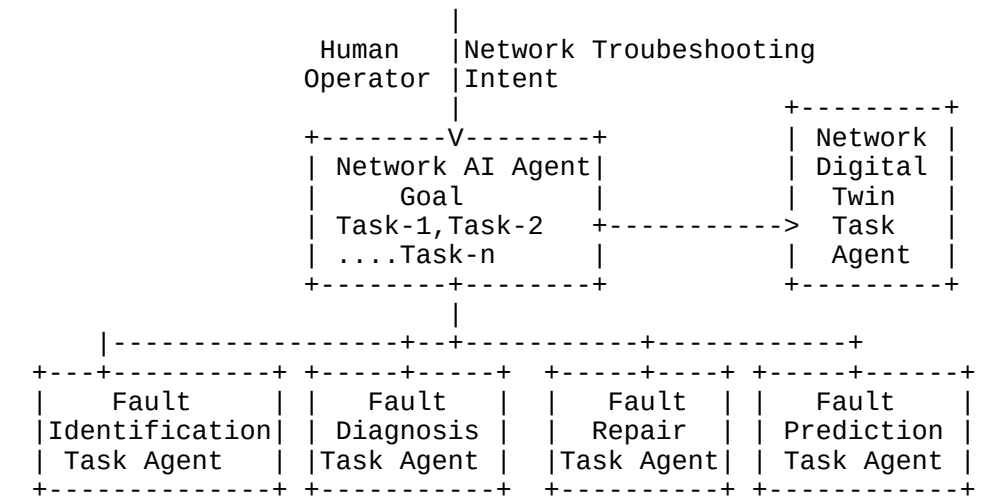


Figure 7: Intent based Network Troubleshooting Usage Example

The network operator inputs the intent of network configuration change into the Network AI Agent using natural language. Network AI Agent could plan and decompose network troubleshooting tasks and coordinate with fault identification task agent, fault diagnosis task agent, fault repair task agent and fault prediction task agent to assist in network troubleshooting in the following significant aspects:

* Fault Identification: Network AI Agent coordinates with fault identification task agent continuously monitors and aggregates data from various sources, the comprehensive data collection provides a holistic view of the network operational state. By analyzing the real-time data, fault identification task Agent could detect network anomalies swiftly, which enables the prompt identification of potential issues before they escalate into major faults, minimizing downtime or service disruptions. In some cases, the Lightweight AI located in the Network Element may handle some simple fault identification tasks (e.g., optical module fault automatic identification) to enhance the awareness, while the fault identification task agent and LLM could leverage their powerful processing capabilities to analyze the time-domain data collected from the optical module.

* Fault Diagnosis: Once a fault is identified, Network AI Agent coordinate with fault diagnosis task agent to delve into diagnosing the exact cause, fault diagnosis task agent may also invoke some existing operations such as "incident-diagnose" RPC defined in [I-D.ietf-nmop-network-incident-yang]. By correlating symptoms and/or applying AI models trained on historical data, fault diagnosis task agent can narrow down the potential causes and pinpoint the exact cause, which accelerates the diagnosis process and reduces the time needed to address the issue.

* Fault Repair: After diagnosing the fault, Network AI Agent can coordinate with fault repair task agent to generate targeted repair solutions. These solutions range from specific configuration adjustments to more complex fixes (e.g., hardware replacement). Fault Repair task Agent would also communicate with the Network Digital Twin task agent to simulate the proposed repair solutions and get feedback from the Network

Digital Twin task agent. In advanced setups, fault repair task agent may automatically execute these repairs, ensuring quick restoration of normal operations and enhancing the overall reliability and efficiency of network management. But the fault repair task agent may also first present the fault details and repair advice to the network operator for review, and proceed to carry out the repair task once it is confirmed.

