DMSC Working Group
Internet-Draft
Intended status: Standards Track
Expires: 1 September 2026

X. Li
China Telecom
J. Liu
Beijing University of Posts and Telecommunications
C. Du
Zhongguancun Laboratory
L. Zhang
AsiaInfo Technologies (China) Inc

28 February 2026

Multi-agent Collaboration Protocol Suite
draft-li-dmsc-macp-02

Abstract

This document specifies a Multi-agent Collaboration Protocol Suite, which enables scalable, secure, and semantically driven collaboration among distributed agents across heterogeneous networks. The protocol suite introduces Agent Gateways as key entities responsible for agent registration, authentication, capability management and other functions, while preserving direct peer-to-peer semantic interactions among agents.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

As multi-agent systems become increasingly distributed across heterogeneous networks and administrative domains, efficient, secure, and semantically meaningful collaboration among agents becomes a critical challenge. Traditional service-oriented or message-based interaction models are insufficient to capture agent-level capabilities, dynamic task decomposition, and semantic intent-driven communication.

This document specifies a Multi-agent Collaboration Protocol Suite. The suite defines a set of coordinated protocols that enable agent registration, authentication, capability synchronization, task-based invocation, and peer-to-peer semantic interaction. The Agent Gateway is a functional entity that serves as the infrastructure that provides interconnection functions for agent collaboration, mediates control, policy enforcement, and orchestration, and enables authorized agents to directly exchange semantic information.

The protocol suite is aligned with the architectural principles of control/forwarding plane separation, least-privilege authorization, and session-scoped semantic communication.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] .

3. Terminology

The following terms are defined in this draft:

* Agent: An Agent is a software entity with autonomous decision-making and execution capabilities, capable of perceiving the environment, acquiring contextual information, reasoning, and performing tasks independently or collaboratively with other agents.

* Agent Gateway: The Agent Gateway is a functional entity that serves as the infrastructure for enabling interconnection and collaboration among agents. While its core role remains consistent, it is inherently flexible in deployment and can be realized in various forms—ranging from a network service to a dedicated gateway—depending on the architectural and operational requirements of different network environments.

* Agent Management Center (AMC): It is the trusted infrastructure service responsible for agent identity lifecycle management and credential issuance.

* Agent Identity Code (AIC): An Agent Identity Code (AIC) is a verifiable, globally unique identifier that represents the identity of an Agent.

* Agent Capability Specification (ACS): An Agent Capability Specification (ACS) is a structured description of an agent’s capabilities and service information that can be stored, retrieved, and matched.

* Agent Credential: An Agent Credential is a tamper-resistant data object issued by an Agent Management Center(or its credential authority component), used by an Agent to prove identity attributes and/or authorization to a relying party. Examples include X.509 certificates and security tokens.

* Agent Autonomous Domain: An Agent Autonomous Domain is an administrative and governance domain organized and managed by a specific IoA service provider. An Agent Autonomous Domain typically includes one Agent Management Center, one or more Agent Gateways, and the agents registered within its scope.

* Agent Registration Protocol (ARP): The ARP governs how an agent formally registers with a locally attached agent gateway.

* Agent Authentication and Authorization Protocol (AAAP): The AAAP defines how authentication and authorization decisions are requested and enforced.

* Capability Digest and Synchronization Protocol (CDSP): The CDSP synchronizes abstracted capability digest information across agent gateways based on gateway-generated ACS records.

* Task-based Invocation Protocol (TIP): The TIP defines how tasks, once decomposed by the application logic, are coordinated across agents.

