Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 24 January 2026

D. Liu, Ed.
Alibaba

23 July 2025

Agent Context Protocol
draft-liu-agent-context-protocol-00

Abstract

This specification defines a standard message and protocol for communicating Agent Context information between AI agents.

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 24 January 2026.

Copyright Notice

Table of Contents

1. Introduction

In addition to data and content, the interaction between AI agents should also include the communication of agent context information. This context information is crucial for the called AI agent to generate more accurate content and complete tasks that meet user needs while adhering to the security and policy configurations of the application and internal organizations.

Considering that there may be multiple coexisting protocols between AI agents, and the ways in which AI agents call each other are diverse, it is necessary to maintain semantic consistency and standardization in the transmission of agent context information. This is critical for improving interoperability among AI agents and building a scalable AI agent ecosystem.

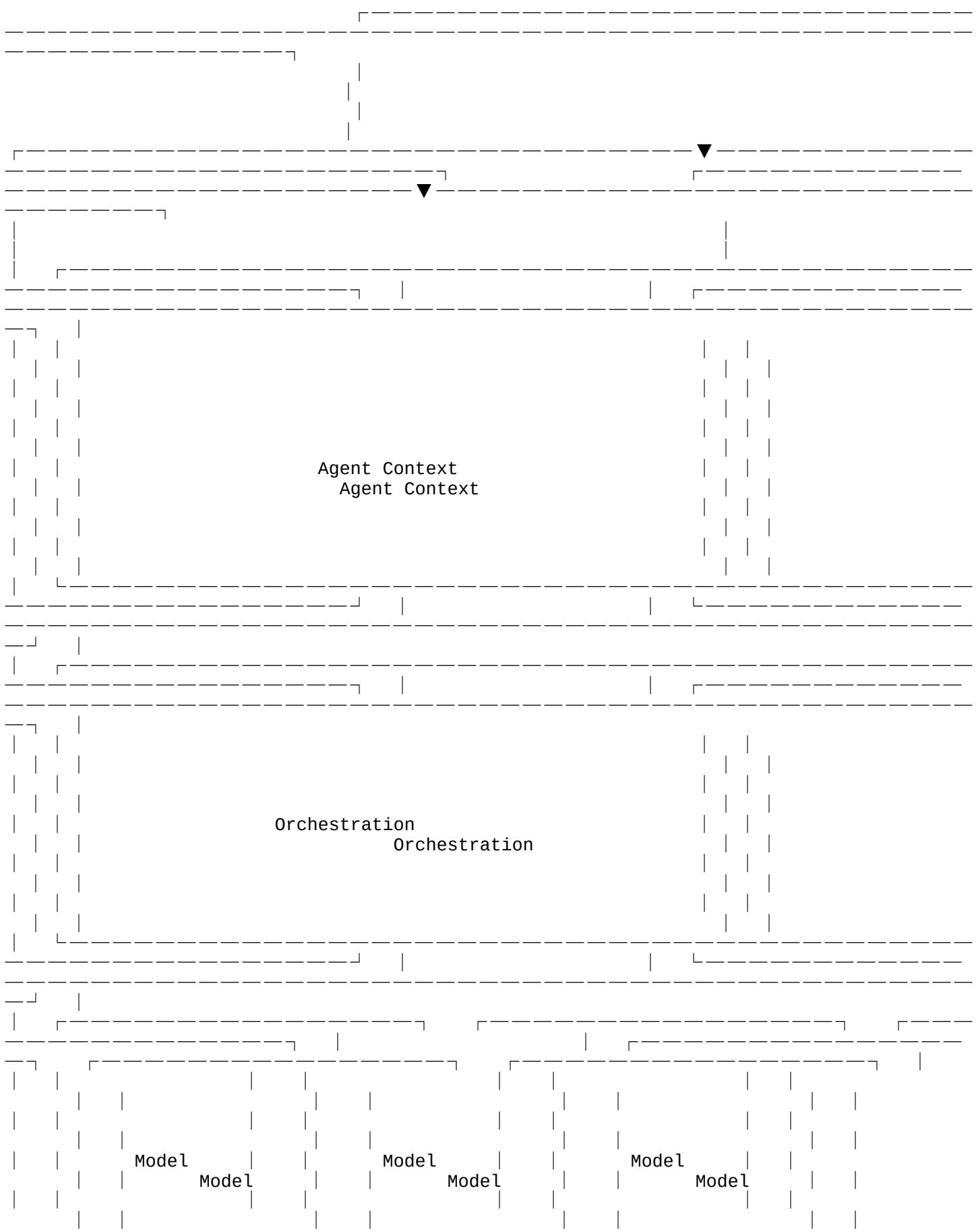
This specification defines the semantic standard and its protocol for AI agents to transmit context information, independent of the agent-to-agent communication protocol. Any communication protocol between agents can utilize this specification to implement bindings based on its semantics.

