Network Working Group
Internet-Draft
Intended status: Informational
Expires: 3 September 2026

E. Song
Y. Song
S. Zhang
X. Li
J. Zhao
Alibaba Cloud

2 March 2026

Problem Statement and Requirements for Dynamic Multi-agent Secured Collaboration (DMSC)
draft-song-dmsc-problem-statement-00

Abstract

Current LLM-based AI agent systems require each agent to implement communication capabilities (service discovery, encryption) and collaboration logic (e.g., task delegation decisions), leading to code bloat, security risks, and inefficient resource usage in cloud-native and hybrid-cloud deployments. This fragmentation impedes scalable multi-agent application development, especially in multi-tenant scenarios where inconsistent security policies and cross-domain connectivity barriers arise. This document analyzes these challenges and proposes requirements for a Dynamic Multi-agent Secured Collaboration (DMSC) infrastructure. DMSC leverages a centralized gateway layer to offload secured communication, cross-domain connectivity, multi-tenant policy enforcement, and dynamic collaboration assistance - enabling developers to focus solely on agent core functionality while ensuring consistent security, interoperability, and operational efficiency across heterogeneous environments.

Requirements Language

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] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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/.

Copyright Notice

Table of Contents

1. Introduction

The proliferation of LLM-based AI agents in cloud-native and hybrid-cloud environments has intensified the need for robust, scalable inter-agent collaboration. Current frameworks (e.g., AutoGen, CrewAI) require developers to embed communication protocols, service discovery mechanisms, and rudimentary collaboration logic directly into agent codebases. This tight coupling increases cognitive load, introduces security vulnerabilities through inconsistent implementations, and hinders scalability in multi-tenant deployments. For instance, an e-commerce agent system spanning public cloud (for customer interaction) and private data centers (for sensitive inventory) must manually handle network tunneling, certificate rotation, and capability matching across domains. Dynamic Multi-agent Secured Collaboration (DMSC) proposes a dedicated infrastructure layer decoupling communication and collaboration concerns from agent business logic. The centralized gateway handles: (1) secured transport (end-to-end encryption, mutual TLS termination), (2) cross-domain network bridging (protocol translation, firewall traversal), (3) multi-tenant policy enforcement (isolation, rate limiting), and (4) dynamic collaboration assistance (capability-based routing, load-aware delegation). By offloading these concerns, DMSC enables agents to remain lightweight, accelerates development cycles, and ensures consistent security posture across heterogeneous deployments.

2. Problem Statements

2.1. Tight Coupling of Collaboration and Communication Logic

Agents must implement service registration, discovery, retry mechanisms, and dynamic task delegation decisions (e.g., "which agent handles this ambiguous query?"). This increases development complexity and error-proneness [AutoGen]. In practice, developers spend up to 40% of implementation effort on communication plumbing rather than core agent logic. For example, an agent handling customer support queries must embed logic to discover available "billing" or "technical" specialist agents, validate their current load, and retry failed delegations. This duplication across agent systems leads to inconsistent behavior, version skew during updates, and heightened maintenance costs. Furthermore, embedding capability-matching logic within agents prevents centralized optimization (e.g., global load balancing across agent pools).

2.2. Security Fragmentation

Each agent independently handles encryption, certificate management, and authentication checks. Inconsistent implementations create security gaps, especially in multi-tenant hybrid-cloud deployments. A survey of 15 open-source agent frameworks revealed 7 distinct TLS configuration patterns, with 30% lacking certificate pinning and 25% using hardcoded credentials. In cross-organization collaborations (e.g., healthcare agents sharing anonymized data across institutions), fragmented security enforcement complicates compliance with regulations like HIPAA or GDPR. Agents deployed at edge locations (e.g., IoT devices) often lack resources for robust crypto operations, forcing trade-offs between security and performance. Centralized security policy management is absent, making audit trails and incident response fragmented across agent logs.

2.3. Inefficient Multi-Tenant Management

Configuring tenant isolation policies and cross-cloud connectivity requires manual, error-prone updates across all agents. Centralized policy enforcement is lacking. In a SaaS platform hosting 100+ enterprise tenants, each tenant’s agents require unique network policies (e.g., "Tenant A agents cannot communicate with Tenant B"). Today, these policies are hardcoded into agent configurations or managed via fragile external scripts. During tenant onboarding/ offboarding, operators must update every agent instance—a process taking hours and risking configuration drift. Cross-cloud scenarios (e.g., agents in AWS communicating with agents in Azure) exacerbate this: network security groups, DNS mappings, and certificate trust stores must be synchronized manually. A single misconfiguration can lead to data leakage or service disruption, as observed in 12% of multi-tenant agent deployments per industry incident reports.

2.4. Lack of Dynamic Collaboration Assistance

When an agent cannot resolve a task, it must implement custom logic to select collaborators. This leads to duplicated effort and suboptimal routing across agent systems. Current approaches range from static routing tables (inflexible to agent churn) to broadcast queries (inefficient at scale). For instance, a legal research agent receiving a query about "EU data privacy laws" must independently determine whether to delegate to a "GDPR specialist" or "Schrems II expert" agent—without visibility into their current workload, expertise depth, or availability. This results in uneven load distribution (some agents overloaded while others idle) and degraded user experience due to latency from sequential delegation attempts. Without infrastructure-level capability indexing and real-time health monitoring, agents cannot leverage global context for optimal

collaboration decisions.

