Network Working Group
Internet-Draft
Intended status: Informational
Expires: 23 April 2026

K. Yao
China Mobile

20 October 2025

Further considerations on AI Agent Authentication and Authorization Based on OAuth Extension draft-yao-agent-auth-considerations-01

Abstract

Agent Communication Network(ACN) is becoming a promising and fundamental infrastructure for most vertical industries. To construct and build a scalable and trustable ACN, authentication and authorization of AI agents are critical requirements. This document extends the model of OAuth and proposes new workflows for AI agent authentication and authorization.

Table of Contents

1. Introduction

With the rapid development of large language models(LLMs) and AI applications, an AI agent is becoming an emerging entity that can help improve working efficiency. There are different types of AI agents, e.g., physical and virtual entities, and many promising use cases of AI agents have been mentioned in [I-D.rosenberg-aiproto-framework]. All of these AI agents will join

to build a new Internet infrastructure, which is called as Agent Communication Network(ACN). To build such future networks is not easy, challenges and requirements for new protocols are discussed in [I-D.rosenberg-aiproto-framework]. Key requirements include, the discovery, procedures and mechasims of AI agents to establish message routing and connection management. A fundamental component and of critical importance for any new AI agent protocol is ensuring robust authentication and authorization.

The complexity of AI agent authentication and authorization exists in that agent may have different roles and capabilities. AI agents may work On-behalf-of(OBO) users, itself, or other AI agents. In different cases, there are different requirements on the authentication and authorization, leading to different workflows. More importantly, agents may communicate with API proxy server, like MCP server to call API and access external resources, this makes the authentication and authorization problems more complex. [I-D.oauth-ai-agents-on-behalf-of-user] considers the case when AI agents work OBO their users, and [I-D.rosenberg-oauth-aauth] defines the extension of OAuth 2.1 for AI agents. This document further considers more cases on AI agents authentication and authorization based on OAuth extensions.

This document describes an extension to the OAuth to support authentication and authorization among AI agents within ACNs. It defines three operational modes in which agents act on behalf of users, themselves, or other agents, introduces the concept of an AID, and discusses integration with API proxy servers such as the MCP.

2. Definition of Terms

* AI Agent: an entity with built-in intelligence, which can help or replace humans implement jobs and improve work efficiency.

* Types of AI Agents:

  - Physical AI Agent: a physical entity with embedded intelligence, usually refers to some AI terminals, e.g., AI robot, embodied AI.

  - Virtual AI Agent: a virtual entity that can provide intelligence, usually refers to some softwarized AI assistant, e.g., a chat assistant with a mobile application.

* Ownership and Roles of AI Agents:

  - OBO User: the agent may require the authorization from users to implement jobs for users.

- OBO Agent Itself: the agent itself has identification and implements jobs and represents itself.

- OBO Other Agent(s): the agent may represent other agents to implement some jobs, given that other agents may not have the capabilities to implement the jobs.

* Agent Communication Network(ACN): a network infrastructure which supports the interconnection, routing, capabilities announcement, and task collaboration of different types of AI agents.

* Agent Identifier(AID): An independent identifier of AI agent.

* API Proxy Server: A middlebox server that helps AI agent to call APIs or access external data resources. A typical example is the Model Context Protocol(MCP) server.

* Domain:

  - Task-wise Domain: a temporary domain that is created to target on a specific job, when the job is finished, the domain is destroyed. A typical example is a domain that is created by 5G/6G core network.

  - Application Domain: a durable domain that is created by a specific application, e.g., a web or mobile application offered by cloud service providers.

  The definition of resource server, resource owner, authorization server, and client reuse the definition in [RFC6749].

3. Aauth: Agent OBO Its User(s)

3.1. Example Case

A typical use case of AI agent on-behalf-of the user to access public resources can be a virtual agent assistant. For example, a trip planning app assistant helps the user to plan trips. This case is obvious and similar ones have been mentioned in [I-D.rosenberg-aiproto-framework]. In this case, when the trip assistant needs to search for the current balance of the user, it needs the authorization of the user to access his bank account app. While it may also need to search for hotels, and it may require the account of the user under some booking applications. In addition, it may also need to use some of the data stored in the cloud personal gallery, which may require temporary access token. Therefore, AI agent OBO user(s) need different levels of authentication and authorization to access these resources.

