DMSC Working Group J. Liu
Internet-Draft K. Yu
Intended status: Informational K. Li
Expires: 2 August 2026 K. Chen
Beijing University of Posts and Telecommunications
29 January 2026

Agent Collaboration Protocols Architecture for Internet of Agents
draft-liu-dmsc-acps-arc-03

Abstract

Internet of Agents (IoA) aims to facilitate interconnection and collaboration among heterogeneous agents to address complex tasks and support diverse applications.

This IETF draft proposes the Agent Collaboration Protocols (ACPs) architecture, which includes conceptual domains, functional components and reference interfaces, to achieve agent interconnection and collaboration. ACPs cover all stages of agents in the network, from their access to collaboration, supporting agent trusted registration, agent identity authentication, agent discovery, agent interaction, tool invocation, and agent monitoring. The long-term vision of ACPs is to support the future large-scale interconnected agents and construct the key infrastructure for IoA.

Table of Contents

1. Introduction

With the rapid development of artificial intelligence (AI), particularly large language model (LLM) technology, the number of AI agents has grown dramatically. With the capability of autonomous perception, decision-making, and execution, agents' applications are becoming increasingly widespread.

To overcome the limitations of single-agent systems, and to break free from the constraints of proprietary multi-agent frameworks developed by various vendors, the Internet of Agents (IoA) has emerged. IoA aims to enable seamless connectivity and efficient collaboration among agents, through standardized communication protocols and interfaces.

This draft proposes the functional architecture of Agent Collaboration Protocols (ACPs) [ACPs-Github], which is designed for IoA to enable wide-area connectivity, cross-domain interoperability, and secure collaboration among heterogeneous agents. The main characteristics of ACPs are as follows:

* Multi-centralized architecture, which consists of multiple autonomous domains, each domain containing its own management nodes (such as registration, authentication, and discovery functions), to support efficient, reliable, and manageable large-scale agent interconnection scenarios.

* Standardized communication mechanisms with peer-to-peer and grouping mode, allowing agents to self-organize and negotiate autonomously, to facilitate rapid and accurate information exchange, as well as efficient task Collaboration.

* Robust registration and authentication mechanisms, to ensure the trusted access of agents and prevent unauthorized access and data breaches.

* Reliable registration and management of agent capability, along with intra-domain and cross-domain discovery based on capability matching, to support universal and efficient discovery of collaborative agents.

* Real-time monitoring of agent status and behavior, to support more complex application requirements such as agent auditing and transactions.

ACPs cover all stages of agents in the IoA, from their access to collaboration, to construct the key infrastructure for agent communication, task collaboration and resource allocation.

2. Conventions

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

Agent: An agent is a software or hardware entity with autonomous decision-making and execution capabilities, capable of perceiving the environment, acquiring contextual information, reasoning, and learning. Agent can perform tasks independently or collaboratively with other agents. Each agent is created by a specific agent provider and needs to complete processes such as registration and authentication before providing services to obtain a legitimate identity and relevant credentials.

Agent Identity Code (AIC): AIC is a certifiable, globally unique identity that represents the identity of an agent. AIC MAY contain the following information: the registration service center, the agent provider, the serial number of agent entity and instance, and the check code.

Agent Capability Specification (ACS): ACS is a detailed description of an agent’s capabilities and information that can be saved and retrieved. ACS MAY use the JSON [RFC8259] format, typically including the following information: AIC, the functional capabilities of agent, the technical characteristics, and the service interfaces.

Agent Credential: Agent Credential is a tamper-resistant data object issued by a credential authority (e.g., certificate, token) , used by an agent to prove identity attributes and/or authorization to a relying party.

Tool: A tool is a device, software component, or service that provides a specific function and can be accessed and used by an agent.

Agent Autonomous Domain: Agent Autonomous Domain is an administrative and governance domain organized and managed by a specific IoA service provider. It MAY include service functions such as identity management, credential management, capability description management, agent discovery, message distribution, and agent monitoring.

4. ACPs Architecture from Functional Perspective

This section defines a functional architecture for agent autonomous domain in IoA. The interconnection environment is divided into different conceptual domains. Each conceptual domain includes several functional components, and there are different interfaces between functional components.

