network working group
Internet-Draft
Intended status: Informational
Expires: 1 June 2026

B. Liu
N. Geng
X. Shang
Q. Gao
Z. Li
Huawei Technologies
J. Gao
CAICT

28 November 2025

Requirements for Agent Gateway
draft-liu-rtgwg-agent-gateway-requirements-01

Abstract

This document discusses the requirements for introducing Agent Gateways into Agent-to-Agent communications for better scalability, communication efficiency, and security etc. This document also discusses the gaps of current hardware/software gateways that could not fulfil the task, so that a new kind of entity, e.g. Agent Gateway, is needed.

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 1 June 2026.

Table of Contents

1. Introduction

The Internet of Agents (IoA) is driving a shift in the communication model from the traditional host-to-host (or client-server) model to an Agent-to-Agent interaction model.

One of the core characteristics of an Agent is autonomy, which consequently generates a demand for highly dynamic networking. Introducing Agent Gateways that acting as a crucial intermediary layer is a good approach to enable efficient Agent discovery, information distribution among Agents, Quality of Service (QoS) and security assurance for Agent communications.

This document discusses the requirements for introducing Agent Gateways into Agent-to-Agent communications in terms of what benefits could be brought by the Agent Gateways.

In theory, Agents could be interconnected without any intermidiate Agent-specific gateways. However, significant chanllenges (including issues of scalability, security, communication efficiency etc.) could rise if Agents just interconnect in a fully open and un-managed way.

1.1. Requirements Language

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.

2. Functional Requirements of Agent Gateway

2.1. Delegated Agent Discovery

In dynamic, heterogeneous, and variable-scale AI Agent environments, traditional discovery and addressing mechanisms based on static configuration or simple service names are no longer sufficient to meet the demand.

The complexity of Agent communication requires the Agent Gateway to provide higher levels of abstraction and intelligent services. This subsection discusses two critical enhanced discovery and addressing mechanisms: Agent ID based addressing and task decomposition based Agent discovery.

2.1.1. ID-based Agent Location Resolution

In an Agent network, every Agent should be assigned a globally unique, persistent, and infrastructure-agnostic identifier, known as the Agent ID. One of the core functions of the Agent Gateway is to act as the resolver for this unified namespace, providing transparent addressing services for Agents:

* Agent ID Registration (Optional): Each Agent, upon startup, can delegate the ID registration task to the Agent Gateway, thereby eliminating the need to specially implement a set of protocols and logic for communicating with a registration server.

* Addressing Request: When an Initiator Agent needs to communicate with a Target Agent, it does not need to know any network details of the target, such as its IP address. It only needs to specify the Target Agent’s ID in the communication request.

* Address Resolution and Routing: The Agent Gateway receives the request and queries its local or external servers for the latest reachability information corresponding to the Agent ID. Subsequently, the gateway is responsible for establishing or forwarding the communication flow, accurately routing the message to the Target Agent. This process may involve complex low-level operations such as protocol translation and Network Address Translation (NAT) traversal.

This mechanism offers the following benefits:

* Decoupling and Simplification: Agent developers can focus on their business logic without having to deal with network-layer IP address resolution, achieving complete decoupling between business logic and the communication infrastructure.

* Security and Policy Enforcement: The Gateway, acting as a central control point, can enforce security policies during the addressing resolution phase, such as verifying whether the initiator has permission to communicate with the target, thereby embedding security control at the very beginning of communication establishment.

2.1.2. Task-based Agent Searching

In many collaborative scenarios, the Initiator Agent may not know which specific Agent it needs to interact with but rather has a high-level task objective to accomplish. This necessitates the Agent Gateway providing an Intent-Based or Capability-Based discovery mechanism:

* Capability Registration (Optional): In addition to the Agent ID, each Agent must declare a set of "capabilities" it provides to the Gateway upon registration. These capabilities can be described using structured attributes and should follow a common capability model schema.

