OAuth2.0 Extension for Multi-AI Agent Collaboration

A Proposal for Task-Oriented Authorization

Yurong Song

March 2026

Background: Multi-AI Agent Collaboration

Core Idea

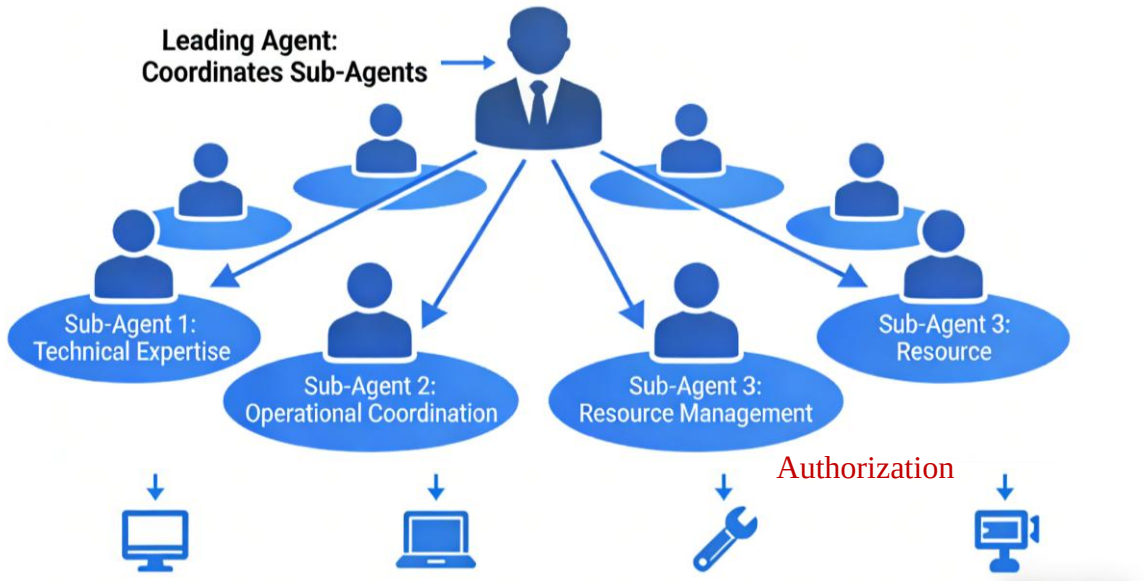
Complex tasks require collaboration among multiple AI agents with specialized capabilities.

Coordination Mechanism

A leading agent coordinates sub-agents to form a specialized Task Group for goal achievement.

Real-world Example

"Real-time health advice" integrates data collection, status prediction, and advice generation agents.



Multi-AI Agent Collaboration

Authorization Challenges in Multi-Agent Collaboration

Inefficiency

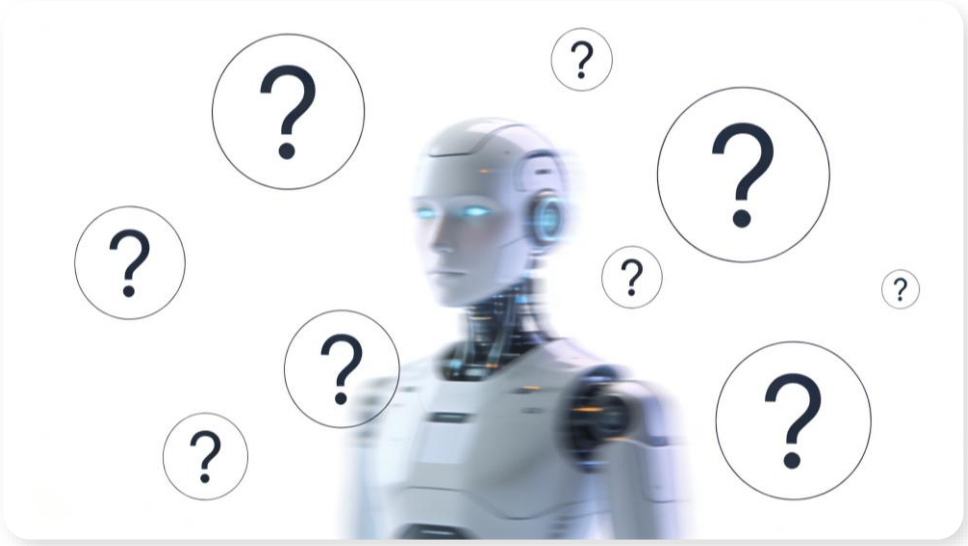
Each sub-agent applying for a token individually causes frequent interactions and high overhead.