4. Multi-Agent Collaboration Function Entity Architecture

The DMSC architecture defines four functional entities and one external actor (User). Each functional entity represents a logical role in the IoA architecture; implementations MAY combine multiple entities into a single product or distribute a single entity across multiple services.

```
                    +-----------------------+
                    |Agent Management Center|
                    |     (Identity +        |
                    |      Credential)       |
                    +---------+-------------+
                              |  AAAP
                              |
                        +-------------------+
                        |   Agent Gateway   |
                        |   (Capability     |
                        | + Discovery       |
                        | + Registration    |     +---------------------+
                        | + Msg Distribution|     | External Resource   |
                        | + Trust Policy)   |     | Service             |
                        +--+-----+----------+     +----------+----------+
                           |     |                       |
                     ARP   |     | TIP                   |  MCP
                           |     |                       |
             +-----------------------+            +----------------------+
             |        Agent          |     A2A    |        Agent         |
             |      (Role A)         |----------- |      (Role B)        |
             +----------+------------+            +----------------------+
                        |
             +----------+------------+
             |          User         |
             +-----------------------+
```

Figure 1 Function Entity Architecture Overview

4.1. Agent

An Agent is a software entity with autonomous decision-making and execution capabilities, capable of perceiving the environment, acquiring contextual information, reasoning, and performing tasks independently or collaboratively with other agents. Each Agent is created by a specific Agent Provider and MUST complete registration with an Agent Management Center before providing services in an IoA deployment.

An Agent is responsible for:

* Maintaining its own identity information (e.g., AIC) and credentials locally.

* Maintaining its capability description (e.g., ACS) and ensuring consistency with its current state.

* Performing authentication and authorization checks for interconnection, including mutual verification of peer agents and validation of presented credentials. - Conducting agent-to-agent interaction, including session establishment, message exchange, and task/context management.

* Accessing external resources when required to fulfill tasks.

* Producing monitoring and logging data for troubleshooting, auditing, and governance purposes.

These are internal agent capabilities described here for informational purposes. They are NOT standardized as separate architectural functional components. In an interaction, an Agent MAY assume different roles depending on the collaboration mode. The DMSC architecture does not constrain the set of possible roles; specific collaboration protocols MAY define role semantics appropriate to their interaction patterns.

For example, in a task-driven collaboration:

* Leader: The Agent that initiates tasks and organizes collaboration.

* Partner: The Agent that accepts tasks and provides services, executing assigned tasks and returning results to the Leader.

A single Agent implementation MAY act in different roles across different interactions. Role assignment is per-interaction, not per-deployment.

4.2. Agent Management Center

The Agent Management Center is the trusted infrastructure service responsible for agent identity lifecycle management and credential issuance. The Agent Authentication serves as the trust anchor for an IoA deployment.

The Agent Management Center provides the following functions:

* Credential Management: Issuing, renewing, suspending, revoking, and publishing status of Agent Credentials (e.g., X.509 certificates, security tokens) bound to AICs.

* Credential Validation Support: Providing credential validation material (e.g., issuer public keys, certificate revocation lists, OCSP endpoints) to relying parties.

Multiple Agent Management Center MAY exist in an IoA deployment, each managing a subset of agents within its administrative scope. A deployment MAY realize the identity management function and the credential authority function as separate services, provided they maintain consistent identity-to-credential binding.

4.3. Agent Gateway

The Agent Gateway is a functional entity that serves as the infrastructure for enabling interconnection and collaboration among agents. While its core role remains consistent, it is inherently flexible in deployment and can be realized in various forms— ranging from a network service to a dedicated gateway— depending on the architectural and operational requirements of different network environments.

The Agent Gateway provides the following functions: identity management, agent registration, capability directory, capability discovery, semantic routing, message distribution and so on. The Agent Gateway SHOULD support synchronization with the Agent Management Center to ensure accuracy and timeliness of identity and capability information. Multiple Agent Gateways MAY exist in an IoA deployment, each serving a defined administrative scope. A deployment MAY co-locate the Agent Gateway with the Agent Management Center. More specific requirements are specified in [draft-liu-dmsc-gw-requirements][GW-REQ].

4.4. External Resource Service

The External Resource Service is a device, software component, or network-accessible service that provides specific functions to agents. Examples include APIs, databases, computation services, and other external resources.

The External Resource Service provides the following functions:

* Resource Exposure: Registering or exposing available resources and their invocation interfaces to authorized agents.