* Task Declaration: The Initiator Agent submits a task request to the Gateway, such as "translate this Chinese document into English" or "analyze the buildings in this satellite image." This request is a declaration of the desired outcome, rather than a call to a specific service.

* Task Decomposition and Capability Matching: The Agent Gateway either contains or can access a task decomposition engine. For complex tasks, the gateway can break them down into a series of sub-tasks (e.g., "document parsing" -> "Chinese-to-English translation" -> "format re-structuring"). Subsequently, the gateway searches its registration repository for Agents whose capability descriptions match each (sub) task.

* Agent Recommendation or Orchestration: The Gateway returns the list of discovered Agents that can satisfy the task requirements to the Initiator Agent, which then selects one. In a more automated mode, the Gateway can directly act as an Orchestrator, routing the task request to the most suitable Agent, or even coordinating multiple Agents to collectively complete a complex task chain.

Gateway-based mechanism offers the following benefits:

* Flexibility for Local Collaboration: For Agents to break away from statically orchestrated workflows or service call chains, they usually rely on the inference and orchestration capabilities of Large Language Models (LLMs). However, in certain security/ privacy-sensitive or highly localized scenarios, some Agents may only be available locally. LLMs cannot discover these Agents and thus cannot orchestrate their tasks. By using a locally deployed Gateway, local Agents, once their capabilities are registered, can be automatically included in the scope of task consideration. This allows Agent ecosystems that are only deployed locally to dynamically adapt and evolve.

* Increased Abstraction Level: The interaction between Agents is elevated from the level of "how to call" to "what needs to be done," significantly simplifying the programming model for complex tasks. This is equivalent to an Agentic upgrade of the traditional TCP/IP Socket interface.

* Compute-Network Integrated Load Optimization: Through mechanisms such as CATS (Compute-Aware Traffic Scheduling, or similar), the Gateway can select the optimal Agent from multiple Agents with the same capability to execute a task, based on comprehensive network and compute factors like bandwidth, latency, and load. It is difficult for LLMs alone to dynamically perceive the network and compute load status of individual Agents.

2.2. Agent Communication Access Control

The Agent Gateway can implement fine-grained, context-based access control policies, ensuring that interactions only occur between authorized Agents and in permitted ways. The execution of access control policies is based on context information across multiple dimensions:

* Identity-Based Access Control (IBAC): This is the most fundamental policy. Policy rules can be defined as: "Agent A is allowed to communicate with Agent B," or "Agents belonging to 'Department X' are allowed to access Agents of 'Database Y'." Before routing a message, the Gateway verifies whether the initiator ID and the target ID are within the allowed communication matrix.

* Capability-Based Access Control (CBAC): This policy introduces the dimensions of intent and capability on top of identity. Policy rules can be more dynamic and semantic, for example: - Only Agents with the 'Financial Data Analysis' capability can access the 'Core Financial Database' Agent. - An Agent requesting the 'Medical Diagnosis' capability must itself possess the capability tag for 'Compliant Patient Data Handling' before it can call the corresponding diagnosis Agent. This approach ensures that Agents are not only discoverable but are also used within the correct, authorized context.

* Attribute-Based Access Control (ABAC): This is a more generalized model that can make dynamic decisions by combining identity, capability, and environmental attributes (such as time, communication frequency, data sensitivity tags of the request, etc.). For example: "Only Agents with 'High-Level' security certification, coming from a secure internal network, and operating during working hours, are allowed to call system administration Agents."

Gateway-based mechanism offers the following benefits:

* Principle of Least Privilege: Through fine-grained policy definition, it is ensured that each Agent only possesses the minimum communication privileges necessary to complete its task, significantly reducing the attack surface.

* Dynamic Policy Enforcement: The Gateway, as the policy enforcement point, can intercept and evaluate every communication attempt in real-time, ensuring policy consistency in a dynamic environment.

