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

K. Yao
China Mobile
Z. Sarker
Nokia

2 March 2026

Problem Space Analysis of AI Agent Protocols in IETF
draft-yao-catalist-problem-space-analysis-01

Abstract

This document aims to identify IETF-relevant problem space and potential areas and working groups, exploring internal and external coordination for AI Agent protocols by analyzing open source efforts. It may serve as a target for CATALIST BoF discussions.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

With the rapid development of AI technology, AI Agents are becoming key Internet interaction entities, driving growing demand for Agent-to-Agent (A2A) and Agent-to-Tool (A2T) interworking. Open source projects like A2A, Model Context Protocol (MCP) are actively advancing related protocols with focused use cases. While these efforts lay a preliminary foundation, there are still some missing pieces and potential protocol design aspects that should be handled by open standardization body like IETF to ensure global interoperability.

IETF has held multiple side meetings on AI agent protocol during IETF 123 and IETF 124 meetings, bringing discussions over AI agent identity and identifier, discovery, interaction, authorization, and multi-modal transport. These meetings clarified key directions and highlighted standardization urgency.

Coordinating A2A list of efforts (CATALIST) BoF meeting is planned in IETF125 meeting to facilitate consensus on the actual scope that IETF should work on, figure out potential area(s) and working group(s) to proceed the work, and explore coordination activities in and out IETF.

This document does not propose any detailed solution or protocol, but tries to propose the problem space that IETF should care about by analyzing existing open source projects efforts. This document may serve as a target document for CATALIST BoF meeting discussion.

2. Definition of Terms

** AI Agent: An autonomous, adaptive intelligent software system that uses AI to complete a specific task. While doing so it makes decisions, executes actions, and interacts with other Agents, tools, or humans.

** A2A: Agent-to-Agent, Interconnection and interaction between AI Agents (data transmission, context sharing, collaboration) standardized by dedicated protocols for cross-vendor interoperability.

** A2T: Agent-to-Tool, Interaction between AI Agents and external tools (APIs, databases, etc.), focusing on standardizing tool invocation to leverage external resources efficiently.

3. Problem Space Issue 1: Inter-domain Discovery

3.1. A2A Coverage

Existing A2A protocol (as analyzed from available open source schema definitions [A2A-spec]) provides a foundational discovery mechanism centered on the "Agent Card" construct, which encapsulates critical metadata foragent identification and interaction:

** Core Metadata: Agents advertise identity (name, version, provider), capabilities, skills, authentication requirements, input/output modes, and communication interfaces (URLs, protocol bindings) via the Agent Card.

** Static Retrieval: Protocols support direct retrieval of Agent Card metadata via dedicated requests (e.g., Get Agent Card Request), enabling clients to obtain necessary information to initiate communication.

** Tenant Differentiation: A "tenant" field supports basic multi-tenancy, allowing agents to serve multiple isolated groups within a single administrative domain.

** Extension Points: Agent Extension allows agents to declare custom protocol extensions, enabling domain-specific discovery metadata.

3.2. MCP Coverage

MCP is a typical A2T protocol. Since MCP connects tools to the agent, the developer will have to know beforehand the tools urls and thus it is done in a manual local configs manner. The usual approch is to provide resource links as a primary application entry point. Existing MCP protocol (as analyzed from available open source schema definitions [MCP-spec]).

3.3. Gaps and Potential Work Space in Open Internet

The current discovery mechanisms are insufficient for open Internet deployments, where agents and clients operate across administrative domains, lack pre-configured knowledge of each other, and require dynamic, secure discovery. Current A2A protocol allow three types of extension on discovery mechanisms. A Well-known URI labelled by server domain, registry or catalog based approach, and direct configuration. Based on this, in open Internet, the following should be considered:

** Dynamic Directory Services: Open Internet scenarios require agents to be discoverable via standardized directory services or registries. The current model relies on clients having prior knowledge of an agent’s URL to retrieve its Agent Card, preventing "directory-based discovery" of unknown agents.