* Invocation Handling: Receiving and processing resource invocation requests from agents, executing the requested operations, and returning results.

* Access Control: Enforcing access control policies on resource invocations, including authentication of requesting agents and authorization checks.

An External Resource Service MAY represent a single resource, a resource gateway, or a resource execution environment. Deployments MAY use existing resource access protocols (e.g., MCP [Model Context Protocol], A2T [Agent-to-Tool Protocol]) for agent-to-resource interaction.

4.5. Data Objects

The following are protocol data objects referenced by the functional entities. They are not functional entities themselves.

Agent Identity Code (AIC): An Agent Identity Code (AIC) is a verifiable, globally unique identifier that represents the identity of an Agent. An AIC is allocated by an Agent Gateway during agent registration.

Agent Capability Specification (ACS): An Agent Capability Specification (ACS) is a structured description of an agent’s capabilities and service information that can be stored, retrieved, and matched. An ACS MAY use the JSON [RFC8259] [RFC8259] format, typically including: the agent’s AIC, functional capabilities, technical characteristics, service interfaces, and security requirements.

Agent Credential: An Agent Credential is a tamper-resistant data object issued by an Agent Authentication (or its credential authority component), used by an Agent to prove identity attributes and/or authorization to a relying party. Examples include X.509 certificates and security tokens.

Agent Autonomous Domain: An Agent Autonomous Domain is an administrative and governance domain organized and managed by a specific IoA service provider. An Agent Autonomous Domain typically includes one Agent Authentication, one or more Agent Gateways, and the agents registered within its scope.

4.6. Entity Summary

The following table provides a summary of all functional entities and external actors in the simplified DMSC architecture.

#	Entity	Type	Role in Architecture
1	Agent	Core entity	Autonomous task execution and collaboration
2	Agent Management Center	Infrastructure	Identity lifecycle management and credential issuance
3	Agent Gateway	Infrastructure	Capability directory, discovery, routing, message distribution, trust policy
4	External Resource Service	Infrastructure	External resource exposure and invocation handling
5	User	External actor	Task initiation, authorization, and result consumption

Figure 2 A summary of all functional entities

5. Multi-Agent Collaboration Protocol Suite Overview

The Multi-Agent Collaboration Protocol Suite defines a set of interaction processes as shown in the figure below that collectively enable secure agent onboarding, distributed capability visibility, semantic request resolution, peer-to-peer semantic interaction, and task-oriented multi-agent orchestration. Rather than operating independently, these protocols are designed to be executed in a tightly coupled manner along the agent lifecycle and collaboration workflows. The Agent Gateway (AG) serves as the anchoring point for control-plane coordination.

```
 User        Agent A      AG 1          AG 3              AMC           AG 2        Agent
 B    Agent C
  |             |            |             |                |              |<-------------
-Register|
  |             |Register--->|             |                |              |<--Register|
         |
  |             |            |--Auth Req------------------>|<---Auth Req--|           |
         |
  |             |            |<-----------------Auth Grant |--Auth Grant->|           |
         |
  |             |<-Local Bind|             |                |              |           |
         |
  |             |            |             |                |              |--Local Bind (
B,C)---->|
  |             |            |             |                |              |           |
         |
  |             |            |             |                |              |           |
         |
  |             |            |<----------->|<-----Capa Digest and Sync -->|           |
         |
  |             |            |             |                |              |           |
         |
  |             |            |             |                |              |           |
         |
  |             |            |             |                |              |           |
         |
  |             |            |             |                |              |           |
         |
  | --Req-----> |-SemR Req-->|             |                |              |           |
         |
  |             |            |             |                |              |           |
         |
  |             |            |---SemR Req->|------------------------------>|           |
         |
  |             |            |<------------|<------SemR Resp--------------|           |
         |
  |             |<-SemR Resp |             |                |              |           |
         |
  |             |============================ Semantic Session========================|
         |
  |             |>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>|
         |
  |             |<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<|
         |
  |             |            |             |                |              |           |
         |
  |             |            |             |                |              |           |
         |
  |             |            |             |                |              |           |
         |
  |--Task Req-->|              +-----------------------------+           |           |
         |
  |             | ---------->   | Semantic Routing process    |---------->|           |
         |
  |             |              + ----------------------------+           |           |
         |
  |             |--------------------------------------Invoke----------------------------
-------->|
  |             |<--------------------------------------Executing------------------------
---------|
  |             |            |             |                |              |           |
         |
```