4.1. ACPs Architecture Overview

The overall functional architecture of ACPs is as follows.

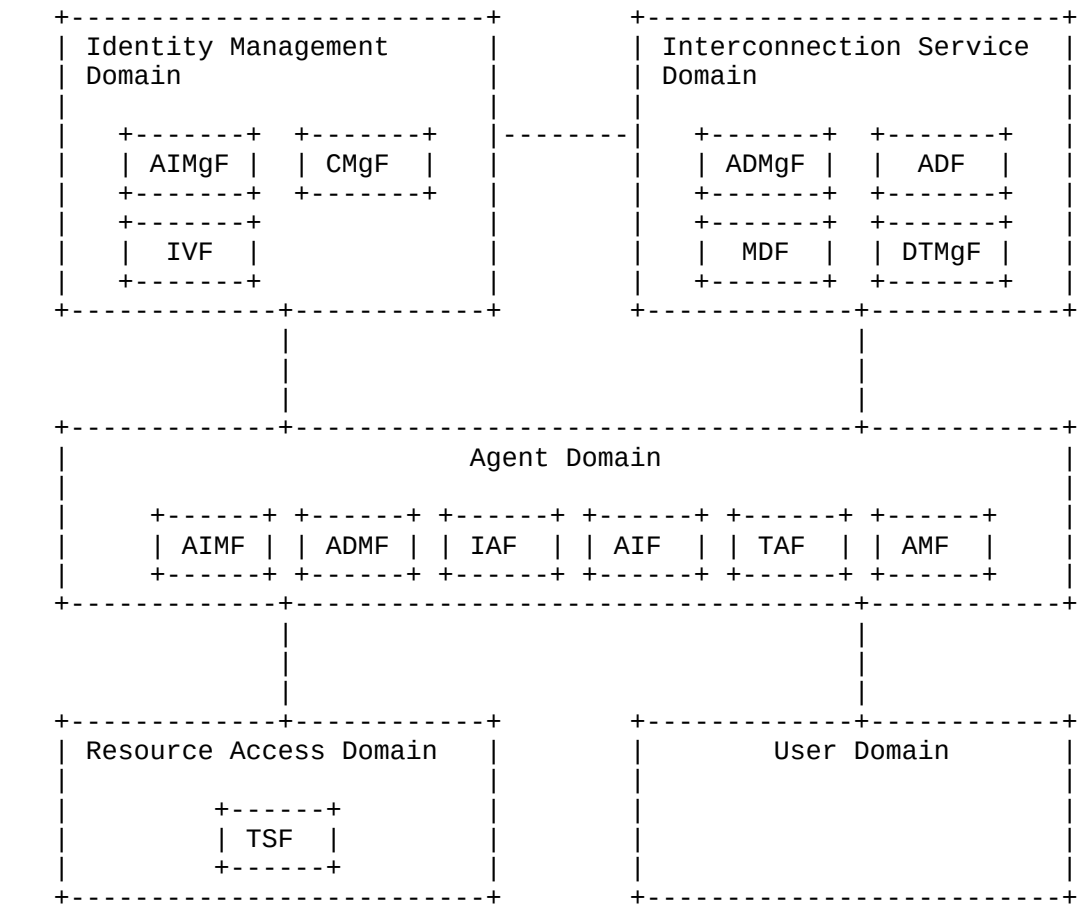

Figure 1: ACPs functional architecture

The agent interconnection environment is organized into five conceptual domains:

1. User Domain

2. Agent Domain

3. Identity Management Domain

4. Interconnection Service Domain

5. Resource Access Domain

The lines between conceptual domains represent information exchange relationships.

Domains are not necessarily physical network segments; they represent responsibility and governance boundaries. A deployment MAY collapse multiple domains into one system, or MAY distribute them across multiple organizations.

4.2. Conceptual Domains

4.2.1. User Domain

The User Domain is the set of users and user environments that initiate tasks and consume final results. A user MAY be a person or an organization.

4.2.2. Agent Domain

The Agent Domain contains agents and agent-side functions required for agent interconnection and collaboration, such as agent description, agent interaction, agent identity maintenance, interconnection authentication, and tool access.