* Fault Prediction As an advanced enhancement of fault management capabilities, fault prediction aims to reduce network risks through proactive management that prevents problems before they occur. Before a fault actually occurs, the fault prediction task agent can coordinate with network digital twin task agent to construct a dynamic simulation model by collecting real-time multi-dimensional operational state data, including network topology, traffic load, and device performance indicators. Based on the network data, the fault predication task agent uses large models and machine learning algorithms (such as time-series prediction models and anomaly detection models) to reason and analyze potential faults—for example, predicting the risk of physical link interruption based on optical cable signal attenuation data. Furthermore, the fault prediction task Agent generates recommended operations to avoid faults and validates them through simulation in the network digital twin task agent, thereby achieving predictive maintenance of the network.

9.3. Multi-Agent Collaboration on Network Optimization

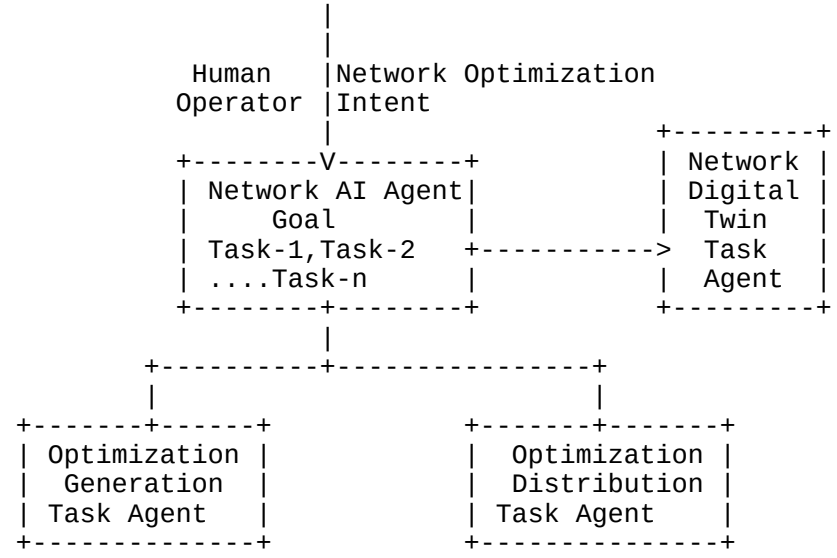


Figure 8: Intent based Network Optimization Usage Example

Network optimization is often introduced due to the Network AI Agent's awareness of some potential network faults or anomalies through continuously monitoring of network operational state, e.g., AI models may predicts network congestion by analyzing historical and real-time network traffic data. It may also be triggered by the network operator actively inputting the network optimization intent.

Based on the analysis of network data and user's intent (if any), Network AI Agent collaborate with Optimization Solution Generation Task Agent to propose network optimization strategies. For instance, once the network congestion sometime in the future is predicted, it may proactively optimize the network configuration, or suggest scaling up to meet specific demands.

Before the network optimization is conducted, Network AI Agent

coordinates with the network digital twin task agent to implement and evaluate the optimization solution using the Network Digital Twin platform. This may need repeated trials and validations based on specific evaluation criteria, before the optimal strategy could be selected. Network AI Agent may also first present the suggested network optimization solution to the network operator for review, and apply it to the physical network through optimization solution distribution task agent after obtaining approval from the network operator.

9.4. Network level Energy Efficiency Management in the IP+Optical network

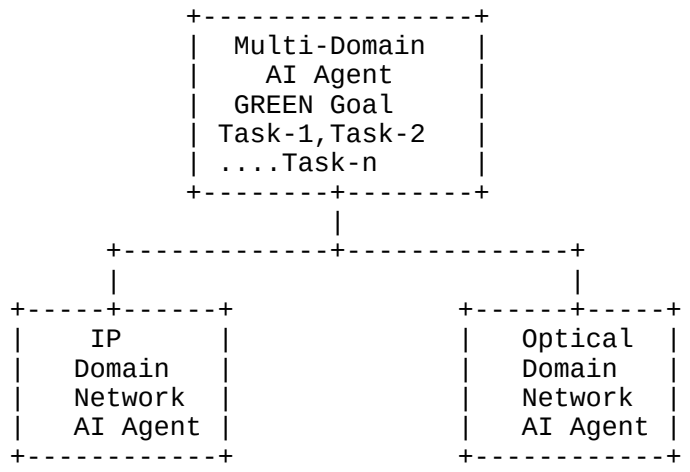


Figure 9: Intent based Network level Energy Efficiency Management Usage Example

Network level Energy Efficiency refer to a set of processes used to discover a inventory of capabilities, use specific metrics to monitor and assess energy consumption of the entire IP+Optical network , operate, and control the use of available energy in an optimized manner while achieving the network' s functional and performance requirements by improving overall network utilization.