Figure 3 The overall sequence diagram of MACP

5.1. Agent Registration and Authorization Process

An agent MUST register with its locally attached Agent Gateway before participating in any collaboration. This process is governed jointly by ARP and AAAP and establishes the agent’ s identity, trust status, and capability binding. Upon receiving a registration request, the Agent Gateway performs preliminary validation of the agent’ s identity attributes and initial capability description. The gateway then initiates an authentication and authorization request to the Central Authentication Service, conveying the agent identity, gateway identity, and requested operational scope.

The Central Authentication Service evaluates the request and returns an authorization grant or denial. Upon successful authorization, the Agent Management Center issues an Agent Credential. And the Agent Gateway finalizes the registration by assigning the agent a globally unique Agent Identity Code (AIC) and Capability Identifier(s). The AIC represents the agent’ s persistent identifier within the architecture. The AIC is allocated by the Agent Gateway only after successful authentication and authorization.

5.2. Capability Digest and Synchronization Process

Each Agent Gateway maintains detailed capability information only for its locally registered agents, and generates a structured Agent Capability Specification (ACS). The ACS is a gateway-generated representation that normalizes and organizes authorized capabilities associated with the assigned AIC. It reflects only those capabilities permitted within the approved operational scope. Gateways do not synchronize full agent capability states with each other. Instead, to support inter-gateway semantic resolution, gateways exchange capability digests using the Capability Digest Synchronization Protocol (CDSP). A capability digest is a locally generated, abstract summary of available capabilities, designed solely to indicate what kinds of capabilities exist behind a gateway, rather than how those capabilities are internally implemented or executed by agents.The structure and semantics of capability digests are intentionally decoupled from agent-internal capability representations, allowing gateways to evolve local capability models without impacting inter-gateway interoperability.

CDSP distributes these capability digests incrementally. An initial exchange establishes basic inter-gateway visibility, while subsequent updates convey only digest changes, such as newly advertised capabilities, capability updates, or withdrawals. Digest updates are versioned and acknowledged to support consistency and conflict resolution. Through this digest-based mechanism, gateways maintain a scalable and privacy-preserving view of distributed agent capabilities without requiring centralized directories or full capability replication.

5.3. Semantic Resolution and Routing Process

When a user issues a request to an agent (e.g., Agent A), the agent abstracts the request into a semantic request and submits it to its locally attached Agent Gateway (AG1). Upon receiving the semantic request, AG1 performs semantic parsing and normalization and consults its local capability directory. If no matching capability identifier is found, AG1 forwards the semantic request to a peer or upstream gateway (e.g., AG3), which repeats the same resolution procedure. If the request remains unresolved, it is further forwarded to another gateway (e.g., AG2).

When a gateway identifies a matching capability in its local directory, it generates a semantic resolution response containing the resolved capability identifier and the corresponding target agent information. This response is propagated hop-by-hop back to the originating gateway and ultimately delivered to Agent A.

Following successful resolution, Agent A and the target agent (e.g., Agent B) directly establish a semantic session. During the lifetime of this session, semantic data is exchanged directly between agents in a peer-to-peer manner, while gateways remain responsible for resolution, authorization scope enforcement, and security policy application during session establishment.

5.4. Task-based Multi-Agent Invocation Process

Task-based collaboration extends semantic resolution to scenarios requiring multiple agents and coordinated execution. When a user initiates a task request, the request is delivered to Agent A, which performs semantic understanding of the task and decomposes it into one or more sub-tasks along with the required capabilities. If Agent A does not possess task decomposition capabilities, its attached Agent Gateway MAY act as a proxy to analyze and decompose the task on behalf of the agent.