The agent is the core execution unit of agent interconnection and collaboration. An agent is typically provided and operated by an agent provider. Agents in this domain MUST be able to present identity and comply with authentication / authorization decisions when interacting with other domains.

4.2.3. Identity Management Domain

The Identity Management Domain contains functions for identity lifecycle management, credential lifecycle management, and identity verification. It provides services for registering agents, issuing credentials, and verifying identity claims during interconnection.

4.2.4. Interconnection Service Domain

The Interconnection Service Domain contains functions that enable agents to find and collaborate with each other, including agent capability description management, agent discovery, and message distribution for group interactions.

4.2.5. Resource Access Domain

The Resource Access Domain contains tool and resource service functions that agents can invoke, such as tool registries, tool execution services, data services, and other external resources.

4.3. Functional Components

The functional components are grouped by conceptual domains. The list below is a quick index. The detailed descriptions are provided in Table 1.

User Domain
- User

Agent Domain
- Agent
- Agent Identity Maintenance Function (AIMF)
- Agent Description Maintenance Function (ADMF)
- Interconnection Authentication Function (IAF)
- Agent Interaction Function (AIF)
- Tool Access Function (TAF)
- Agent Monitoring Function (AMF)

Identity Management Domain
- Agent Identity Management Function (AIMgF)
- Credential Management Function (CMgF)
- Identity Verification Function (IVF)

Interconnection Service Domain
- Agent Description Management Function (ADMgF)
- Agent Discovery Function (ADF)
- Message Distribution Function (MDF)
- Domain Trust Management Function (DTMgF)

Resource Access Domain
- Tool Service Function (TSF)

Figure 2: Functional component index by conceptual domain (informative)

Conceptual Domain	Functional Components	Function Description
User Domain	User	Initiate tasks; provide authorization and policy input; consume results.
Agent Domain	Agent, AIMF, ADMF, IAF, AIF, TAF, AMF	Execute tasks; maintain local identity and capability description; interact with other agents; access tools; record logs.
Identity Management Domain	AIMgF, CMgF, IVF	Manage identity and credential lifecycle; verify identity claims and credential status.
Interconnection Service Domain	ADMgF, ADF, MDF, DTMgF	Manage/publish capability descriptions; enable discovery; support group messaging and domain federation policies.
Resource Access Domain	TSF	Expose and control access to tools/resources; obtain results.

Table 1: Functional components summary

The functional boundaries defined here are intended to support interoperability and clear interface definition. A deployment MAY realize a functional component as software, hardware, or a managed service, and MAY consolidate multiple functional components into one product.

4.3.1. Functional Components in the Agent Domain

1. Agent Identity Maintenance Function (AIMF)

AIMF maintains the agent’s identity information (e.g., AIC) and credentials. AIMF SHOULD support secure storage and update of identity.

2. Agent Description Maintenance Function (ADMF)

ADMF maintains agent capability description information (e.g., ACS) used for interconnection. It supports creating, updating, and withdrawing descriptions. ADMF SHOULD ensure that descriptions remain consistent with the agent's current identity and capability state.

3. Interconnection Authentication Function (IAF)

IAF performs authentication and authorization checks for interconnection, including mutual verification of peer agents and validation of presented credentials. IAF SHOULD support multiple authentication mechanisms. Deployments are RECOMMENDED to use standardized security protocols such as TLS 1.3 [RFC8446] and mutual authentication where appropriate.

4. Agent Interaction Function (AIF)

AIF provides standardized interaction semantics for agent-to-agent communication. It is responsible for session establishment, message exchange, task/context management, and applying security requirements. AIF SHOULD support at least one of point-to-point and grouping interaction patterns between agents.

5. Tool Access Function (TAF)

TAF provides the agent-side capability to invoke tools and resources in the Resource Access Domain. It supports discovering available tools, invoking a selected tool, handling tool results, and enforcing access control decisions.

6. Agent Monitoring Function (AMF)

AMF exports monitoring and logging data produced by the agent and agent-side functions. It SHOULD support integrity protection and privacy controls.

4.3.2. Functional Components in the Identity Management Domain

1. Agent Identity Management Function (AIMgF)

AIMgF performs identity lifecycle management for agents, including allocation, update, and de-registration of AICs. AIMgF MUST define the policies for uniqueness and governance of issued AICs within its scope.

