Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 10 August 2026

J. Diaconu
M. Yannuzzi
H. Muyal
F. Brockners
N. Kale
Cisco
A. Agarwal
Skyfire
J. Hickman
Ory Corp
A. Lal
AWS

6 February 2026

Cross-Domain AuthZ Information sharing for Agents
draft-diaconu-agents-authz-info-sharing-00

Abstract

Distributed Multi-Agent Systems consist of Agents and MCP Servers operating across multiple administrative domains, each with its own Identity Providers (IdPs) and Authorization Servers (AS).

This document discusses the challenges and solution approaches for sharing authorization information securely and flexibly across domains, including the use of dynamic identity, interoperable claims, and verifiable credentials.

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 10 August 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

Distributed Multi-Agent Systems, consisting of Agents and Model Context Protocol (MCP) Servers, are increasingly deployed across multiple administrative domains. Each domain typically operates its own Identity Providers (IdPs) and Authorization Servers (AS), creating a fundamental challenge: how can an agent from Organization A securely prove its identity and authorization to resources in Organization B?

While OAuth 2.0 and related specifications provide robust authorization frameworks for human users and pre-registered clients, the emergence of autonomous agent systems operating across organizational boundaries presents new challenges that existing mechanisms do not fully address.

This challenge is distinct from traditional cross-domain identity federation for human users. Humans rarely delegate authority through multiple organizational boundaries. Agents routinely do so--an orchestrator agent in Domain X may invoke a tool agent in Domain Y, which accesses an MCP server in Domain Z. Existing authorization mechanisms such as OAuth 2.0 Dynamic Client Registration [RFC7591] and Token Exchange [RFC8693] assume single-domain operation or pre-established relationships, and do not address the dynamic, multi-hop delegation patterns that characterize modern agentic systems.

This document analyzes the cross-domain agent authorization problem and evaluates solution approaches. The proposed solution extends OAuth 2.0 Client-ID Metadata [I-D.ietf-oauth-client-id-metadata-document] with W3C Verifiable Credentials [W3C.VCDM2.0] to enable dynamic identity establishment without pre-registration across domains, cryptographically verifiable authorization claims that travel with the agent, multi-hop delegation with monotonic scope narrowing and accountability chains, and selective disclosure of sensitive authorization attributes.

The document is structured as follows: Section 3 defines key terminology. Section 4 presents use cases demonstrating cross-domain scenarios. Section 8.1.1 identifies threats specific to cross-domain authorization sharing. Section 5 establishes requirements and evaluates solution approaches. Section 8.1 covers security considerations.

2. Conventions Used in This Document

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 [RFC2119] and [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Abbreviations / Definitions

Agent An autonomous entity that performs tasks on behalf of a user or another program within a multi-agent system.

MCP Server A program that exposes specific capabilities, like data access or tools, to AI models through the Model Context Protocol (MCP).

AS (Authorization Server) A server that issues authorization tokens to clients after successfully authenticating them.

IdP (Identity Provider) A service that creates, manages, and verifies digital identities for users and service users, often for accessing multiple applications and services.

Subject The principal (user or service).

Claim Essentially a piece of information asserted about a subject.

Domain Scope over which a specific, uniform set of security policies and user management rules are enforced.

VCDM 2.0 Verifiable Credentials Data Model 2.0, a W3C standard for expressing verifiable credentials on the web.

VC Verifiable Credential, a digital credential that is cryptographically secure and can be verified independently.

SD-JWT Selective Disclosure JSON Web Token, a JWT that allows the holder to disclose only specific parts of the token.

DID Decentralized Identifier, a new type of identifier that enables verifiable, self-sovereign digital identities.

4. Use Cases

The following use cases demonstrate scenarios where cross-domain agent authorization presents challenges not addressed by existing single-domain mechanisms.

4.1. Cross-Domain Tool Invocation

A coding assistant agent operated by Developer Tools Inc. needs to invoke a code review tool provided by Code Analysis Corp. The tool is exposed as an MCP Server protected by Code Analysis Corp's Authorization Server.

Cross-Domain Challenge Developer Tools Inc.'s agent must present credentials that Code Analysis Corp's AS can verify, despite having no pre-existing trust relationship. The AS must determine what operations the agent may perform based on claims issued by a foreign domain.

Requirements Verifiable agent and MCP server identity across domain boundaries, authorization claims that are interpretable by the receiving domain.

4.2. Enterprise Partnership Integration

Company A partners with Company B to provide integrated customer support services. Company A deploys a support agent that needs to access Company B's customer ticketing system and knowledge base to resolve customer issues.