Multi-Domain AI Agent can work together with network AI Agent in each autonomous domain to allow network operators not only see real time energy consumption in the network devices of large scale network through interaction with the GREEN Network AI Agent, but also allow them see

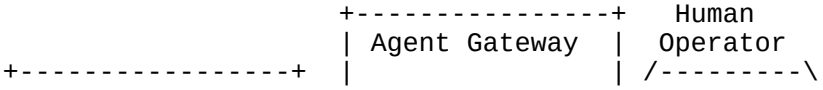
o which network devices enable energy saving, which devices not, which are legacy ones,

o The total energy consumption changing trend over the time of the day, for all network devices,

o Energy efficiency changing trend over the time of the day for the whole network.

On the other hand, With the end to end observability to energy consumption statistics data and energy efficiency statistics data, the Network AI Agent in each autonomous domain can collaborate with network digital twin to know which part of the network need to be adjusted or optimized based on network status change.

9.5. Network Security Drills (Human in the Loop)



```
            | Network AI Agent|  |+--------------+| | Analyze |
            |     Goal        |  ||              || | Define  |
            | Task-1,Task-2   <-->|Observability |<-> Inject  |
            | ....Task-n      |  ||              || \---------/
            +-------+---------+  |+--------------+|
                    |            |                |
                    |            +---------^------+
                    |                      |
            +-------V------ ---------------V---------------+
            | Dynamic Attack and Defense Verification System|
            |        Based on Network Digital Twin          |
            |      +---------+      +---------+             |
            |      | Dynamic |      | Dynamic |             |
            |      | Security|      |Security |             |
            |      | Attack  |      |Defense  |             |
            |      |  Task   |      |  Task   |             |
            |      |  Agent  |      |  Agent  |             |
            |      +---------+      +---------+             |
            +--------------------^---------------------------+
                                 |Data Collection
            +--------------------+---------------------------+
            |            Network Infrastructure              |
            +-----------------------------------------------+
```

Figure 10: Intent based Network Security Drill Usage Example

The human operator can work together with the Network AI Agent to conduct Network security Drill. The human operator can instruct the Network AI Agent with specific injection policy to work with network digital twin help construct a dynamic attack-defense verification system in network security drills through NDT and AI reasoning capabilities. The dynamic attack-defense verification system comprise dynamic security attack task agent and dynamic security defense task agent which are responsible security risk attack task and security risk defense task respectively assigned by the network AI agent. The dynamic security attack task agent uses generative AI to automatically generate diversified attack paths, models network topologies with graph neural networks, covers attack stages such as reconnaissance and penetration, and dynamically adjusts strategies via reinforcement learning to simulate the adaptive characteristics of network attacks. The virtual range built based on the NDT can 1:1 map the production environment, supporting simulations of composite scenarios like ransomware chain attacks and supply chain attacks — such as simulating the entire process of Contivirus laterally penetrating to domain controllers through weak passwords.

During drills, Human operator can instruct the Network AI Agent to work with the dynamic security defense task agent to automatically deploy virtual environments with vulnerabilities, collect defense response data in real time through NDT, and generate attack path heatmaps and repair suggestions. This capability can further verify emergency response processes, inject real-time threat intelligence to dynamically update drill scenarios, and simulate end-to-end automated deployment, vulnerability injection, and real-time analysis of security drills, enhancing the proactive verification ability of defense systems against real-world threats.