Complexity

Managing permissions for a dynamic group of agents is difficult and error-prone.

Traceability

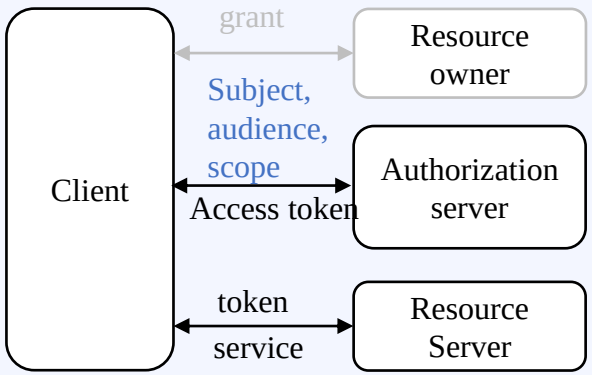
Lack of clear accountability for actions performed by the task group.



AI Agent Identity & Authorization Confusion

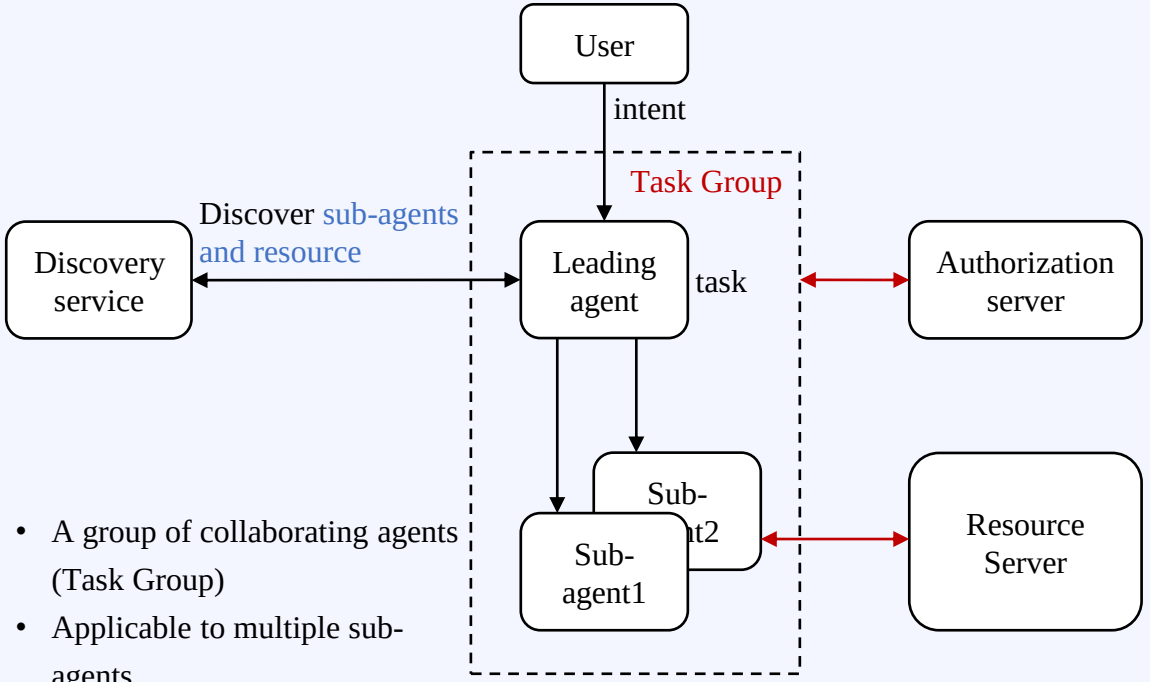
Authorization Challenges in Multi-Agent Collaboration