For each sub-task, Agent A submits a semantic request to its local gateway, triggering the semantic resolution and routing process. Unlike pure point-to-point semantic communication, gateways additionally apply task-level constraints, policy considerations, and capability selection logic to identify suitable target agents.

The resolved results are returned to Agent A, which then directly invokes the selected agents and establishes the necessary semantic sessions for execution. Through this mechanism, multiple agents can be dynamically selected and coordinated to collaboratively execute complex tasks.

5.5. Agent-to-Agent Process

The Agent-to-Agent Protocol (A2A) defines how two authorized agents establish, maintain, and terminate a semantic session for peer-to-peer semantic communication. A2A operates entirely between agents in the data plane. Agent gateways participate only during session authorization and parameter provisioning phases and are not required in the semantic data path once the session is established.

A2A process consists of four phases:

* Session Initiation: A semantic session is initiated when an agent, having obtained a valid resolution result, sends a session initiation message to the target agent. The message includes the Agent Identifiers, a proposed Session Identifier, the resolved Capability Identifier, and a verifiable authorization artifact derived from prior control-plane procedures. Upon transmission, the initiating agent awaits confirmation from the target agent.

* Mutual Verification and Session Establishment: Upon receiving the initiation request, the target agent verifies the initiator’ s identity, validates the authorization context, and confirms that the referenced capability is locally bound and permitted by policy. If validation succeeds, the target agent returns a session acceptance message, and both agents transition to an established state. If validation fails, the request is rejected and no session is created.

* Semantic Interaction: Once established, semantic data is exchanged directly between the agents and is explicitly bound to the Session Identifier and Capability Identifier. Each agent enforces the authorized capability scope locally. The protocol does not mandate a specific transport, but confidentiality and integrity protection are expected to be ensured by the underlying secure channel.

* Session Update: If session parameters require adjustment, either agent may request an update. Any modification must remain within the originally authorized scope unless additional authorization is obtained. Updated parameters take effect upon mutual agreement.

* Session Termination: A session is terminated when either agent sends a termination message or when authorization expires. Upon termination, both agents release associated resources and reject further messages referencing the Session Identifier.

```
     Agent A                            Agent B
       |                                   |
       |---- SESSION_INIT ------------------>|
       |                                     |
       |<--- SESSION_ACCEPT -----------------|
       |                                     |
       |========== Semantic Data ===========>|
       |<========= Semantic Data ============|
       |                                     |
       |---- SESSION_TERMINATE ------------->|
       |                                     |
```

Figure 4 Agent-to-Agent Semantic Session Process

5.6. Agent to External Resource Service Process

When an agent requires capabilities that are not internally available within the agent collaboration domain, it interacts with an External Resource Service (ERS) through a standardized agent-to-resource access protocol. The agent first resolves the target resource endpoint and determines whether the requested operation falls within its authorized trust scope. The agent then establishes a secure session with the External Resource Service using an existing resource access protocol, such as MCP (Model Context Protocol), A2T (Agent-to-Tool Protocol), or other deployment-specific interfaces.

The agent includes its identity reference (e.g., AIC or associated credential) in the invocation request. The External Resource Service performs authentication and authorization checks based on its local access control policies. If approved, the requested operation is executed and the result is returned to the agent.

6. Requirements for the protocol suite

The protocol suite defined in this document supports the complete interaction lifecycle of multi-agent collaboration. Each stage of collaboration —— including registration, authorization, capability visibility, semantic resolution, task coordination, and session establishment —— requires specific protocol. From an end-agent perspective, the corresponding protocols collectively form a Interaction Layer, as shown in the figure below.

Before an agent can participate in any collaboration, it must establish a recognized identity and authorized operational scope within the system. This requirement leads to the definition of the Agent Registration Protocol (ARP) and the Agent Authentication and Authorization Protocol (AAAP). ARP governs how an agent formally registers with a locally attached distributed node, while AAAP defines how authentication and authorization decisions are requested and enforced.