3.2. Model

Figure 1 shows the extension of the OAuth 2.0 model proposed in [RFC6749]. There are several updates and new components in this model. The big difference is the introduction of AI agent and API proxy server. If under traditional OAuth 2.0 model, client(AI agent) will directly communicate with resource owners and resource servers. But this is not efficient and scalable in the AI era, since an agent may consult many resource owners at the same time, and different resource owners may require different levels of privileges to access protected resources. To overcome the issue, industries have proposed Model Context Protocol(MCP). It introduces MCP server and protocol to let AI agents only communicate with the MCP server to request for resources. The MCP server will help AI agents to do API calling and specific data resource access. In Figure 1, MCP server refers to the API proxy server. In the diagram, API proxy server will communicate with different resource owners and resource servers. The agent may also support communicating directly with the resource owner and resource server. For example, the resource owner is another agent that is interconnected with the agent itself.

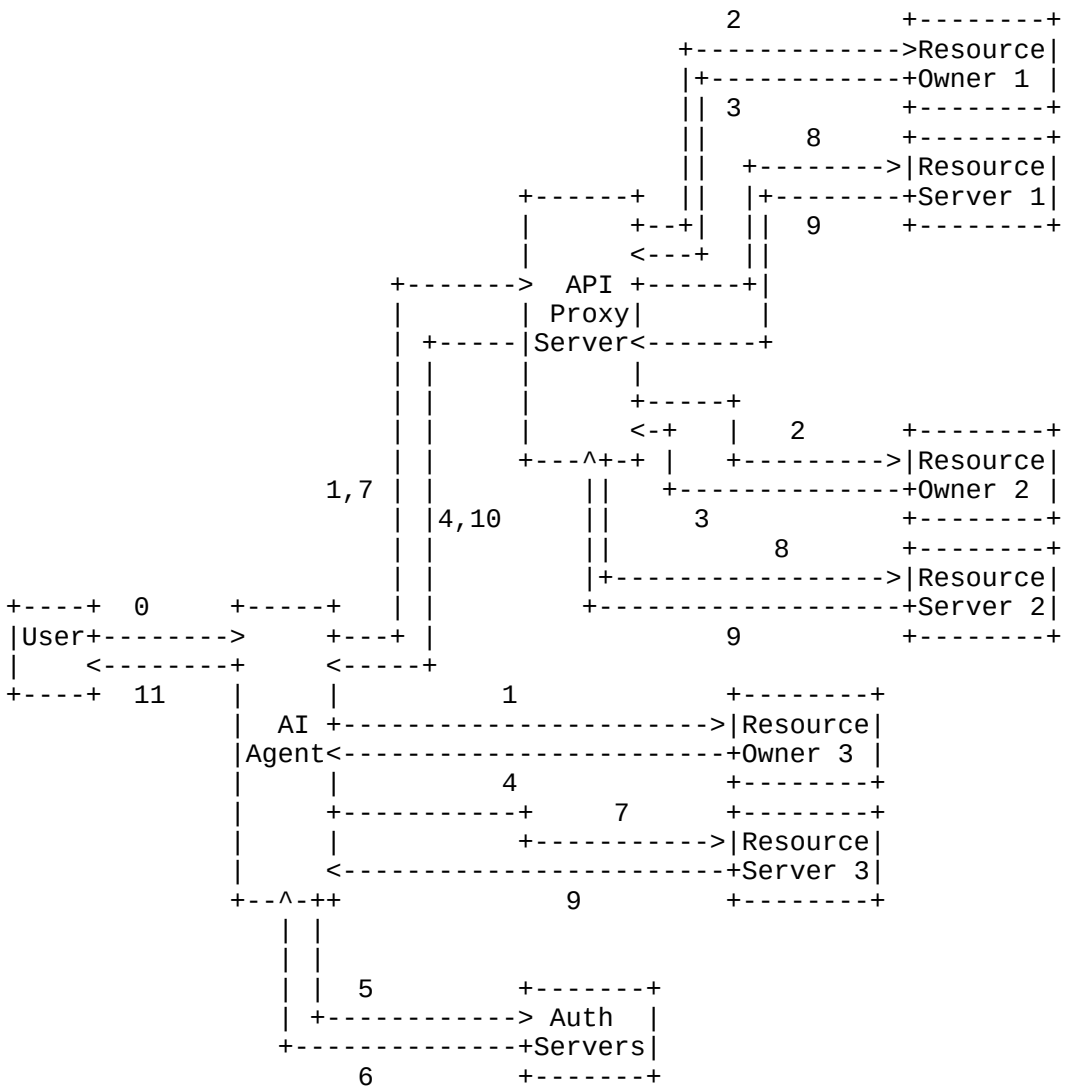

Figure 1: AI Agent OBO User Workflow

3.3. Workflow

The detailed workflow of Figure 1 is as follows:

* Step 0, the user initially asks for AI agent for some tasks via prompts.

* Step 1, the AI agent interpret the prompts and communicate with the API proxy server to search for specific data resources and implement function calling. Alternatively, the agent will ask the resource server for the grant if they are directly connected.

* Step 2, the API proxy server asks for multiple resource owners for the grant of access of the protected data resources.

* Step 3, resource owners reply to the API server what authentication grant(e.g., authorization code) they need for the access to the required resource.

* Step 4, the API proxy server will gather messages from different resource owners and reply to the AI agent containing multiple authorization grants from different resource owners. Alternatively, resource owner 3 will replies to the agent with the authorization grant directly.