OAuth 2.0



- Individual client application
- Bound to a single client
- Token per client

Multi-Agent Collaboration



- A group of collaborating agents (Task Group)
- Applicable to multiple sub-agents
- One token for the entire group

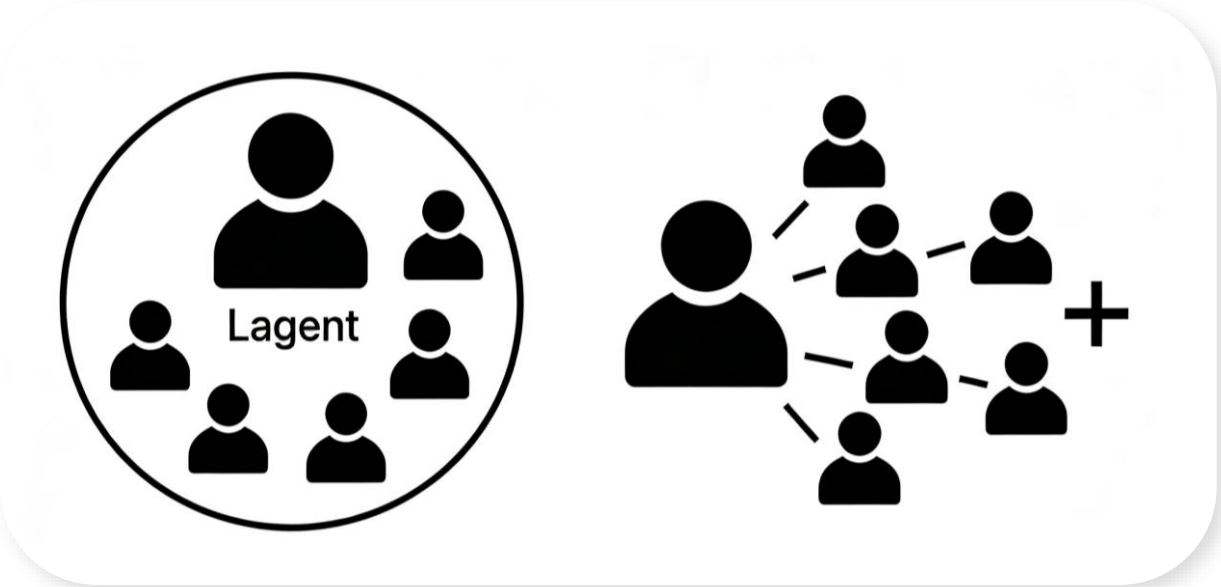
Our Solution: Task Group Authorization

1. Static Task Group Authorization

- The leading agent select sub-agents to form a task group in advance (static policy).
- The leading agent applies for tokens for all sub-agents.

2. Dynamic Task Group Authorization

- The leading agent selects sub-agents during task execution. New sub-agents may join the task group (dynamic policy).
- The leading agent applies for a task token and issues Task Credentials to sub-agents.