2. Credential Management Function (CMgF)

CMgF issues and manages Agent Credentials associated with AICs. CMgF SHOULD support credential issuance, renewal, suspension, revocation, and status query.

3. Identity Verification Function (IVF)

IVF verifies identity claims and credential validity for relying parties. IVF MAY be offered by the same operator as CMgF or by an independent third party. IVF SHOULD support validation of credential authenticity, validity period, and status (e.g., revoked/suspended).

4.3.3. Functional Components in the Interconnection Service Domain

1. Agent Description Management Function (ADMgF)

ADMgF manages the lifecycle of agent capability descriptions in the interconnection ecosystem. It supports accepting capability description submissions, validating required fields, reviewing and approving publication policies, publishing descriptions, and de-listing descriptions.

2. Agent Discovery Function (ADF)

ADF enables capability-based discovery of candidate partner agents. It receives discovery queries, matches them against available agent descriptions, and returns a candidate set. ADF SHOULD support in-domain discovery; and cross-domain discovery is OPTIONAL, within a configured trust scope.

3. Message Distribution Function (MDF)

MDF provides message distribution services for group interactions, such as publish/subscribe or queue-based delivery. MDF is typically used when agents interact in grouping mode.

4. Domain Trust Management Function (DTMgF)

DTMgF maintains trust relationships between autonomous domains (e.g., trusted peer domain lists, federation policy). If present, it constrains cross-domain discovery and service-to-service interactions.

4.3.4. Functional Components in the Resource Access Domain

1. Tool Service Function (TSF)

TSF provides access to tools and resources for agents. TSF SHOULD support tool registration or exposure, tool invocation handling, result return, and access control enforcement. TSF MAY represent a single tool, a tool gateway, or a tool execution environment.

4.4. Reference Interfaces

The reference interfaces between functional components are identified as ACP-IF-XX. An ACP-IF interface definition specifies what information is exchanged and what behavior is expected. It does not mandate a specific protocol.

4.4.1. Core Interfaces

Interface ID	Function A	Function B	Interface description
ACP-IF-01	AIMF	AIMgF	Apply for / update / de-register an AIC; synchronize identity lifecycle state.
ACP-IF-02	AIMF	CMgF	Apply for / renew / revoke Agent Credentials bound to an AIC.
ACP-IF-03	IAF	IVF	Verify a peer agent’s claimed identity and presented credentials; obtain a verification result / assertion.
ACP-IF-04	AIMgF	CMgF	Coordinate identity-to-credential binding; synchronize identity and credential status.
ACP-IF-05	CMgF	IVF	Query credential status and validation material (e.g., issuer keys, revocation state).
ACP-IF-06	ADMF	ADMgF	Submit / update / withdraw Agent Descriptions; receive publication status.

ACP-IF-07	ADMgF	ADF	Synchronize published Agent Descriptions and metadata for discovery indexing.
ACP-IF-08	AIF	AIF	Agent-to-agent interaction (session, task, message exchange) under an agreed interaction mode.
ACP-IF-09	AIF	MDF	Group interaction via message distribution (create / join group, publish / subscribe, deliver messages).
ACP-IF-10	TAF	TSF	Tool / resource invocation (discover tools, invoke tools, obtain results, manage context).

Table 2: Reference interfaces summary

4.4.2. Interface Semantics

* *Identity and credential interfaces (ACP-IF-01..05)* provide the basis for trust establishment. Implementations SHOULD support auditability of lifecycle operations.

* *Description and discovery interfaces (ACP-IF-06..07)* enable capability-based agent discovery. Implementations SHOULD support versioning and de-listing of descriptions.

* *Agent interaction interfaces (ACP-IF-08..09)* define how agents exchange tasks, messages, and context. Implementations SHOULD support confidentiality, integrity, replay protection, and access control.

* *Tool access interface (ACP-IF-10)* enables secure invocation of external tools or resources. Implementations SHOULD support least-privilege access and result integrity.

5. Typical Operational Scenarios

This section illustrates how the functional architecture with conceptual domains, functional components, and interfaces is applied in typical operational scenarios.