10. Challenges of Integrating Network Digital Twin and Agentic AI into Network Management

In addition to the research challenges in coupling AI and network management specified in [I-D.irtf-nmrg-ai-challenges], this document also identifies some challenges that need to be considered when integrating agentic AI together with network digital twin for network operations.

10.1. Create domain specific language for agents and models

Feeding network data into machine-readable knowledge for autonomous AI agents involves transforming raw telemetry and metadata into a structured semantic format. This process typically uses the Model Context Protocol (MCP) and Domain Specific Languages (DSLs) to bridge the gap between low-level network signals and high-level AI reasoning.

o Semantic & Contextual

Network data in the wire format is often low-fidelity and lacks the context for AI to "reason" about intent or impact. AI agents may struggle to understand rigid structure for the interaction,e.g., a numerical value represents "Frame Delay" without explicit ontological mapping. Multi-task collaboration and Agent operation involve some ambiguity and uncertainty. The natural language interactions should be supported for understanding and handling uncertain or ambiguous tasks. With the presence of existing standard of natural language interaction, the task of defining common functions for any agent to agent communications becomes that of defining the semantics of the information that should be transferred, as opposed to defining a rigid structure for the interaction.

o Data Quality & Fragmentation

Information is often fragmented across multiple vendor tools and legacy systems, preventing a unified knowledge model. Processing high-velocity data and non-standardized formats makes it difficult for AI to generalize across different device types or data sources.

o Technical & Computational

Architectural Limits: Standard hardware often lacks the capacity for real-time AI operations at the data-plane level.

Token/Memory Limits: Large AI models have finite "context windows", which restricts their ability to process long or complex network sequences.

o Operational & Trust

"Black Box" Problem: AI decisions can be opaque, making it hard for human operators to audit or trust automated actions.

Skill Gap: There is a significant shortage of professionals with the cross-disciplinary expertise needed for both networking and AI development.

10.2. Trust and Security

Multi-Agent Collaborations and interactions can be break down into 4 typical scenarios:

o Human operator-> AI Agent-> APIs/Tools/APIs/LLMs

In a single-agent scenario, Human operators access services through the network management AI agent. The network management AI agent has multiple functions (fault and optimization), and authentication is required to prevent users from Performing unauthorized actions by exploiting privilege vulnerabilities,e.g., accessing the optimization function interface when they only have fault agent permissions.

o AI Agent -> API Services

In a single-agent scenario, the network management AI agent triggers

tasks automatically based on trace and log information. In some cases, the logging or decision-making process cannot be traced.

o Human operator-> AI Agent -> multiple AI Agent

Multiple agents may call each other. For example, if a faulty agent A calls an optimization agent B, authentication is required to prevent Manipulating communication channels of agents to influence decision-making processes.

o External AI Agent ->AI Agent-> APIs/Tools/APIs/LLMs

External AI agents can directly access the network management AI Agent by simulating human operation through interface protocols. For example, a customer AI agent can access the Network AI Agent through a northbound interface protocol such as A2A, MCP. In some case, there might have target flaws in protocols like MCP or A2A; e.g.,consent bypass, context hijacking, etc.

Ensuring robust security throughout the entire AI-based network management architecture is essential to prevent unauthorized access and maintain the security of the network infrastructure. The security risk can be break down into the following cases:

o External system interacts with AI agent

Human Operators or external systems bypass their privileges to operate the Network management AI agent

* Human operators operate through a network management AI agent, but the service scope that the AI agent can handle may exceed the user’s authorized privileges, leading to unauthorized access.

* Human operators manipulate business operations using agent delegation and authorization.

* Human operators Coerce intelligent agents to manipulate users into performing covert operations.

o AI Agent Interact with Tools/APIs/LLMs

* Interact with Tools/APIs/LLMs with Privilege Escalation

  Since network management AI agents rely on LLM for inference when accessing APIs and tools, there is a possibility of malicious injection scenarios where the APIs accessed by the agent exceed the expected scope,e.g.,Using AI to generate execution environments and inject malicious code.