Static & Dynamic Task Group Authorization

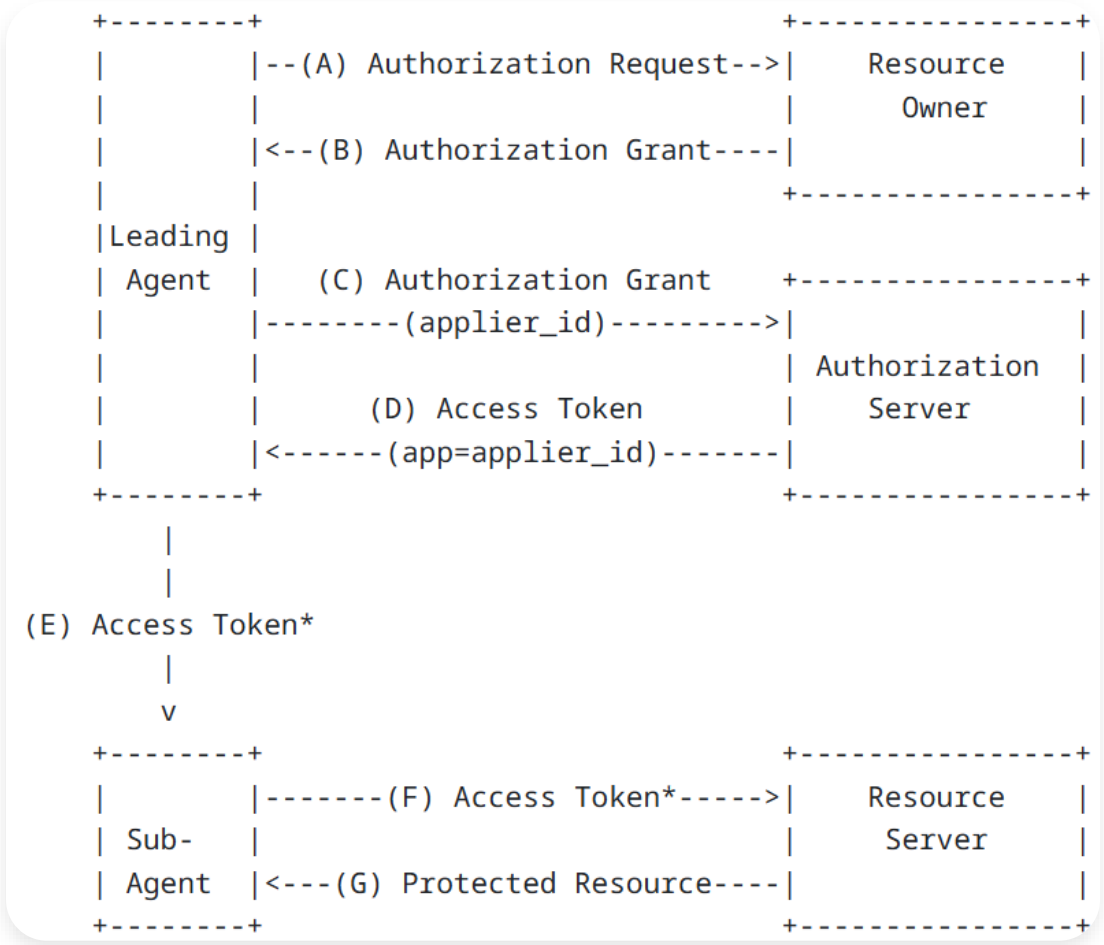
Static Task Group Authorization: Overview

Core Idea

The leading agent acts as an Applier to request a single access token for the entire pre-defined task group.

Key Extension

Introduce applier_id to uniquely identify the leading agent initiating the request.
AS validates that the leading agent is capable of task group authorization.
sub-agents validate that the applier ID refers to a trustworthy leading agent



Static Task Group Authorization Workflow

Static Task Group Authorization: Access Token

The access token is a JWT with an additional **app** claim and multiple **sbj-aud-scope** pairs.

New claim: app

Identifies the Leading Agent (Applier ID)

sbj-aud-scope

Defines permissions for Sub-Agents

```
{
   "iss": "auth-server.example.com",
   "app":"leading-agent-123", // Applier ID
   "sbj":"sub-agent-A“
   "aud":"api.health.com“
   "scope":"read",
   "sbj":"sub-agent-B“
   "aud":"api.advice.com“
   "scope":"write"
}
```