5.1. Agent Onboarding: Identity, Credential, and Capability Description

A typical agent onboarding workflow includes:

1. The agent (via AIMF) registers identity with AIMgF (ACP-IF-01) and obtains an AIC.

2. The agent (via AIMF) obtains credentials from CMgF (ACP-IF-02). CMgF and AIMgF coordinate binding and status (ACP-IF-04).

3. The agent (via ADMF) submits an ACS to ADMgF (ACP-IF-06). The ACS links to the agent’s AIC and includes access and capability information.

4. ADMgF publishes the ACS and synchronizes to ADF (ACP-IF-07) for discovery indexing.

5.2. Agent Discovery and Partner Selection

In ACPs, the interaction between agents is task-driven, based on which agents can be divided into two roles:

* Leader: Agent who issues tasks and organizes interactions. There should only be one Leader in a complete task execution process.

* Partner: Agent who accepts tasks and provides services. After Partner receives a task from the Leader, it executes and returns the execution result.

A typical agent discovery workflow includes:

1. A Leader determines required capabilities for a task.

2. The Leader queries ADF with constraints derived from the task (discovery request semantics are realized by ADP).

3. ADF returns a ranked candidate set of Partner agents.

4. The Leader (via IAF) authenticates selected Partners (ACP-IF-03) before starting interaction.

5.3. Agent Interaction with Point-to-point and Grouping Mode

In AIP, there are three interaction modes between the Leader and the Partner: peer-to-peer mode, grouping mode and hybrid mode.

* peer-to-peer mode: In this mode, the Leader maintains separate communication connections with each Partner, ensuring context isolation. Each message in this mode has only one sender and receiver.

* Grouping mode: In this mode, interaction messages between agents are distributed through a message queue. A message in this mode has only one sender but may has multiple receivers.

* Hybrid mode: In this mode, the Leader and Partner may interact directly or through the message queue according to the Leader’s task planning.

In point-to-point mode, the Leader and each Partner communicate directly via ACP-IF-08.

In grouping mode, a message distribution service (MDF) is used. Agents create/join groups and exchange messages via ACP-IF-09, while still using ACP-IF-03 for authentication / authorization decisions.

5.4. Tool Invocation

When an agent needs external tools / resources, a typical tool invocation workflow includes:

1. The agent uses TAF to discover available tools (locally cached or via TSF).

2. The agent invokes a tool (ACP-IF-10).

3. TSF executes or brokers the tool and returns results to the agent.

5.5. Agent Monitoring

If agent monitoring is required, agents and services export logs / events via AMF. Monitoring data can be used for troubleshooting, policy enforcement, auditing, and governance.

6. Conclusions

This draft introduces the Agent Collaboration Protocols (ACPs) architecture for Internet of Agents. By defining a functional architecture with conceptual domains, functional components and reference interfaces, ACPs can support trusted registration, identity authentication, agent discovery, agent interaction, tool invocation, and agent monitoring. ACPs address the core requirements for interoperability and secure collaboration among heterogeneous agents. The long-term goal of the ACPs is to enable IoA to serve as the critical infrastructure for agent collaboration on complex tasks, as well as to support the future large-scale and diverse applications.

7. Security Considerations

This document focuses on the Agent Collaboration Protocols architecture for IoA. Security of IoA is not detailed in this document. Security considerations relevant to deployment with multiple agent service providers are suggested to be deeply discussed through other proposals.

8. IANA Considerations

This document makes no request for IANA action.

9. References

9.1. 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>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

9.2. Informative References

[ACPs-Github] "ACPs GitHub repository", n.d., <https://github.com/AIP-PUB>.

Authors' Addresses

Jun Liu
Beijing University of Posts and Telecommunications
10 Xitucheng Road, Haidian District
Beijing
100876
China
Email: liujun@bupt.edu.cn

Ke Yu
Beijing University of Posts and Telecommunications
10 Xitucheng Road, Haidian District
Beijing
100876
China
Email: yuke@bupt.edu.cn

Ke Li
Beijing University of Posts and Telecommunications
10 Xitucheng Road, Haidian District
Beijing
100876
China
Email: like1990@bupt.edu.cn

Keliang Chen
Beijing University of Posts and Telecommunications
10 Xitucheng Road, Haidian District
Beijing
100876
China
Email: chenkl@bupt.edu.cn