** Cross-Domain Addressing: There is no standardized mechanism for resolving agent identifiers to network locations across domains.

** Domain Identification and Trust: Protocols lack standardized "domain" identifiers (e.g., organizational ID, network domain) and mechanisms to express cross-domain trust relationships. Clients cannot easily determine an agent’s domain or whether their local domain trusts it.

** Dynamic Metadata Synchronization: Agent Card updates (e.g.,capability changes, endpoint updates) are not propagated across domains. Cross-domain clients may rely on stale metadata, leading to failed interactions.

4. Problem Space Issue 2: End-to-End Session State Management

4.1. A2A Coverage

Existing A2A protocol creates a "TASK" object struct, which serves as the core unit of session management, providing a robust foundation for tracking interaction lifecycles between AI agents:

** Task Object: A Task aggregates all session-related state, including a unique id (task_ID), status (the current status of a Task, including state and a message), history (message log), artifacts (task outputs),and contextId (Unique identifier for the contextual collection of interactions).

** Interaction State Machine: A comprehensive state machine(SUBMITTED, WORKING, COMPLETED, FAILED, CANCELED,INPUT_REQUIRED, AUTH_REQUIRED, REJECTED) covers key interaction scenarios, including user input prompts and authentication interruptions.

** Synchronous/Asynchronous/Streaming Modes: Protocol supports synchronous requests, asynchronous requests (via "pushNotifications"), and streaming responses for incremental results.

4.2. MCP Coverage

MCP is stateful per connection and stateless across connections. It does not support session resumption and native session timeouts, applicaitons can retry and that it implementaiton specific.

4.3. Gaps and Potential Work Space in Open Internet

While the core session model is relatively robust, open Internet deployments impose additional requirements for reliability, and interoperability across heterogeneous implementations:

** Session Timeout and Expiration: A2A Protocol lacks standardized session timeout, idle timeout, and expiration mechanisms. Servers cannot automatically clean up stale sessions, leading to resource leaks, and clients cannot reliably determine if a session is still valid.

** Context Propagation Rules: While contextId supports cross-task context, A2A protocol does not standardize how context is inherited (e.g., which fields are carried over to new tasks), truncated (e.g., handling long message histories), or merged (e.g., combining contexts from multiple agents). This leads to inconsistent behavior across implementations.

** Session Recovery and Reconnection: The protocol lacks detailed mechanisms to recover sessions after network disconnections. Clients cannot resume streaming responses, confirm the last received message, or continue partial task execution. Current designs assume best-effort reconnection semantics and do not specify maximum recovery windows or reconnection deadlines. This gap is particularly critical for long-running or mission-critical agent workflows that rely on uninterrupted session continuity.

** User-Session Binding: Protocols only support tenant isolation but lack standardized user identity fields. This prevents user-level session isolation, cross-device session synchronization, and user-specific session management.

** Extended State Semantics: The state machine lacks semantics for long-running interactions, such as SUSPENDED (temporarily paused), or PENDING_EXTERNAL (e.g., waiting for a response from an external system). This forces long-running tasks to remain in WORKING state, leading to ambiguous semantics.

** Session level constrains: While A2A allows message metadata to provid a deadline, it does not have any scheduling gurantee or univarsal deadline. Session-level Quality-of-Service (QoS) attributes, such as maximum task-completion deadlines, bounded execution times, or task priority levels attached to session operations left unclear or to the application implementaiton.

5. Problem Space Issue 3: Fine-Grained Authorization

5.1. A2A Coverage

Existing A2A protocol provides a foundational authorization framework covering high-level access control requirements:

** OAuth Scope Support: OAuth 2.0 flows support coarse-grained permission grants.

5.2. MCP Coverage

MCP incorporate transport level authorization using OAuth 2.1 and OAuth 2.0 protocol. However, it implements a subset of protocol features for simplicity.