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. Architecture of Agent Context Protocol

As shown in figure 1, Agent Context Protocol is the "semantic layer" of the communication protocol stack between AI agents.


Agent Context
Agent Context
Orchestration
Orchestration
Model
Model
Model
Model
Model
Model

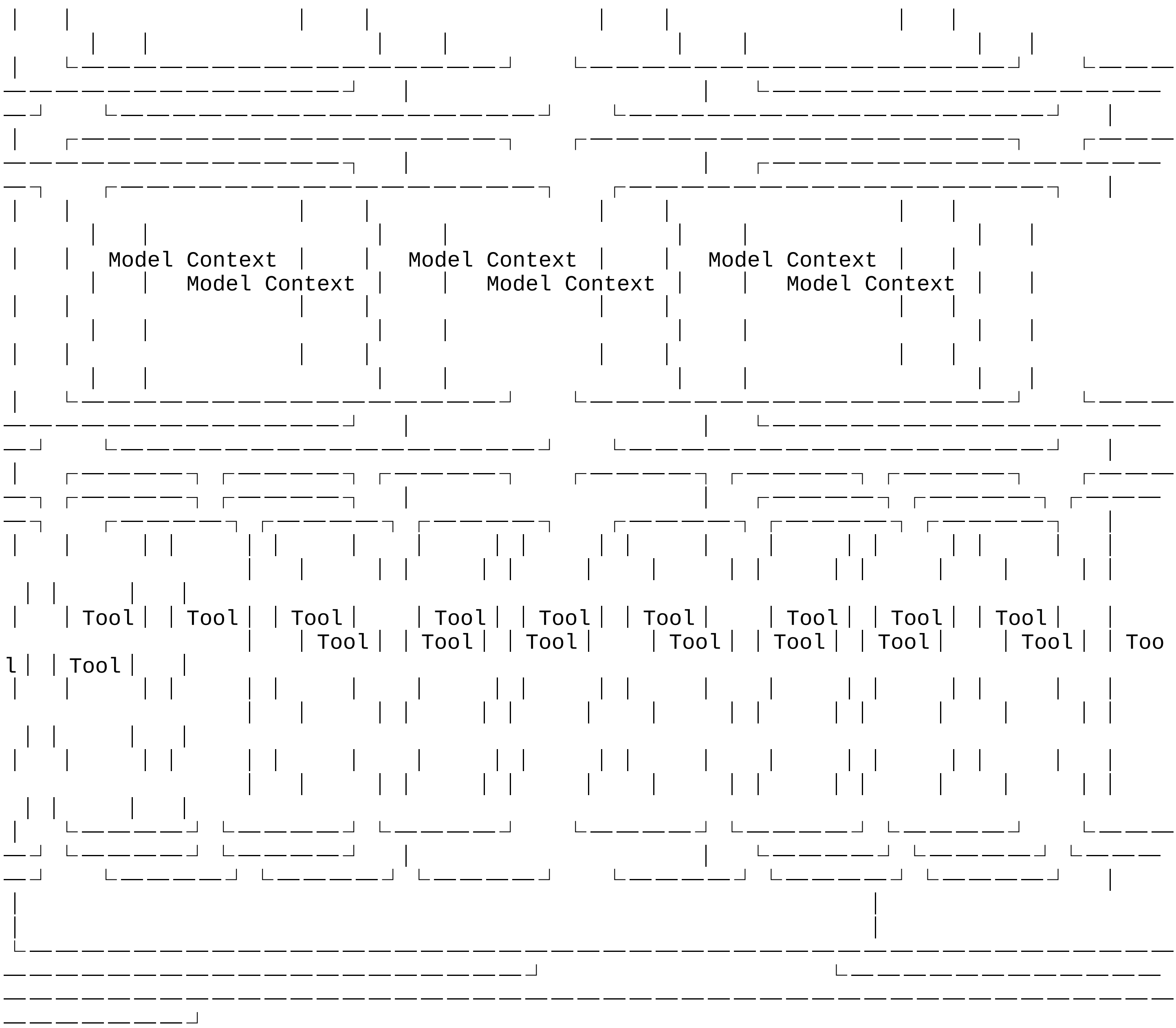


Figure 1: Diagram [REPLACE]

3. Agent Context Message

The context of an AI Agent refers to the context of information that the Agent can access and utilize when performing tasks or interacting with users.

Providing agent context information is critical for the invoked AI Agent to generate more accurate results and complete tasks that meet user requirements, security policies, and configurations. Well-defined agent contextual information enables the AI agent to better understand and meet user needs in a variety of complex situations, thereby improving its overall performance and service quality.

This agent context information is defined as 'Agent Context Message' in this specification. Agent Context Message is passed between orchestration layers where the network traffic originates and is terminated.