Once identity and authorization are established, the system should enable capability visibility across agent gateways. This introduces the need for the Capability Digest and Synchronization Protocol (CDSP), which synchronizes abstracted capability digest information across agent gateways based on gateway-generated ACS records.

For multi-agent task execution, the Task-based Invocation Protocol (TIP) defines how tasks, once decomposed by the application logic, are coordinated across agents. TIP standardizes task state signaling, invocation ordering, and execution control across distributed participants. Importantly, task decomposition algorithms and business workflow reasoning remain within the Application Layer; TIP only governs the distributed coordination of already-defined tasks.

Finally, once authorization and routing decisions have been completed, direct semantic interaction between agents requires session establishment and execution control. This is addressed by the Agent-to-Agent Protocol (A2A), which governs session creation, capability-scoped execution, and termination. A2A enables direct semantic exchange without redefining application logic or semantic interpretation.

Based on the above, the protocol suite consists of the following categories:

* Agent Registration Protocol (ARP) and Agent Authentication and Authorization Protocol (AAAP), which jointly establish agent identity, trust, and operational scope.

* Capability Digest Synchronization Protocol (CDSP), which maintains distributed visibility of agent capability digest across gateways.

* Task-based Invocation Protocol (TIP), which extends semantic routing to multi-agent task decomposition and orchestration.

* Agent-to-Agent Protocol (A2A), which completes the collaboration lifecycle by enabling autonomous, peer-to-peer semantic interaction.

   * Agent to External Resource Service Protocol, which will use
      existing protocol, such as MCP (Model Context Protocol), A2T
      (Agent-to-Tool Protocol), or other deployment-specific interfaces.

```
   +------------------------------------------------------------------+
   |                    Application Layer                             |
   |                                                                  |
   | +------------------+ +------------------+ +------------------+   |
   | | Task             | | Policy and       | | HITL and         |   |
   | | Orchestration    | | Governance       | | Failure Handling |   |
   | | - decomposition  | | - policy rules   | | - escalation     |   |
   | | - state model    | | - dynamic CBAC   | | - fallback/retry |   |
   | +------------------+ +------------------+ +------------------+   |
   +-----------------------------+------------------------------------+
                                 |
                                 | Application-to-Semantic Contract
                                 v
   +------------------------------------------------------------------+
   |                      Semantic Layer                              |
   |                                                                  |
   | +------------------+ +------------------+ +------------------+   |
   | | Ontology and     | | Validation and   | | Natural-Language |   |
   | | Profile Model    | | Rules            | | Intent           |   |
   | | - classes and    | | - consistency    | | Normalization    |   |
   | |   properties     | | - applicability  | | - text to intent |   |
   | | - constraints    | | - deterministic  | | - confidence and |   |
   | +------------------+ +------------------+ |   ambiguity      |   |
   |                                           +------------------+   |
   | +------------------+ +------------------+                        |
   | | Alignment and    | | Knowledge        |                        |
   | | Version          | | Resources        |                        |
   | | Governance       | | - glossary/codes |                        |
   | | - mapping        | | - remediation    |                        |
   | |   confidence     | |   knowledge      |                        |
   | | - compatibility  | |                  |                        |
   | |                  | |                  |                        |
   | +------------------+ +------------------+                        |
   +-----------------------------+------------------------------------+
                                 |
                                 | Semantic-to-Interaction Contract
                                 v
   +------------------------------------------------------------------+
   |                   Coordination Layer                             |
   |       +---------------------+   +---------------------+          |
   |       | Register & Identity |   |  Authentication     |          |
   |       | - ARP               |   |  & Authorization    |          |
   |       |                     |   |  - AAAP             |          |
   |       +---------------------+   +---------------------+          |
```

```
   |       +---------------------+   +---------------------+          |
   |       | Invocation & Session|   | Sync & Resolution   |          |
   |       | - TIP               |   | & Routing           |          |
   |       | - A2A               |   | - CDSP              |          |
   |       +---------------------+   +---------------------+          |
   +------------------------------------------------------------------+
                                 |
                                 v
   +------------------------------------------------------------------+
   |       Transport Connectivity + Security Enforcement              |
   +------------------------------------------------------------------+
```

Figure 5 Unified Layered Architecture and Modules

7. Deployment Example: Fixed Network with Agent Gateway Realization

This section provides a deployment example illustrating how the proposed architecture can be realized in a fixed network environment. The purpose of this example is not to constrain implementation choices, but to demonstrate how distributed nodes defined in the architecture may be instantiated using existing or enhanced network elements.

