DMSC Working Group
Internet-Draft
Intended status: Standards Track
Expires: 2 August 2026

X. Li
A. Wang
China Telecom

29 January 2026

Dynamic Multi-agents Secured Collaboration Infrastructure Architecture
draft-li-dmsc-inf-architecture-03

Abstract

This document presents an architectural framework for dynamic multi-agent collaboration from an infrastructure perspective. It outlines the network requirements introduced by large-scale agent collaboration, and proposes a systematic approach to enabling Dynamic Multi-agent Secured Collaboration (DMSC) through infrastructure capabilities. The architecture focuses on how network control and forwarding functions can actively participate in agent collaboration.

Table of Contents

1. Introduction

Intelligent agents have evolved rapidly in recent years, driven by advances in artificial intelligence models, computing platforms, and network connectivity. Early forms of agents were typically embedded within isolated systems and designed to perform narrowly defined tasks under predefined conditions. Their interactions with external entities were limited and often mediated by tightly coupled application logic [IoA].

With the increasing availability of large-scale AI models, edge computing resources, and programmable network infrastructures, agents are becoming more autonomous, adaptive, and capable of operating across distributed environments. Modern agents can perceive changes in their environment, make decisions based on local or shared information, and interact with other agents and tools in order to achieve complex objectives. These interactions are no longer confined to static configurations or single administrative domains, but increasingly span devices, networks, and application platforms.

As agents continue to proliferate, they are forming large-scale collaborative systems in which multiple agents dynamically discover each other, exchange information, and coordinate actions. Such systems exhibit highly dynamic behavior, including frequent changes in agent population, roles, and interaction patterns. The resulting agent ecosystems resemble an open, interconnected environment rather than a collection of isolated applications.

The evolution toward large-scale, dynamic agent ecosystems introduces new challenges for the underlying network infrastructure. While agents are capable of sophisticated reasoning and decision-making, their ability to collaborate effectively depends on the availability of common, scalable, and interoperable networking support.

This document focuses on the architectural aspects of enabling dynamic multi-agent collaboration from a network and infrastructure perspective. It examines how network control and forwarding functions can be extended to recognize agents as first-class entities and provide generic support for agent identification, discovery, semantic routing, and coordination. The architecture is intended to support a wide range of agent types, including on-device agents, network-resident agents, and third-party agents, without imposing assumptions about their internal implementation.

The scope of this document is limited to architectural concepts and functional building blocks. It does not define specific protocols, data models, or security mechanisms, nor does it prescribe particular deployment scenarios or application workflows. Instead, it provides a foundational framework upon which more detailed specifications, including protocol designs and security architectures, can be developed in subsequent documents.

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:

* DMSC: Dynamic Multi-agent Secured Collaboration. The framework and infrastructure enabling secure and efficient collaboration among dynamic agents.

* Agent: An autonomous software entity capable of perception, planning, decision-making, and execution.

* SemR: Semantic Routing. The process of routing an Agent request based on the meaning or intent of the request, rather than solely on IP address.

4. Network Requirements

The proliferation of intelligent agents fundamentally reshapes interaction patterns and control dynamics in future networks. Agent interactions are typically short-lived, context-dependent, and driven by task semantics rather than static endpoints. Moreover, agents may dynamically join or leave collaborative groups, migrate across administrative domains, or change roles over time. These characteristics introduce new requirements for network infrastructures, including agent-level identity management, capability-aware communication, scalable registration and discovery, cross-domain collaboration support, and adaptive routing, as also reflected in [draft-yu-ai-agent-use-cases-in-6g].[usecase]

Collectively, these requirements indicate that future networks must go beyond passive connectivity and actively support dynamic multi-agent collaboration. The core idea of Dynamic Multi-agent Secured Collaboration (DMSC) is to elevate key collaboration-related functions into the network infrastructure. Instead of embedding all coordination logic within applications or agent frameworks, DMSC leverages infrastructure-level capabilities exposed through control-plane and forwarding-plane functions. This approach enables the network to recognize agents as first-class entities, maintain high-level collaboration context, and make informed decisions on discovery, routing, and coordination support in a scalable and interoperable manner.