* Interact with Tools/APIs/LLMs without audit

  When an agent interacts with an API or tool, the logs are currently recorded as system logs, which cannot distinguish between different agents, the logging or decision-making process for specific agent cannot be traced and pose a risk of repudiation.

o Multiple Agent Collaboration and Communication

* Multi-Agent Communication with Privilege Escalation

  When agents communicate with each other via intent communication and understanding, there is a cascading permission amplification problem,e.g.,AI-generated false information disrupts the reasoning process , leading to Privilege Escalation.

* Interaction with internal AI Agent

  As a microservice, a network management AI agent can be accessed by other services or AI agents,e.g., Performing unauthorized operations by exploiting authentication vulnerabilities,which poses a risk of Privilege Escalation.

10.3. Protocols between Agent and Agent/Human operator/Tools

10.3.1. High Risk Operations

"network change" in the network management field refers to the modification, adjustment, or configuration of physical or logical resources in the existing network. Because these operations directly affect the continuity and stability of existing network services, even minor unauthorized access or errors can lead to large-scale network outages. In case of high risk operation, the following measures should be taken into account.

o Regardless of whether the initiator of the operation is a human or an agent, each network change request must be re-authenticated.

o For operations involving "bulk deletion" or "core route changes", the system must require approval from a second high-privilege account before issuing the command or enforcing the policy.

o High-risk configurations should be tested on a very small scale first. The system should automatically monitor the indicators, and if any abnormalities are detected, the configuration should be automatically rolled back within milliseconds.

10.3.2. The Timeliness Requirements of Collaboration

For real-time network operation and maintenance scenarios with high real-time requirements, such as scheduling strategy optimization and critical fault repair, the rapid generation of network optimization decisions is crucial. However AI Agents based on large models adopt a "Token-based" generation and reasoning approach, which is limited by computing power and algorithms, resulting in generally slow reasoning speeds. In addition, the simulation and verification process of Network Digital Twin (NDT) further increases decision latency, which leads to long end-to-end decision-making time in complex scenarios and is difficult to meet the real-time requirements of services.

Also tasks such as fault diagnosis, complaint handling, and user experience improvement often have strict time constraints. For example, if a fault is not resolved within a set time, it can trigger an escalation of the complaint, requiring efficient collaboration among multiple agents. In addition, In a network management environment, you might need agents to subscribe to real-time network alarms or telemetry events. However Google-initiated A2A protocol primarily follows a task-oriented "request-response" model and doesn’t support a native pub/sub or event-driven architecture.

To improve decision efficiency, continuous efforts are needed in lightweight NDT modeling algorithms, optimizing large model reasoning frameworks (such as quantization technology and parallel computing), and deploying high-performance AI acceleration hardware.

10.3.3. Collaboration Reliability

Fault diagnosis and complaint handling in the network management field are complex tasks, typically involving 10 to 20+ fields for one single message exchange between two AI Agents and requiring a high level of expertise. In addition, Reliable task collaboration is

incomplete and not sufficient for network management field, e.g., Google-initiated A2A protocol doesn’t defined handling strategies for task rejection, missing information during task collaboration, and failure to achieve task objectives.

also In network management area, data sources can be diverse and heterogeneous, leading to potential issues such as data inconsistencies, missing, or outdated data. Poor-quality data may result in inaccurate AI predictions and decisions. For example, if incorrect or outdated network configuration data is provided, the model may provide incorrect repair advice when diagnosing network incidents or faults, it may suggest checking an non-existing interface. Ensuring that data is properly cleaned, validated, and maintained is a significant challenge in providing reliable inputs for AI-driven network management.

10.4. Benchmarking

10.4.1. Single Agent Benchmarking

The core of Single Agents assessment lies in its "omnipotence" and the quality of its direct interaction with the environment.

o Task complexity bottleneck

When dealing with long-term, multi-step tasks, single agents are prone to hallucination accumulation and reasoning chain breaks. Benchmark tests need to evaluate their persistent logic capabilities under context window constraints.

o Command compliance and control

The challenge lies in whether the model can strictly follow complex system prompts, especially when there are many tool options, which can easily lead to tool invocation errors or the omission of some instructions.

o Generalization and memory

To evaluate how monolithic models can efficiently manage long-term memory to handle new tasks without relying on external collaborative support.