In a fixed broadband network, distributed nodes can be realized as enhanced Agent Gateway functions deployed at aggregation layers, or service edge points. These nodes are positioned logically between access networks and service domains, enabling them to perform registration, authorization mediation, capability abstraction, and semantic routing functions without requiring changes to end-user access infrastructure.

In such deployments as shown in figure 6:

* Agent registration and authentication are performed via the AG and Agent Management Center (AMC).

* Capability digest exchange occurs between AGs.

* Semantic resolution is handled hop-by-hop across AGs.

* Agents establish peer-to-peer semantic sessions after gateway-mediated coordination.

This example demonstrates that distributed nodes, which encapsulate logical coordination functions, can be embedded into agent gateways in fixed networks.

```
                                    +-------------------------------+
                               +----|    Agent Management Center    |------+
                               |    +-------------------------------+      |
                               |                                           |
      +----------+      +---------+                                  +---------+   +-----
----+
      | Agent A  | ---- | AG 1    |<-------------+    +------------->|  AG 2   |---| Agen
t B |
      +----------+      +---------+              |    |              +---------+   +-----
----+

                                                 |    |
                                             +------------+
                                             |  Routers   |
                                             +------------+
```

                     Figure 6 Deployment Example: Fixed Network with Agent Gateway Reali
zation

8. Deployment Example: Mobile Operator Network

In a mobile operator network (e.g., a 6G network), AI capabilities and technologies are expected to be leveraged subject to operator policies and configurations. AI Agents, which refer generally to agents that autonomously perform tasks on behalf of users, systems, and/or applications, can understand complex requests and improve network efficiency. In addition, capabilities and services such as sensing, real-time data processing, telemetry, analytics, and others within a 6G network may also be provided as “Tools” to third-party applications.

Below figure shows an example of AI agents, an Agent Management Center and an Agent Gateway deployed in the mobile operator network. The architecture of 6G is still in discussion, thus some functionalities are described by using some 5G NFs as typical examples.

User initiates intent in nature language and receive human-readable result. Intent analysis takes place at two stages:

* (Optional, based on UE capabilities) The UE converts natural-language intent to operator-defined intent and also translates operator-defined results to human-readable results based on internal implementation.

* (Mandatory) The network function (NF) with agent capability in the mobile operator core network comprehends and analyzes the intent. Based on the analyzed intent, a subsequent workflow is generated. Typically, the NF that terminates NAS messages (e.g., AMF in 5G) can either comprehend the intent (when the AMF includes agent capability) or forward the intent to an agent-enabled NF in the core network (when the AMF does not include agent capability).

Agent-enabled NFs request AIC allocation from the Agent Gateway. The NF (e.g., AMF in 5G) that terminates NAS messages and manages UE registration may request AIC from the Agent Gateway on behalf of the agent in the UE. The Agent Management Center is responsible for agent authentication and authorization. The authentication and authorization procedure for agents (Agents Agent Gateway Agent Management Center) may reference the authentication and authorization procedure for UEs (e.g., UEs <-> AMF <-> UDM <-> AUSF in 5G).

Agent-enabled NFs register with the Agent Gateway and discover each other by querying the Agent Gateway. The registration and discovery mechanism is similar to the functionality provided by the NRF in 5G networks. However, standardization of agent capabilities is essential for the accuracy and efficiency of agent discovery. The NF (e.g., AMF in 5G) that terminates NAS messages and manages UE registration may interact with the Agent Gateway on behalf of the agent in the UE. However, the discovery of the agent on the UE takes the UE connection state into account.