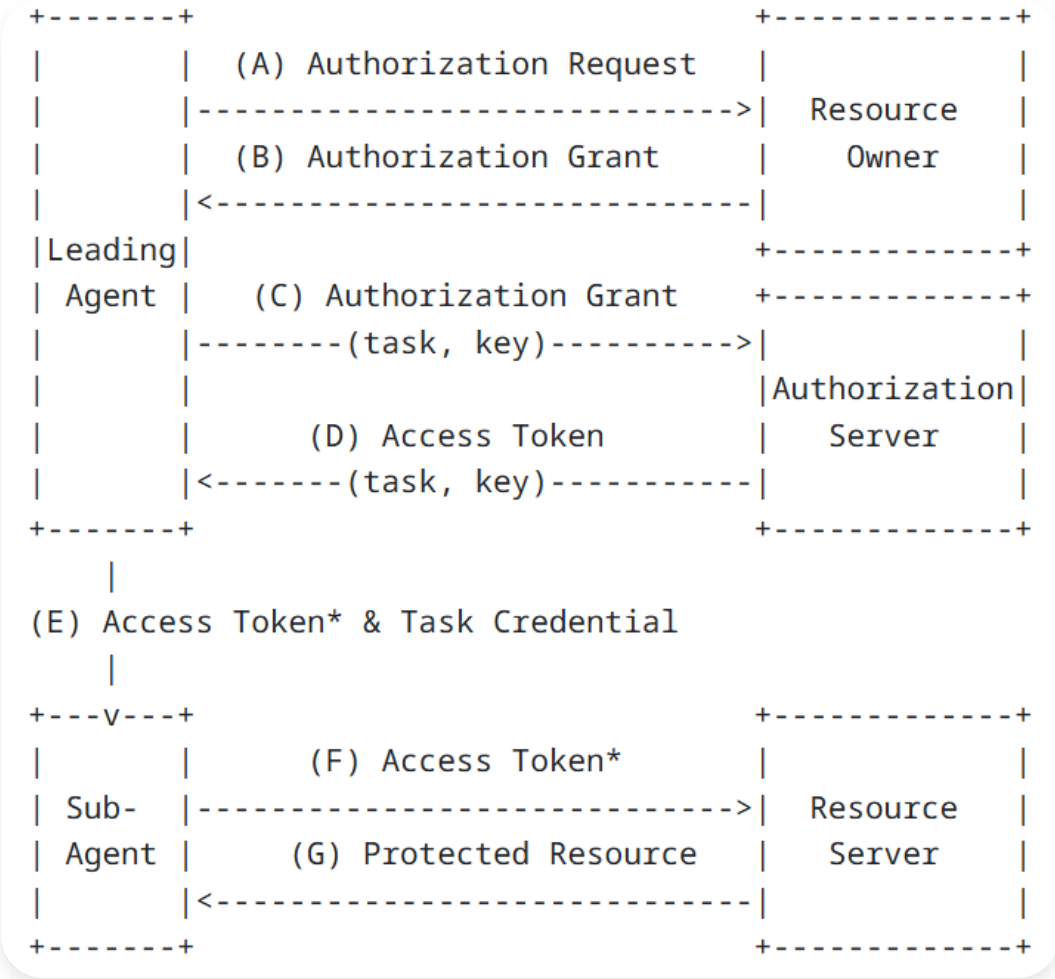
Dynamic Task Group Authorization: Overview

Core Idea

The leading agent first obtains a Task Token for a specific task, then issues Task Credentials to dynamically selected sub-agents.

Key Extension

Task Credentials are bound to the Task Token and grant permissions for the specific task only, ensuring security.
AS validates that the leading agent is capable of task group authorization, and holds keys.
Sub-agents validate that the token is related to the credential



Dynamic Task Group Authorization Flow

Dynamic Task Group Authorization: Access Token

The access token is a JWT with an additional **att** claim to to denote that authorization is dedicated to a specific task, where the **sbj** is used to index keys that will be used to endorse the attributes

New claim: att

The attribute of the sub-agent that is required to permit the sub-agent to request resources

sbj

The identity of the leading agent, also can be the public key related to the specific task

```
{
   "iss": "auth-server.example.com",
   "sbj":"MIIBIjANBgkqhkiG9…"//public key for task, or leading agent ID
   "aud":"api.health.com"
   "scope":"read"
   "att": "Real-time health advice" // task ID
}
```

What is a Task Credential?

Definition

A lightweight credential issued by the leading agent to a sub-agent, enabling delegated access.