10.4.2. Multi-Agent Benchmarking

The focus of collaborative assessment has shifted from "individual capabilities" to "system dynamics":

o Coordination and communication overhead:

Multi-agent systems (MAS) involve complex task decomposition and allocation (global planning). Benchmarking needs to measure the communication efficiency between agents, alignment consistency, and the high latency and cost caused by multi-turn interactions.

o Error tracing and attribution:

When the final task fails, it is extremely challenging to accurately pinpoint which "role" went wrong or in which round of collaboration there was a deviation (failure attribution) due to the presence of multiple participants.

o Swarm intelligence and social emergence:

It is necessary to assess the dynamics of competition and cooperation among intelligent agents, including the existence of systemic risks

such as information silos, resource waste, or vicious cycles.

11. Security Considerations

The security consideration from [I-D.irtf-nmrg-network-digital-twin-arch] apply here. In addition, the following architectural risks need to be considered:

o Memory Poisoning：If the AI/ML models used by the network AI Agent or Network digital twin are compromised or poisoned with malicious/ fake data,they could begin making incorrect or malicious decisions. Robust checks and validation are necessary to ensure the integrity of these models. Session isolation or memory access authentication is also required to mitigate such risk.

o Misuse of Tools： When network AI Agent interacts with tools, Deceptive prompts or commands can be introduced. Tool access verification, tool monitoring, log tracking of AI tool usage are required to mitigate such risk.

o Privilege Compromise：When human operators interact with Network AI Agent or Network AI Agent interact with tools/APIs/LLM, unauthorized actions might be performed by exploiting privilege vulnerabilities. Fine-grained permission control, dynamic access verification, and role change monitoring are required to mitigate such risk.

o Resource Overload：When Network AI Agent interact with tools/APIs/ LLMs, system failures might be caused by exploiting resource-intensive features. Deployment of resource management controls to limit high-frequency task requests from agents is required to mitigate such risk.

o Cascading Hallucinations：In case of multi-agent collaboration or communication, AI-generated false information might disrupt the reasoning process.Output verification, secondary verification of AI-generated knowledge are required to mitigate such risk.

o Intent Breaking &Goal Manipulation：When external AI Agent or human operators interact with the network AI Agent, reasoning through agent planning capabilities might be manipulated. Planning verification, managing reflection processes, goal consistency protection are required to mitigate such risk.

o Misaligned & Deceptive Behaviors：When Network AI Agent interact with tools/APIs/LLMs, harmful operations might be performed by exploiting reasoning vulnerabilities. Manual confirmation of high-risk operations, logging, monitoring, and deception detection are required to mitigate such risk.

o Repudiation & Untraceability：In case of multi-agent collaboration or communication, the logging or decision-making process might not be traced. Logging & cryptographic signature, cryptographic verification are required to mitigate such risk.

o Identity Spoofing & Impersonation：When Human operators interact with the network AI Agent or in case of multi-agent collaboration or communication,unauthorized operations might be performed by exploiting authentication vulnerabilities. Comprehensive identity verification, trust boundary control, and continuous monitoring are required to mitigate such risk.

o Overwhelming HITL(Human In The Loop)：In case of multi-agent collaboration or communication,fatigue auditors attack might take place. Developing advanced human-machine interaction frameworks and adaptive trust mechanisms are required to mitigate such risk.

o Unexpected RCE & Code Attacks：When the network AI Agent interacts with tools/APIs/LLMs, Using AI to generate execution environments and inject malicious code might take place. Restrict AI code generation permissions, sandbox isolation, and manual review of generated code are required to mitigate such risk.

o Agent Communication Poisoning：In case of multi-agent collaboration or communication, communication channels of agents might be manipulated to influence decision-making processes. Message authentication, communication verification, implementing multi-agent authentication mechanisms.