3. Requirements for DMSC

3.1. Non-Intrusive Agent Integration

Agents communicate via standard protocols (HTTP/gRPC); traffic interception MUST be transparent (e.g., iptables, eBPF). Agent code modification MUST NOT be required. The infrastructure SHOULD support zero-trust onboarding where agents register capabilities via secure metadata endpoints without embedded SDKs. For legacy agents, protocol adapters (e.g., REST-to-gRPC translators) MAY be deployed at the gateway to normalize communication. This ensures seamless adoption across greenfield and brownfield agent deployments while preserving developer autonomy over agent implementation languages and frameworks.

3.2. Centralized Gateway for Secured Collaboration

The gateway MUST handle: service discovery with real-time health monitoring; mutual TLS termination and certificate lifecycle management; tenant-scoped policy enforcement (network isolation, rate limiting, data tagging); cross-domain protocol bridging (HTTP/2 to MQTT translation for edge agents); and dynamic collaboration assistance including capability-based routing (matching query intent to agent expertise metadata), load-aware delegation, and circuit breaking for failed agents. The gateway SHOULD maintain a global capability registry indexed by semantic tags (e.g., "finance", "low-latency") and update routing decisions based on real-time metrics (CPU load, queue depth). This transforms the gateway from a passive proxy into an active collaboration orchestrator.

3.3. Multi-Tenant Isolation

Tenant data and policies MUST be cryptographically isolated using tenant-specific encryption keys and namespace separation. Configuration updates SHOULD be tenant-scoped to minimize control-plane overhead. The infrastructure MUST prevent tenant policy leakage (e.g., Tenant A’s rate limits must not affect Tenant B). For cross-tenant collaborations (e.g., partner integrations), explicit policy whitelists MUST be required. Audit logs MUST include tenant identifiers to enable compliance reporting. This isolation model supports both strict separation (for regulated industries) and controlled sharing (for consortium deployments).

3.4. End-to-End Security Offloading

All inter-agent traffic MUST be encrypted in transit using TLS 1.3 or equivalent. The gateway MUST handle certificate lifecycle management (issuance, rotation, revocation) and authentication (OAuth 2.0, mTLS). Sensitive information (PII, credentials) SHOULD be avoided in agent payloads; where unavoidable, the gateway MAY provide data masking capabilities. The infrastructure MUST generate immutable audit trails for all collaboration events (delegation decisions, policy violations). This offloading reduces agent attack surface, ensures cryptographic best practices, and simplifies compliance certification for agent developers.

3.5. Platform-Agnostic Deployment

DMSC MUST support agents deployed across Kubernetes clusters, VMs, bare-metal servers, and edge devices in hybrid-cloud topologies. Gateway deployment options MUST include centralized (for tight control), regional (for latency optimization), and embedded (for air-gapped environments). The data plane SHOULD leverage hardware acceleration (SmartNICs, DPUs) where available to minimize latency overhead. Configuration APIs MUST be consistent across deployment models to enable unified management. This flexibility accommodates diverse operational constraints— from cloud-native startups to regulated enterprises with on-premises requirements.

4. Security Considerations

This information document introduces no any extra security problem to the Internet.

5. Acknowledgement

TBD.

6. IANA Considerations

None.

7. References

7.1. 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/info/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/info/rfc8174>.

7.2. Informative References

[AutoGen] Microsoft, "AutoGen: Enabling Next-Gen LLM Applications", Online https://microsoft.github.io/autogen/, 2023.

[CrewAI] Team, C., "CrewAI Framework Documentation", Online https://crewai.com/, 2024.

[I-D.li-dmsc-architecture] Li, X., "Architecture for Distributed Multi-agent Secured Collaboration", Work in Progress, Internet-Draft, draft-li-dmsc-architecture-00, 2024, <https://datatracker.ietf.org/doc/html/draft-li-dmsc-architecture-00>.

[VPC-Lattice] Services, A. W., "AWS VPC Lattice", Online https://aws.amazon.com/vpc/lattice/, 2023.

Authors' Addresses

Enge Song
Alibaba Cloud
Alibaba Beijing Chaoyang Science & Technology Park
Beijing
100124
China
Email: enge.seg@alibaba-inc.com

Yang Song
Alibaba Cloud
Alibaba Beijing Chaoyang Science & Technology Park
Beijing
100124
China
Email: song288954@alibaba-inc.com

Shaokai Zhang
Alibaba Cloud
Alibaba Beijing Chaoyang Science & Technology Park
Beijing
100124
China
Email: shaokai.zsk@alibaba-inc.com

Xing Li
Alibaba Cloud
Alibaba Beijing Chaoyang Science & Technology Park
Beijing
100124
China
Email: lixing.lix@aliyun-inc.com

Jiangu Zhao
Alibaba Cloud
Alibaba Beijing Chaoyang Science & Technology Park
Beijing
100124
China
Email: jiangu.zjg@alibaba-inc.com