5.3. Gaps and Potential Work Space in Open Internet

The current authorization framework is insufficient for open Internet deployments, where cross-domain access, fine-grained resource control,and dynamic trust relationships are required:

** Resource-Level Authorization: Protocols only support agent-level authorization. There is no mechanism to enforce permissions at the resource level (e.g., Task, Artifact, or Message), preventing use cases such as "allow read access to this task but not that one".

** Delegation Authorization: Cross-domain and multi-agent scenarios require delegation (e.g., Agent A acting on behalf of a user to access Agent B). Protocols lack standardized delegation mechanisms, including delegation scope, time limits,and revocation.

** Cross-Domain Permission Propagation: When an agent delegates a task to a cross-domain agent, there is no mechanism to propagate permissions in a controlled manner (e.g., "Agent A can access Agent B’s read skill on behalf of the user, but not write"). This leads to either over-privileged delegation or failed cross-domain interactions.

** Authorization Auditing: There is no standardized mechanism to log authorization events (e.g., who accessed what resource, when, with what permission). This hinders compliance with regulatory requirements and security incident investigation.

6. Problem Space Issue 4: Multi-Modal Transport

6.1. A2A Coverage

Existing A2A protocols provide a foundational multi-modal transmission framework centered on the "part" construct, enabling exchange of diverse data types:

** Unified Multi-Modal Carrier: The "part" construct supports multiple data types, including text, binary data, etc., with "mediaType" to indicate the data format(e.g., text/plain, application/json, image/png).

** Streaming Multi-Modal Transmission: The protocol supports incremental transmission of multi-modal data, including (e.g.,streaming video frames, incremental text + images).

6.2. MCP Coverage

MCP is desinged primarity for text based JSON-RPC communication.

6.3. Gaps and Potential Work Space in Open Internet

While the core multi-modal framework is functional, open Internet deployments require additional support for large data, dynamic adaptation, and interactive use cases:

** Large File and Chunked Transmission: There is no support for chunked upload/download of large multi-modal data (e.g., videos, high-resolution images). The raw field uses base64 encoding for binary data, which is inefficient for large files, and there is no mechanism for hash verification. ** latency bounded transmission: The protocols lack mechanism for predictable/controlled ordering and loss handling of task for critical agent message. As a result, timing-sensitive agent behaviors (e.g., cooperative planning loops) cannot rely on predictable inter-agent message timing.

** Ordering, and loss-handling of messages : MCP and A2A both lack mechanisms for conveying message ordering requirements across multi-modal data transmission. Similarly, no mechanisms exists to distinguish between messages that e.g. must be reliably delivered, those that may be dropped or superseded.

** Message importance/scheduling support : While MCP has non-interperable way to annotate priority as a tool argument, both MCP and A2A left prioritization and scheduling of message and task completion to the host of the application or agent orchestrator, not in the protocol. This would be a potential feature required in cross domain functioning of agents.

7. Security Considerations

Beyond identity authentication and authorization, Agent interconnection faces additional security challenges that require IETF attention to ensure ecosystem security and trustworthiness.

** Data Encryption: All Agents interaction data (context, task requests, results) must be encrypted in transit and at rest to prevent tampering. The IETF should enforce encryption requirements for multi-modal data and ensure compatibility with existing TLS standards.

** Anonymity and Privacy: Agent interactions may involve sensitive user/Agent data. The IETF should investigate privacy-preserving mechanisms to protect data while enabling effective interconnection.

** Malicious Agent Mitigation: Malicious Agents may launch prompt injection, or spoofing attacks. The IETF should investigate attack detection and mitigation mechanisms.

8. IANA Considerations

TBD.

9. Acknowledgements

10. Informative References

[A2A-spec] "A2A Specification", n.d., <https://a2a-protocol.org/latest/definitions/>.

[MCP-spec] "MCP Specification", n.d., <https://modelcontextprotocol.io/specification/2025-11-25/basic/authorization>.

Authors' Addresses

Kehan Yao
China Mobile
Email: yaokehan@chinamobile.com

Zaheduzzamam Sarker
Nokia
Email: zaheduzzaman.sarker@nokia.com