o Rogue Agents in MAS：In case of multi-agent collaboration, there might be malicious or compromised agents. Restricting the autonomy of agents and conducting regular AI testing are required to mitigate such risk.

o Humans Attacks on MAS：In case of multi-agent collaboration, business operations might be manipulated using agent delegation and authorization. Restricting the delegation mechanism, implementing AI agent identity authentication, and isolate tasks in segments are required to mitigate such risk.

o Human Manipulation：When Human operators interact with the network AI Agent, AI agents might be coerced to manipulate users into performing covert operations. Safety guardrails, content moderation, output content detection are required to mitigate such risk.

o Insecure Inter-Agent Protocol Abuse：In case of multi-agent collaboration, there might be target flaws in protocols like MCP or A2A; e.g.,consent bypass, context hijacking, etc. Strong authentication,data validation, restricting delegation to scoped function,logging agent and tool invocations and encrypting communications are required to mitigate such risk.

o Supply Chain Compromise：In case of multi-agent collaboration, Vulnerable, malicious, outdated, harmful components might be included into the agent. Digital signatures of SBOMs (AI_, Agent_),applying version control, chaining authentication, environment isolation are required to mitigate such risk.

o Lifecycle security: The entire management lifecycle of the network AI agents and the network digital twin—from initial deployment and configuration to updates and decommissioning—must be secured against unauthorized access and manipulation.

12. IANA Considerations

This document has no requests to IANA.

13. Conclusion

The following items were felt to be good starting points for IETF work:

* Nature Language Interaction protocol to ensure both The accuracy and efficiency of structured data for deterministic tasks and natural language interactions for understanding and handling uncertain or ambiguous tasks.

* Translate Service Level YANG Data model and Network Level YANG Data model into DSL payloads and APIs which can be consumed by Agents and Models.

* Human and Agent Interaction to support explainable, observability and controllable capabilities.

* Define semantic information transfer or agent prompt language template for Agent to Agent Communication.

* Agent to Agent Protocol extensions for IP Network Agent and Network Element Agent Collaboration.

14. References

14.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

14.2. Informative References

[A2A] "Agent2Agent (A2A) protocol", April 2025, <https://google-a2a.github.io/A2A/#/documentation?id=agent2agent-protocol-a2a>.

[A2A-T] "Agent to Agent Protocol for Telecoms (A2A-T)", 2025, <https://www.tmforum.org/resources/introductory-guide/ig1453-agent-to-agent-protocol-for-telecoms-a2a-t-v1-0-0/>.

[Agent-skills] "Agent Skills", 2025, <https://agentskills.io/home>.

[Google-Agents-Whitepaper] "Agents", 2024, <https://www.kaggle.com/whitepaper-agents>.

[I-D.hong-nmrg-agenticai-ps] Hong, Y., Youn, J., Wu, Q., and B. Claise, "Motivations and Problem Statement of Agentic AI for network management", Work in Progress, Internet-Draft, draft-hong-nmrg-agenticai-ps-00, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-hong-nmrg-agenticai-ps-00>.

[I-D.ietf-nmop-network-incident-yang] Hu, T., Contreras, L. M., Wu, Q., Davis, N., and C. Feng, "A YANG Data Model for Network Incident Management", Work in Progress, Internet-Draft, draft-ietf-nmop-network-incident-yang-08, 13 February 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-network-incident-yang-08>.

[I-D.ietf-nmop-terminology] Davis, N., Farrel, A., Graf, T., Wu, Q., and C. Yu, "Some Key Terms for Network Fault and Problem Management", Work in Progress, Internet-Draft, draft-ietf-nmop-terminology-23, 18 August 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-terminology-23>.

[I-D.irtf-nmrg-ai-challenges] Franois, J., Clemm, A., Papadimitriou, D., Fernandes, S., and S. Schneider, "Research Challenges in Coupling Artificial Intelligence and Network Management", Work in

Progress, Internet-Draft, draft-irtf-nmrg-ai-challenges-05, 18 March 2025, <https://datatracker.ietf.org/doc/html/draft-irtf-nmrg-ai-challenges-05>.