5. DMSC Infrastructure Architecture

5.1. DMSC Infrastructure Architecture

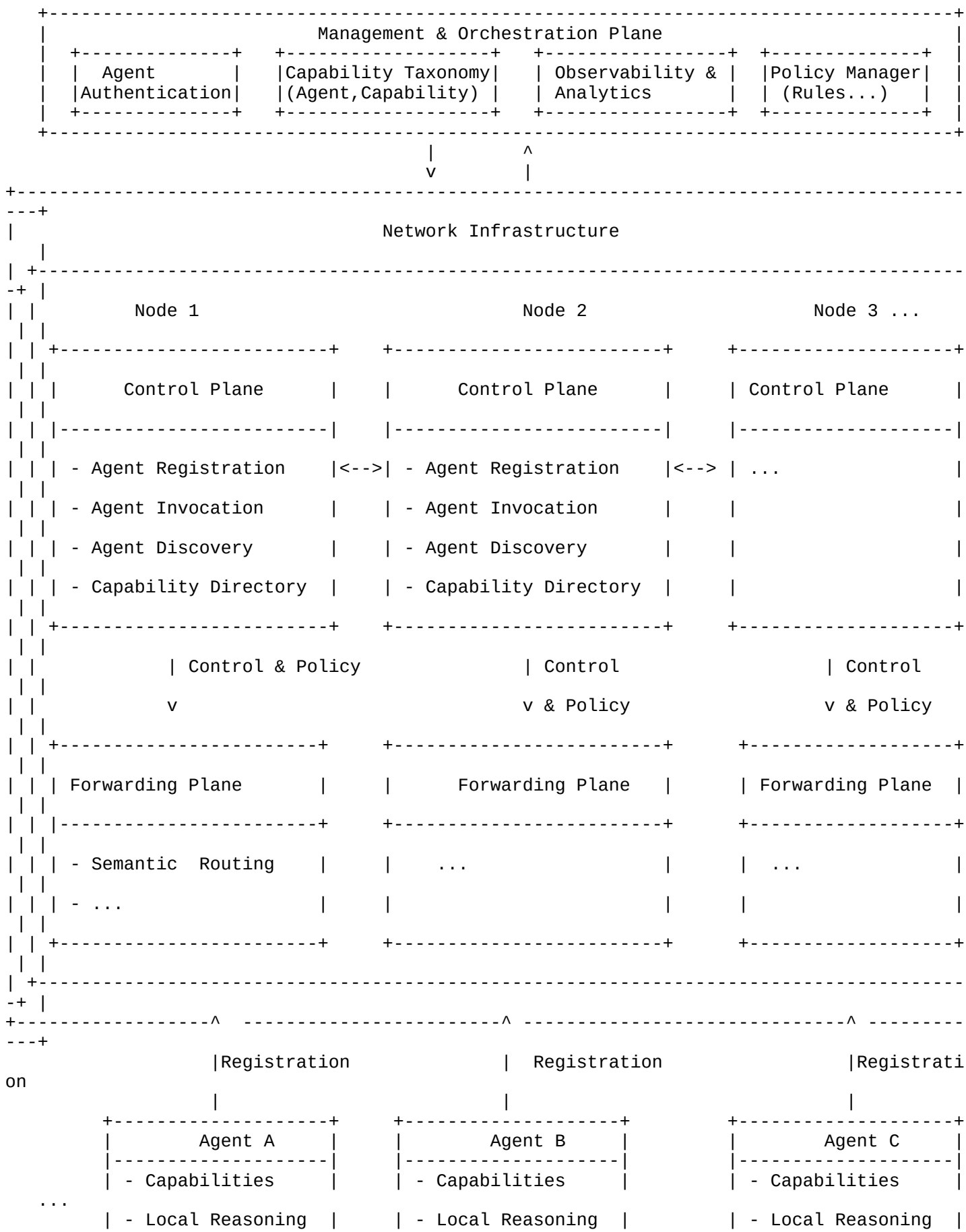
Figure 1 illustrates the overall architecture for dynamic multi-agent collaboration from an infrastructure-centric perspective. The architecture positions the network infrastructure as an active participant in agent collaboration, while preserving the autonomy and task-level reasoning of individual agents. In this architecture, the network does not execute agent logic or interpret task semantics. Instead, it provides generic support functions that enable agents to collaborate more efficiently and reliably. Agents remain autonomous, while the network supplies shared infrastructure capabilities.