* Step 5, the AI agent communicates with different authorization servers with multiple authorization grants.

* Step 6, auth servers reply to the AI agent with the required access tokens.

* Step 7, AI agent replies to the API server with the required access tokens. Alternatively, it directly sends the access token to the resource server(resource server 3).

* Step 8, the API proxy server sends different access tokens to different resource servers respectively.

* Step 9, resource servers send protected data resources to the API proxy server. The resource server 3 directly sends the protected data resources to the AI agent.

* Step 10, the API proxy server replies to the AI agent with the information that it needs.

* Step 11, AI agent replies to the user with the answer.

4. Aauth: Agent OBO Itself

4.1. Example Case

In addition to virtual AI agents, there is another type, physical AI agent. AI robots, embodied AI, and other AI terminals have differentiated capabilities to implement multiple jobs. For example, in a remote rescue case, an AI search and rescue vehicles, and an robot dog can be dynamically formed into a networked system. The robot dog can scan and transmit the live videos to remote console and control signals to the AI search and rescue vehicle, while the AI search and rescue vehicles can use its robotic arm to move obstacles based on the signals from the robot dog or the remote console. Some other similar cases are mentioned in [TR22.870].
Compared to the case in which the agent is OBO its user, the major difference in this case is that agent need identification of itself, that is the agent identification(AID). In previous case, agent can be assigned a Client ID(CID). The scope of its authority is determined by its user, and is less than its users' authority scope. While if agent is OBO itself, it is the user. So it has the same privilege as the user. How to define the AID needs further discussion and is beyond the scope of this document.

4.2. Model

The model of the agent OBO itself can be extended from[I-D.ietf-oauth-v2-1]. As shown in Figure 2, the AI agent is the resource owner. When assigned with some specific tasks, it may enable some software functions, and then it will trigger the client within these software functions for authentication and authorization.