[I-D.irtf-nmrg-network-digital-twin-arch]
Zhou, C., Yang, H., Duan, X., Lopez, D., Pastor, A., Wu, Q., Boucadair, M., and C. Jacquenet, "Network Digital Twin: Concepts and Reference Architecture", Work in Progress, Internet-Draft, draft-irtf-nmrg-network-digital-twin-arch-11, 6 July 2025, <https://datatracker.ietf.org/doc/html/draft-irtf-nmrg-network-digital-twin-arch-11>.

[MCP] "Model Context Protocol", November 2024, <https://modelcontextprotocol.io/>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/rfc/rfc6241>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <https://www.rfc-editor.org/rfc/rfc7011>.

[RFC7854] Scudder, J., Ed., Fernando, R., and S. Stuart, "BGP Monitoring Protocol (BMP)", RFC 7854, DOI 10.17487/RFC7854, June 2016, <https://www.rfc-editor.org/rfc/rfc7854>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/rfc/rfc8040>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <https://www.rfc-editor.org/rfc/rfc8639>.

[RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <https://www.rfc-editor.org/rfc/rfc8641>.

[RFC9232] Song, H., Qin, F., Martinez-Julia, P., Ciavaglia, L., and A. Wang, "Network Telemetry Framework", RFC 9232, DOI 10.17487/RFC9232, May 2022, <https://www.rfc-editor.org/rfc/rfc9232>.

[RFC9315] Clemm, A., Ciavaglia, L., Granville, L. Z., and J. Tantsura, "Intent-Based Networking - Concepts and Definitions", RFC 9315, DOI 10.17487/RFC9315, October 2022, <https://www.rfc-editor.org/rfc/rfc9315>.

[TMF-1251D]
"AN Agent Architecture v1.0.0", May 2025, <https://www.tmforum.org/resources/introductory-guide/ig1251d-an-agent-architecture-v1-0-0/>.

[TMF-1258] "Autonomous Networks Glossary v1.2.0", May 2025, <https://projects.tmforum.org/wiki/display/PUB/IG1258+Autonomous+Networks+Glossary+v1.2.0>.

Appendix A. Acknowledgements

This work has benefited from the discussions of NMRG interim meeting on Agentic AI. Thanks Chris Janz for wonderful comments and discussion on proactive close loop.

Appendix B. Changes between Revisions

v02 - v03

* Agentic AI Architecture Update.
* Rewrite Functional Interfaces section for user to agent, agent to agent, agent to tools communication.
* Explore Relationship Between Management Characteristics and Functional Components in section 6.
* Rewrite a collection of use cases to support multi-agent collaborations.
* Rewrite Challenges section to cover Trust and Security, protocol and benchmarking.
* Add workflows for Agent Registration, Discovery, team forming.
* Add workflow for Agent to Agent Communciation Security.
* Rewrite Security Consideration Section.
* Add Conclusion Section.

v00 - v01

* Add Security Consideration Section;
* Add Acknowledge Section;
* Clarify the relation between knowlege and tools;
* Clarify the souce of knowlege;
* Clarify the key characteristics of Network AI Agent to adpat to the environment change.

Contributors

Qiufang Ma
Huawei
Email: maqiufang1@huawei.com

Zhenqiang Li
CMCC
Email: lizhenqiang@chinamobile.com

Lionel Tailhardat
Orange Research
Email: lionel.tailhardat@orange.com

Authors' Addresses

Qin Wu
Huawei
China
Email: bill.wu@huawei.com

Cheng Zhou
China Mobile
China
Email: zhouchengyjy@chinamobile.com

Luis M. Contreras
Telefonica
Email: luismiguel.contrerasmurillo@telefonica.com

Sai Han
China Unicom

China
Email: hans29@chinaunicom.cn

Yong-Geun Hong
Daejeon University
Email: yonggeun.hong@gmail.com