From an infrastructure perspective, the architecture is organized into three logical layers:

* Management Plane: governs authentication, capability taxonomy, observability, policies aspects.

* Control Plane: Manages agent registration, discovery, invocation, lifecycle, and capability information maintenance and so on.

* Forwarding Plane: Supports semantic routing for agent interactions.


Management & Orchestration Plane
Agent Authentication
Capability Taxonomy (Agent,Capability)
Observability & Analytics
Policy Manager (Rules...)
Network Infrastructure
Node 1
Node 2
Node 3 ...
Control Plane
- Agent Registration
- Agent Invocation
- Agent Discovery
- Capability Directory
Control & Policy
Control & Policy
Control & Policy
Forwarding Plane
- Semantic Routing
- ...
...
Registration
Registration
Registration
Agent A
Agent B
Agent C
- Capabilities
- Local Reasoning

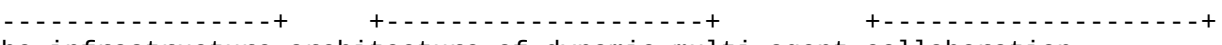


Figure 1 The infrastructure architecture of dynamic multi-agent collaboration

On top of this architecture, agents engage in collaborative activities driven by task intents, shared goals, and capability information. Agents are responsible for local reasoning, decision-making, and execution of task-specific logic. The network does not interpret agent semantics or execute agent logic; instead, it provides common infrastructure capabilities that support efficient and scalable collaboration among agents. Above the network

infrastructure, a Management and Orchestration Plane provides non-real-time management functions, including agent authentication, agent capability taxonomy management, policy management, observability and analytics. This plane supplies policy, trust, and state-related inputs to the network infrastructure.

The network infrastructure itself is composed of multiple network nodes, each implementing a common set of logical functions. Within each node, the Control Plane provides agent-aware control functions, including agent identity management, capability directory maintenance, registration, and discovery control. These functions enable the network to recognize agents as first-class entities and maintain a consistent view of agent-related information across the infrastructure. By decoupling agent identity from physical location through capability identifiers, the control plane supports dynamic agent lifecycle events such as mobility, instantiation, and termination.

The Forwarding Plane supports semantic routing by forwarding requests based on capability identifiers—such as /capability/ocr—rather than static IP addresses. When multiple instances of a capability are available, the forwarding plane may select a target based on real-time health and availability information—such as liveness or load—provided by the control plane. In the event of failure, it can perform fast failover to an alternative instance within the same capability group, ensuring continuity of service.

When an agent joins the network, it authenticates and obtains trusted authorization through network controller; only verified agents may register their capabilities—such as “supports high-precision OCR” or “performs GDPR-compliant de-identification”—which are stored locally to form a capability directory. When another agent issues a capability-based discovery request (e.g., “find an OCR agent”), the local node either responds directly or securely synchronizes capability information with other nodes—including across domains or clouds—to locate eligible candidates. Once a target is identified, the request is forwarded via semantic routing (e.g., using /capability/ocr) to the appropriate instance.

Overall, this architecture establishes a clear division of responsibilities: agents focus on intelligent behavior and task execution, while the network infrastructure provides capability-based control, semantic forwarding, and secure coordination mechanisms. This separation enables dynamic multi-agent collaboration to scale across heterogeneous environments—on-premises, at the edge, or in the cloud—while allowing agents and the network to evolve independently.

6. Infrastructure Functions Enabling Active Network Participation

6.1. Agent Identification and Capability Directory

In large-scale dynamic multi-agent environments, agents cannot be effectively supported using traditional host- or service-based identifiers alone. Agents may be instantiated dynamically, migrate across network locations, or operate concurrently on the same physical node. As a result, the network requires a mechanism to identify agents as logical entities that are decoupled from network topology.

The proposed architecture introduces network-visible agent identifiers that represent agents independently of their physical location or hosting environment. These identifiers enable the network to consistently recognize agents across control and forwarding functions, even as underlying network bindings change. Beyond basic identification, the architecture supports to form agent capability directory on nodes.