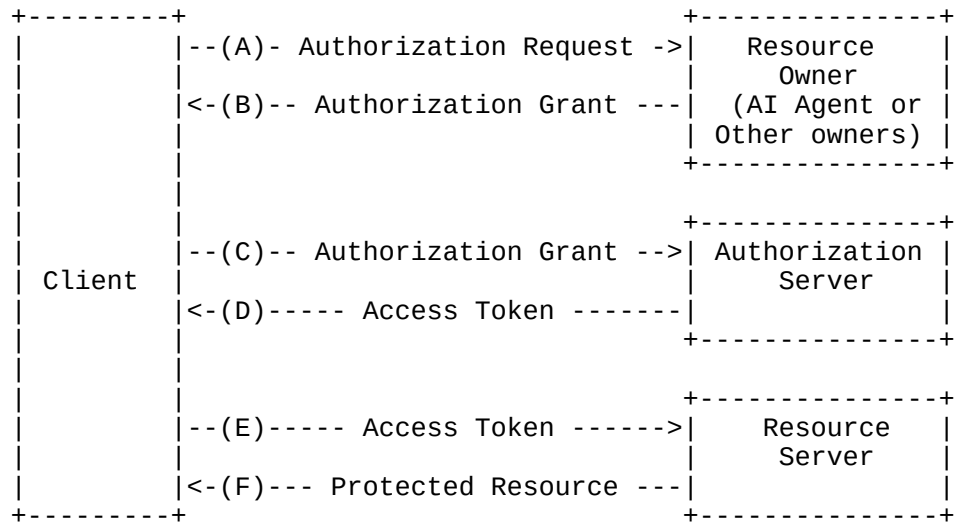

Figure 2: AI Agent OBO Itself Workflow

4.3. Workflow

Detailed workflow is similar to [I-D.ietf-oauth-v2-1], and will not be stated in detail.

5. Aauth: Agent OBO Other Agent(s)

5.1. Example Case

This case is more complex, but may be required if AI agents have differentiated capabilities and need to collaborate to finish some jobs, whatever the agents are from intra-domain or inter-domain, which has been discussed in [I-D.rosenberg-aiproto-framework]. In this case, if one AI agent wants to access some protected data resources of other agents, or of the users of other agents. It needs the authorization from them. This is different compared to the situation that this agent only needs to seek for grant from the user it represents.

5.2. Model

Figure 3 shows the model when multiple agents work collaboratively, and there will be a chained authentication and authorization workflow. Before AI Agent A requests AI agent B to help it implement a job, there should be prior knowledge that both agents need mutual trust. If they come from the same domain, whatever it is a task-wise domain created by a console(e.g., 5G/6G core) or a durable domain created by a web application, they need verification from the 5G/6G core or the web application administrator. If they come from different domains, they may need verification from an open platform that organize and issue the AIDs. There are some ongoing work defining AID[I-D.narajala-ans],[I-D.narvaneni-agent-uri]. With this pre-verification, agents can communicate with each other. But when one agent(AI agent B) needs access of the protected data resource of the user of AI agent A or AI agent C itself, it still need temporary authentication and authorization. In this chained authentication and authorization workflow, the core idea here is to let the agent OBO user or the agent OBO itself always to communicate with the authorization server to get the access token.