* Enhanced Ecosystem Security: By shifting access control from the application layer down to the communication layer, a security perimeter near the edge is provided for the entire AI Agent ecosystem. This manages risks at the source, thereby enhancing the security of the Agent network and simultaneously reducing the overall cost of protection.

2.3. Information Distribution among Agents

This section describes the information distribution of AI agents from two dimensions. One dimension is the number of communication participants, which is divided into Point-to-Point Communication (2 AI agents) and Group Communication (3 or more AI agents), and the section is divided into two sub-sections based on this dimension.

This section specifically discusses the content that Agent Gateways are involved in both cases.

2.3.1. Point-to-Point Distribution

In the Point-to-Point information distribution process, the typical processing performed by a gateway includes, but is not limited to:

* Application Layer Proxy (to facilitate monitoring/auditing of AI agent communication behavior, or to hide AI agent identity, etc.)

* Relay (to forward communication messages, making cross-domain communication easier, etc.)

* Traffic aggregation (to provide a tree-structured traffic regulation, improving communication efficiency).

2.3.2. Pub/Sub

One AI agent publishes the information to the gateway, and the gateway then distributes the information to the subscribing Agents based on their Subscribe status. At the application layer, Pub/Sub is a common and efficient method of information distribution, especially suitable for large-scale group communication scenarios.

3. Gap Analysis of Current Gateways

Despite the existence of various forms of gateway devices and software in current networks, their primary design goals revolve around traditional network forwarding, network protocol translation, security isolation, and load balancing. When faced with the unique requirements of AI Agent communication, these existing solutions exhibit significant shortcomings in both architecture and

functionality. This chapter will analyze why existing hardware and software gateways fail to meet the aforementioned Agent communication needs.

3.1. Hardware form Gateways

Common hardware gateway forms include, but are not limited to: network routers/switches, firewalls, dedicated protocol translation gateways, and broadband access equipment (BRAS/BNG). These devices are typically deployed as dedicated hardware, focusing on processing at the Network and Transport layers.

The core limitation of hardware gateways lies in their low processing layer. They focus on network packets rather than Agent entities possessing identity and intent, and thus lack the necessary abstraction and computational power to handle high-level semantics and dynamic relationships. Hardware gateways often utilize processor architectures, such as Network Processors (NPs), that are optimized for network-layer computation. Consequently, they are difficult, if not impossible, to extend to support the high-level, semantic-based logical operations required for Agent communication.

3.2. Software form Gateways

Common software gateway forms include: API Gateways and Cloud Load Balancers. These gateways are widely used in cloud-native environments and operate closer to the application layer, but their design paradigm remains tightly coupled with the architecture of services or API microservices.

Software gateways are closer to meeting the requirements than hardware gateways, but their fundamental problem lies in the limitations of the fixed "service" paradigm. They treat backends as relatively static services defined by APIs, rather than a dynamic collective of Agents possessing intent and collaborative capabilities. Their security model and discovery mechanisms lack native support for the Agent’s autonomy, dynamism, and capability semantics.

4. Agent Gateway Implementation Considerations

TBD.

5. Security Considerations

TBD

6. IANA Considerations

This document has no IANA actions.

7. Acknowledgements

TBD

8. 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>.

Authors' Addresses

Bing Liu
Huawei Technologies
No. 156 Beiqing Road
Beijing
China
Email: leo.liubing@huawei.com

Nan Geng
Huawei Technologies
No. 156 Beiqing Road
Beijing
China
Email: gengnan@huawei.com

Xiaotong Shang
Huawei Technologies
No. 156 Beiqing Road
Beijing
China
Email: shangxiaotong@huawei.com

Qiangzhou Gao
Huawei Technologies
No. 156 Beiqing Road
Beijing
China
Email: gaoqiangzhou@huawei.com

Zhenbin Li
Huawei Technologies
Beijing
China
Email: robinli314@163.com

Jing Gao
CAICT
Beijing
China
Email: gaojing1@caict.ac.cn