Security Binding

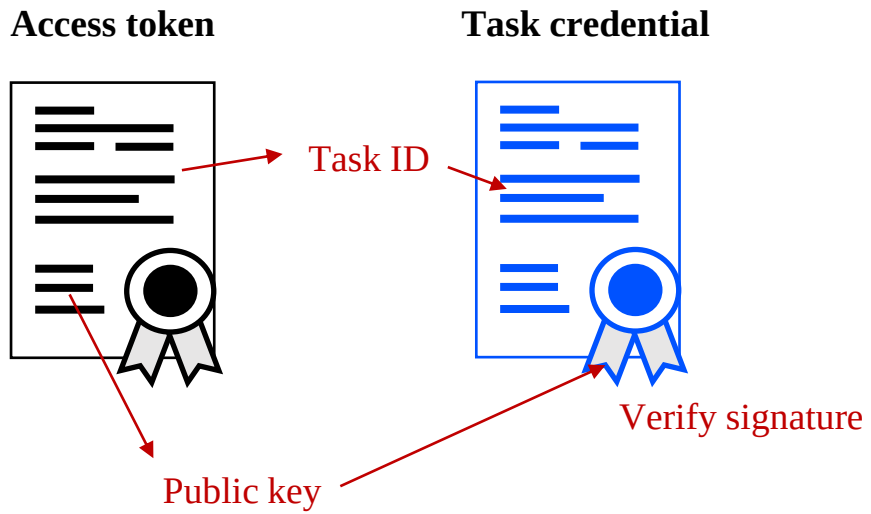
Cryptographically bound to the main Task Token, ensuring authenticity and preventing tampering.

Credential Structure

- **Leading agent ID:** Identity of the credential issuer
- **Task ID:** Unique identifier/description for the task
- **Sub-agent ID:** Identity of the receiving agent
- **Token hash:** Hash of the related access token
- **Signature:** Digital signature from the leading agent (generated with the private key related to the public key in the access token)

Dynamic Task Group Authorization: Token Verification

- Sub-agent presents the **Task Token** and **Task Credential** to the Resource Server.
- Resource Server verifies the credential (validity, signature, subject ID=sub-agent ID).
- Resource Server verifies the **Task Token** (validity, signature, scope, audience).
- Resource Server verifies the relationship of **Task Token** and **Task Credential** (task ID, token hash, key).



Verification of the relationship

Comparison of the Authorization Methods

Aspect	DPoP	Dynamic Task Group Authorization
Effect	Cryptographically binds access tokens to client public keys	Cryptographically binds access tokens to client attributes
Access token extension	Public key	Attribute (task)
		Public key used to endorse the credential
Credential	Public key credential	Attribute credential (task)
Issuer	Authority	Leading agent

Comparison of the Authorization Methods

Aspect	Static Task Group Authorization	Dynamic Task Group Authorization
Group Formation	Pre-defined (Static)	Dynamic (On-the-fly)
Token Type	Single multi-purpose token	Task Token + Task Credentials
Efficiency	High (one token request)	High (one task token + lightweight credentials)
Flexibility	Low	High
Use Case	Predictable tasks	Adaptive, complex tasks

Key Advantages

Improved Efficiency

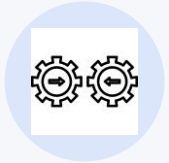
Reduces interactions with the Authorization Server, streamlining the authentication process.

Enhanced Flexibility

Supports both static and dynamic collaboration scenarios, adapting to diverse business needs.

Simplified Management

Centralized authorization by the leading agent, reducing administrative overhead.



Maintained Compatibility

Extends OAuth 2.0 without breaking existing workflows, ensuring a smooth transition.

Conclusion & Discussion

Authorization Challenges

Multi-AI agent collaboration presents new and unique authorization challenges like inefficiency, complexity and traceability.

OAuth 2.0 Extensions

The draft proposes two OAuth 2.0 extensions specifically designed to enable efficient, task-oriented authorization for AI agents.

Static & Dynamic Solutions

Static authorization for pre-defined groups, and dynamic authorization using Task Credentials for flexible groups.

Benefits & Compatibility

The solution significantly improves efficiency, flexibility, and manageability while maintaining full compatibility with existing systems.

- Is there existing authorization method suitable for multi-agent collaboration?
- Is it feasible to extend access token with “applier” and “attribute”, or reuse existing parameters?

THANK YOU

Collaboration and comment is welcomed

songyurong1@huawei.com