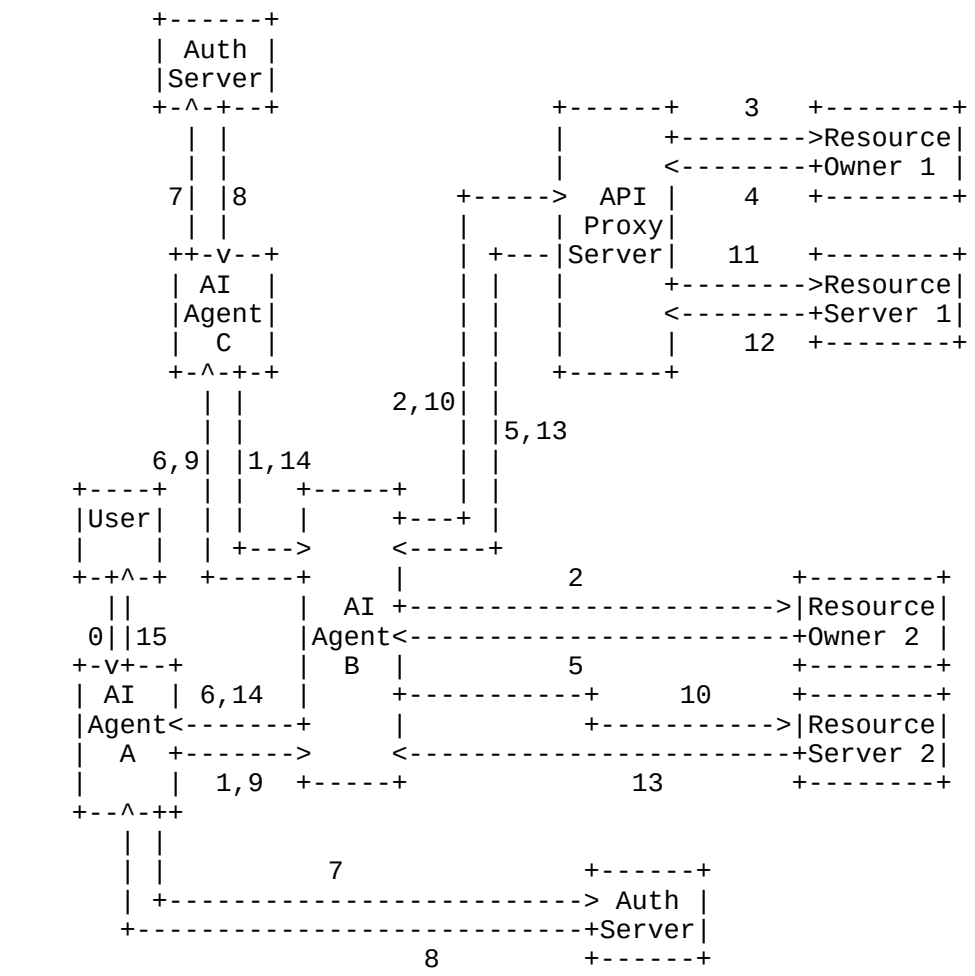

Figure 3: AI Agent OBO Other Agents Workflow

5.3. Workflow

The detailed workflow of Figure 3 is as follows:

Step 0, a user asks its AI agent(AI agent A) for some help.

Step 1, AI agent A redirects the help to Agent B when A is not capable, and Agent C asks for B’s help too.

Step 2, agent B ask API proxy server to call APIs. Agent may also request resource owner’s grant for its protected data resources.

Step 3, the API proxy server send requests to resource owner 1.

Step 4, the resource owner 1 send access grant to the API proxy server.

Step 5, the API proxy server and resource owner 2 send the access grant to AI agent B.

Step 6, AI agent B redirects the access grants to AI agent A or AI agent C, considering the resource belong to whom.

Step 7, AI agent A and AI agent C send access grants to authorization servers.

Step 8, authorization servers replies with authorization tokens.

Step 9, Agent A and agent C pass the access tokens to agent B.

Step 10, AI agent B passes the access tokens to API proxy server or resource owner 2, considering what resource agent B wants to access.

Step 11, the API proxy server passes the access token to resource owner 1.

Step 12, resource owner 1 gives the protected data resource to the API proxy server.

Step 13, agent B gathers the data resources from the API proxy server and resource owner 2.

Step 14, agent B sends back the processing result to agent A or agent C.

Step 15, agent A processes further and sends back the final result to the user.

6. Considerations on Other Important Factors

6.1. Grant Dynamicity

[RFC7591] mentions the mechanism to realize dynamic client registration. Whether and when to grant AI agents with short-lived or long-lived authentication and authorization may need further discussion, considering various use cases that AI agents participant.

6.2. Specific Identity for AI Agent

As the second and the third case mentions in the previous section, AI agents may have independent identities in ACN. The definition of AIDs is not within the scope of this document, but directly impacts authentication and authorization methods.

6.3. Can AI Agents Represent Users to Consent Requests?

In the third case mentioned in this document when agent is OBO other agents. This document assumes that agent can represent its user to request for access tokens, while the choice on whether to consent requests(authorization grant) is still left to the user or the agent(agent C). But it may evolve to the situation that AI agent can represent its user to give permission if there are some pre-validation on the content scope that AI agent can give external grants independently.

7. Security Considerations

AI Agent authentication and authorization introduce additional risks beyond those covered in [RFC6749] and [I-D.ietf-oauth-v2-1]. This section highlights specific threats and corresponding mitigations applicable to agent-to-agent (A2A) environments and Agent Communication Networks (ACN).

Attention must be given to agent discovery, agent sandboxing, audit logging, and revocation mechanisms, and interactions should apply appropriate transport-layer security. It is worth noting autonomous agents may act persistently and at machine speed, which creates several additional security challenges.