6.2. Infrastructure-Level Agent Discovery

Agent discovery is a fundamental prerequisite for collaboration, yet traditional discovery mechanisms are typically designed for relatively static services or tightly scoped environments. In contrast, multi-agent collaboration requires discovery mechanisms that can operate across heterogeneous platforms, adapt to dynamic agent populations, and respect administrative boundaries.

In DMSC architecture, agent discovery is provided as an infrastructure-level function, rather than being entirely implemented within agent frameworks. The network supports discovery queries based on agent identifiers, advertised capabilities. This allows agents to locate suitable collaborators without requiring global knowledge or centralized coordination.

6.3. Semantic Request Routing

Traditional routing mechanisms forward packets based on destination addresses without awareness of application intent. In dynamic multi-agent collaboration, however, interactions are driven by *what* is needed— such as a specific capability— rather than *where* a fixed endpoint resides. The DMSC architecture addresses this by introducing semantic request routing: requests are expressed in terms of agent capabilities (e.g., "/capability/ocr").

Semantic routing enables flexible agent invocation without hard-coded endpoints. A request for a given capability can be routed to any authorized and available instance that has declared that capability. If the selected instance becomes unavailable, the network may fail over to another instance within the same capability group, provided such redundancy exists.

6.4. Secure Collaboration Capability and Policy

Effective collaboration among dynamic agents requires consistent handling of capabilities and policies, especially when interactions span multiple domains or network segments. The DMSC architecture supports secure synchronization of capability declarations and policy constraints at the infrastructure level.

Capability information associated with an agent—such as its declared functions (e.g., OCR, payment validation) and security attributes (e.g., GDPR compliance, authentication requirements)—can be registered with the control plane and synchronized across domains. Where appropriate, these capability descriptions may be bound to policy rules that govern access and interaction. Security-related attributes, such as authorization scope or domain-specific constraints, can be attached to capability entries to ensure that interactions remain compliant with local regulations and trust boundaries. In cross-domain scenarios, policy abstraction mechanisms support controlled translation or normalization to enable interoperability while respecting local governance.

6.5. Operational Visibility

As multi-agent systems scale, gaining visibility into collaboration-level behavior becomes essential for effective operation and troubleshooting. Traditional network observability focuses on flows and endpoints, offering limited insight into agent interactions and coordination dynamics. The DMSC architecture introduces operational visibility at the collaboration level by collecting and exposing key interaction events.

Observable entities include:

* Agent registration and capability declaration.

* Capability-based discovery requests and responses.

* Semantic request routing paths, - Invocation outcomes (success/ failure).

* And their association with network resources and policy enforcement points.

This visibility is not intended to expose agent internals or infer application logic, but to provide sufficient information for monitoring, auditing, root cause analysis, and performance optimization. The collected telemetry can be used by management and orchestration systems to support long-term optimization— such as identifying underutilized capabilities or detecting policy violations. It may also inform policy refinement and infrastructure planning, but does not drive real-time control-plane decisions or forwarding-plane behavior. To address privacy and security concerns, exposure of observable data is controlled through policy mechanisms, ensuring that only authorized parties can access relevant information.

7. Security Considerations

This architecture introduces several security considerations, including risks related to agent identity spoofing, capability misrepresentation, semantic routing manipulation, cross-domain trust inconsistencies, and information leakage through enhanced observability. Detailed security mechanisms are outside the scope of this document.

8. IANA Considerations

TBD

9. Acknowledgement

TBD

10. Normative References

[IoA] L, J., "Internet of Agents Definition, Architecture and Applications. https://aip.openatom.tech/explore/journalism/detail/501037383572131840", October 2025.

[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>.

[usecase] Y, M., "draft-yu-ai-agent-use-cases-in-6g. https://datatracker.ietf.org/doc/html/draft-yu-dmsc-ai-agent-use-cases-in-6g", July 2025.

Authors' Addresses

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

Aijun Wang
China Telecom
Beiqijia Town, Changping District
Beijing
Beijing, 102209
China
Email: wangaj3@chinatelecom.cn