The orchestration layer should create the content of Agent Context Message and passed as a side channel along with agent workloads. Each orchestration layer can choose if it is important to include the Agent Context Message into the model invocation prompt, depending on the application. For example, a party might want the orchestrator to convert the JSON format of the Agent Context Message into something that performs better with models context windows (e.g. Markdown). As another example, the Orchestrator might take the list of rule from the Agent Context Message and put them as guidance to the agent when making ethical determinations.

This specification uses JSON to describe the Agent Context Message. A variety of agent communication protocols can implement binding using the semantics of the Agent Context Message as specified in this document.

The Agent Context Message is defined as follows:

* Task Context: AI agents collaborate with other agents to complete a specific task and need to share information about the task, such as its progress, goals, and to-do items. On the one hand, this information is provided to the collaborating agent. It can also be fed back to the user, so that the user can know the overall progress in a timely manner.

* MCP Context: The MCP Context information of the Agent includes the MCP service list, MCP Function list, and function descriptions. This information can be exchanged when a connection is initiated between agents and can be stored in the context of the Agent. When the Agent's MCP service is updated, this information can be

pushed to the Agents that have established connections with it, updating the context information. When a user inputs a task request to an Agent or one Agent calls another, the MCP Context information can also be provided simultaneously to the other Agent in order to select the appropriate MCP Function.

* Policy Context: The organizations that deploy the Agent can configure policies, such as certain rules that the Agent should follow, and provide these policies to local or peer Agents when the Agent is started.

* Resource Context: Resource context information pertains to the resources required for executing tasks and tool functions, as well as the resources that the system can provide.

* Security Context: The interaction between agents needs to consider security aspects such as access control, authentication, and data protection. The security-related context can provide information, for example, the authorization token that the initiator agent has obtained to simplify the user authorization experience.


```
{
 "agentContext": {
   "taskContext": {
     "taskId": "T123456",
     "taskName": "Data Analysis Project",
     "taskProgress": "30%",
     "goals": ["Analyze Q1 sales data", "Generate predictive models"],
     "todoItems": [
       {"itemId": "I001", "description": "Collect raw sales data from Q1"},
       {"itemId": "I002", "description": "Clean and preprocess the data"}
     ],
     "statusUpdates": [
       {"updateId": "U001", "timestamp": "2023-10-01T12:00:00Z", "message": "
Data collection completed."}
     ]
   },
   "mcpContext": {
     "mcpServiceList": [
       {"serviceName": "DataFetcher", "version": "1.0.0"},
       {"serviceName": "DataProcessor", "version": "2.0.1"}
     ],
     "mcpFunctionList": [
       {"functionName": "fetchData", "description": "Fetches raw data from a
specified source."},
       {"functionName": "processData", "description": "Cleans and processes t
he fetched data for analysis."}
     ]
   },
   "policyContext": {
     "policies": [
```

```
                    {"policyId": "P001", "rule": "Do not share sensitive data with externa
l parties without explicit user consent."},
                    {"policyId": "P002", "rule": "Ensure all communications are encrypted
using TLS 1.2 or higher."}
                  ]
                },
                "resourceContext": {
                  "requiredResources": [
                    {"resourceType": "CPU", "quantity": "8 cores"},
                    {"resourceType": "Memory", "quantity": "16 GB RAM"}
                  ],
                  "availableResources": [
                    {"resourceType": "Storage", "quantity": "1 TB SSD"},
                    {"resourceType": "Network", "bandwidth": "1 Gbps"}
                  ]
                },
                "securityContext": {
                  "authenticationInfo": {
                    "token": "eyJhbGciOiJIUzy***"
                  },
                  "accessControl": {
                    "role": "DataAnalyst",
                    "permissions": ["read:data", "write:reports"]
                  },
                  "dataProtection": {
                    "encryptionEnabled": true,
                    "encryptionAlgorithm": "AES-256"
                  }
                }
              }
            }
```


Figure 2: Agent Context Message

4. Agent Context Protocol

The Agent Context Protocol is used to exchange agent context message defined in section 3 between agents. The Agent Context Protocol is based on the JSON format and uses HTTP as the underlying transport protocol. The Agent Context Protocol support bidirectional communication between agents(TBD).

5. IANA Considerations

TBD

6. Security Considerations

The Agent Context Message should be protected by encryption and authentication (TBD). Since the intent of this Agent Context Message is to provide soft guidance and the Agent Context Message should never be load bearing on security.

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>.

Acknowledgements

The author would like to thank Jason Clinton, CISO of Anthropic for his constructive suggestions and review.

Contributors

TBD

Author’s Address

Dapeng Liu (editor)
Alibaba
Beijing
Beijing
100035
China
Email: max.ldp@alibaba-inc.com