Specific aspects of security will be discussed in more detail in this document.

7.1. Agent Impersonation during Discovery

An adversary may attempt to impersonate a legitimate AI agent during the discovery phase, registering false endpoints or replaying previously valid credentials to obtain unauthorized access tokens or sensitive data.

Agents must perform authenticated discovery using DNS or another verifiable directory service and complete a cryptographically signed handshake before establishing trust. Implementations might need to reject interactions with agents that cannot present verifiable handshake proofs.

7.2. Excessive Attribute Disclosure

During discovery and authorization, agents may disclose more identifying information than is necessary, creating privacy risks and enlarging the surface of attack. Privacy-preserving credential mechanisms may be used to reveal only the minimum attributes necessary for the transaction. Authorization servers and relying parties should avoid requesting attributes not strictly required for policy enforcement.

7.3. Token Replay and Intermediary Exfiltration

Access tokens exchanged between agents or through API proxy servers (e.g., MCP) may be intercepted or replayed by unauthorized intermediaries, resulting in impersonation or unauthorized resource access.

A method should be used so that access tokens may be sender-constrained using mechanisms. Furthermore, tokens may be configured to have short lifetimes or be scoped narrowly to the required operation. For sensitive or human-triggered actions, a method to ensure that leaked tokens cannot independently authorize sensitive actions would be required.

8. IANA Considerations

TBD.

9. Acknowledgements

10. References

10.1. Normative References

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <https://www.rfc-editor.org/rfc/rfc6749>.

[RFC7591] Richer, J., Ed., Jones, M., Bradley, J., Machulak, M., and P. Hunt, "OAuth 2.0 Dynamic Client Registration Protocol", RFC 7591, DOI 10.17487/RFC7591, July 2015, <https://www.rfc-editor.org/rfc/rfc7591>.

10.2. Informative References

[I-D.ietf-oauth-v2-1] Hardt, D., Parecki, A., and T. Lodderstedt, "The OAuth 2.1 Authorization Framework", Work in Progress, Internet-

Draft, draft-ietf-oauth-v2-1-14, 19 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-v2-1-14>.

[I-D.narajala-ans]
Huang, K., Narajala, V. S., Habler, I., and A. Sheriff, "Agent Name Service (ANS): A Universal Directory for Secure AI Agent Discovery and Interoperability", Work in Progress, Internet-Draft, draft-narajala-ans-00, 16 May 2025, <https://datatracker.ietf.org/doc/html/draft-narajala-ans-00>.

[I-D.narvaneni-agent-uri]
Narvaneni, Y., "The agent:// Protocol -- A URI-Based Framework for Interoperable Agents", Work in Progress, Internet-Draft, draft-narvaneni-agent-uri-02, 15 October 2025, <https://datatracker.ietf.org/doc/html/draft-narvaneni-agent-uri-02>.

[I-D.oauth-ai-agents-on-behalf-of-user]
Thilina, T. and A. Dissanayaka, "OAuth 2.0 Extension: On-Behalf-Of User Authorization for AI Agents", Work in Progress, Internet-Draft, draft-oauth-ai-agents-on-behalf-of-user-02, 25 August 2025, <https://datatracker.ietf.org/doc/html/draft-oauth-ai-agents-on-behalf-of-user-02>.

[I-D.rosenberg-aiproto-framework]
Rosenberg, J. and C. F. Jennings, "Framework, Use Cases and Requirements for AI Agent Protocols", Work in Progress, Internet-Draft, draft-rosenberg-aiproto-framework-00, 19 October 2025, <https://datatracker.ietf.org/doc/html/draft-rosenberg-aiproto-framework-00>.

[I-D.rosenberg-oauth-aauth]
Rosenberg, J. and P. White, "AAuth - Agentic Authorization OAuth 2.1 Extension", Work in Progress, Internet-Draft, draft-rosenberg-oauth-aauth-01, 19 October 2025, <https://datatracker.ietf.org/doc/html/draft-rosenberg-oauth-aauth-01>.

[TR22.870] 3GPP, "Study on 6G Use Cases and Service Requirements", n.d..

Author’s Address

Kehan Yao
China Mobile
Email: yaokehan@chinamobile.com