Service-based interfaces may be enhanced to support agent communication, or agent-based interfaces may be introduced to support AI traffic carrying Agent-to-Agent semantic data. The AI capabilities supported by the mobile operator network may be exposed as agent tools and invoked by third-party AI agents via the Agent Gateway, which performs protocol conversion for different agent communication protocols if necessary, while AI traffic transmitted through the mobile operator network may be identified to guarantee performance requirements.

The NF supporting session management (e.g., SMF in 5G) can be enhanced, or a new NF may be introduced to support the management of Agent-to-Agent semantic sessions between the agent residing in the UE and the agent residing in the operator core network.

```
     +-------+
     |  Uesr |
     +-+-----+
       |  ^
       |  |
       |  |
+------+--+-------+                 +----------------------------------------------------
------------------+                +------------+
|      |  |       |                 |                                                   M
obile Operator    |                | Third Party|
|      v  |       |                 |
  Corenetwork     |                |            |
|  +------+----+  | Operator-defined|
                  |                |            |
|  |  UE APP   |  | <-------------> |    +---------+    +------------+   +-----------+
                  |<-------------> |  +------+  |
|  +--+--------+  |                 |    |   NF    |    | NF with    |   |   Agent   |
                  |                |  | Agent|  |
|     |   ^       |                 |    |        |    |   Agent    |   |  Gateway  |
                  |                |  +------+  |
|     v   |       |                 |    +---+-----+    +-----+------+   +-----+-----+
                  |                |            |
|  +------+----+  |                 |        |               |                |
                  |                |  +------+  |
|  | UE OS Layer  |                 |        |               |                |
                  |                |  | Agent|  |
|  |   Agent   |  |                 |        |               |                |     Servi
ce Based Interface|                |  +------+  |
|  +--+--------+  | Intent Execution|        |               |                |      (Age
nt Enhancement)   |                |            |
|     |   ^       | <-------------> |--------+----+----------+----+-----------+-----+-----
-------------     | <-------------->|     .      |
|     |   |       |      Result     |             |               |                 |
                  |                |     .      |
|     |   |       |                 |             |               |                 |
                  |                |     .      |
|     v   |       |                 |             |               |                 |
                  |                |     .      |
|  +------+----+  |                 |      +------+----+     +----+---+    +-------+-----
+                 |                |     .      |
|  | UE Modem  |  |                 |      |  NF with  |     |   NF   |    |    Agent
|                 |                |            |
|  |           |  |                 |      |   Agent   |     |        |    | Management
|                 |                |  +------+  |
|  +-----------+  |   AI traffic    |      +-----------+     +--------+    |    Center
|                 |    AI Traffic  |  | Agent|  |
|                 | <-------------> |                                      +-------------
+                 | <-------------->|  +------+  |
|                 |     Transfer    |
                  |     Transfer   |            |
| UE              |                 |
                  |                |            |
+-----------------+                 +----------------------------------------------------
------------------+                +------------+
```

Figure 7 Deployment Example: Mobile Operator Network

9. IANA Considerations

TBD

10. Acknowledgement

TBD

11. Normative References

[GW-REQ] L, B., "Gateway Requirements for Dynamic Multi-agents Secured Collaboration. draft-liu-dmsc-gw-requirements. <https://datatracker.ietf.org/doc/draft-liu-dmsc-gw-requirements/>", 16 January 2026.

[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/info/rfc2119>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

Authors’ Addresses

Xueting Li
China Telecom
Beiqijia Town, Changping District
Beijing
Beijing, 102209
China
Email: lixt2@foxmail.com

Jun Liu
Beijing University of Posts and Telecommunications
10 Xitucheng Road, Haidian District
Beijing
100876
China
Email: liujun@bupt.edu.cn

Chenguang Du
Zhongguancun Laboratory
Beijing
100094
China
Email: ducg@zgclab.edu.cn

Lianhua Zhang
AsiaInfo Technologies (China) Inc
Beijing
100000
China
Email: zhanglh2@asiainfo.com