Cross-Domain Challenge Company A's agent authenticates using Company A's IdP and holds a Verifiable Credential issued by Company A. Company B's Authorization Server has no trust relationship with Company A's IdP--it cannot verify Company A's tokens. Company B needs a mechanism to verify the agent's identity and authorization claims without requiring bilateral federation setup for every partner.

Requirements Dynamic identity verification across domains, verifiable compliance attestations, and scope restrictions that Company B's AS can enforce based on Company A's authorization grants.

4.3. Multi-Vendor Workflow Orchestration

A security operations workflow involves agents from multiple vendors: an orchestrator agent from Security Vendor X coordinates a threat intelligence agent from Vendor Y and a remediation agent from Vendor Z. The orchestrator must delegate appropriate permissions to each specialized agent.

Cross-Domain Challenge The orchestrator in Domain X must delegate

authority to agents in Domains Y and Z. Each receiving domain must verify the delegation chain, ensure the delegated scope does not exceed what the originating principal authorized, and maintain an auditable record of the delegation chain. Receiving domains must ensure that the authorization decisions do not rely on revoked, suspended, or superseded delegations by performing runtime delegation chain verification according to the local policy.

Requirements Multi-hop delegation with scope narrowing, cross-domain accountability chains, privacy-preserving identity claims, and domain-specific policy enforcement at each trust boundary crossing.

4.4. Supply Chain Data Exchange

A manufacturer's inventory management agent needs to query suppliers' availability systems. Multiple suppliers operate their own authorization infrastructure. The agent must access each supplier's MCP servers with appropriate credentials.

Cross-Domain Challenge Each supplier's AS must verify the manufacturer's agent without requiring manual pre-registration. Dynamic Client Registration [RFC7591] would require the agent to register separately with each supplier's AS, creating operational overhead and not providing cryptographic verification of the agent's authorization claims from its home domain. The agent may need different permission levels at different suppliers based on partnership agreements. Suppliers must be able to audit which manufacturer agents accessed their data.

Requirements Dynamic agent provisioning across multiple foreign domains, verifiable identity without pre-registration, and domain-specific scope mappings.

4.5. Regulatory Compliance Across Jurisdictions

A financial services agent operating in the EU needs to access data from a partner institution in the US. Both jurisdictions have different regulatory requirements.

Cross-Domain Challenge The receiving domain's AS must verify that the requesting agent's compliance attestations meet local policy requirements. The VC issued by the originating domain asserts compliance with GDPR; the receiving domain must determine whether this satisfies its own data handling requirements without assuming equivalence of compliance frameworks.

Requirements Verifiable compliance attestations that receiving domains can evaluate against local policies, and data sensitivity clearance, verification across regulatory boundaries.

5. Requirements Summary

A system for cross-domain sharing of authorization information of agents needs to meet the following set of requirements:

* *Dynamic Identity for Agents and MCP Servers:* The Agents and MCP Servers can onboard dynamically and get an assigned identity in the IdP.

* *Interoperability Across Domains:* The system must enable seamless interaction between Agents across different domains.

* *Flexible Definition of Claims:* Claims should be adaptable as they may vary from organization to organization. They may include information related to authorization, compliance requirements, chains of delegation, and other domain-specific concerns.

* *Dynamic Management of Authorization Information:* The system must allow creation, removal, updating, and deletion of authorization information for agents, as agents can be highly dynamic.

* *Security and Privacy:* Ensuring secure and private sharing of authorization information across domains.

* *Compliance and Auditability:* Support for compliance with regulatory standards and auditability of authorization exchanges.

6. Solution Approaches

6.1. Solution Approach 1: Dynamic Client Registration (RFC 7591)

Description: Dynamic Client Registration as outlined in [RFC7591] allows clients to register with an AS dynamically, facilitating the management of client credentials and metadata.

Discussion:

Requirement	Met or Partially Met	Not Met
Dynamic Identity for Agents and MCP Servers	Met	
Interoperability Across Domains	Partially Met (Granted by the Dynamic Registration, however no common ground between AS)	
Flexible Definition of Claims		Not Met (OAuth2 claims only)
Dynamic Management of Authorization Information		Not Met (Delete or Update not supported)
Security and Privacy		Not Met (Claims are shared but no common schema)
Compliance and Auditability	Partially Met (Claims are shared, APIs are auditable, but no regulatory standards are defined for the schema)	

Table 1

6.2. Solution Approach 2: SCIM extensions

[I-D.abbey-scim-agent-extension] proposes extending the SCIM (System for Cross-Domain Identity Management) protocol to standardize the provisioning, management, and governance of AI agents and other digital workers.

[I-D.wahl-scim-agent-schema] is a companion document to the draft-abbey-scim-agent-extension. Its purpose is to define the specific SCIM schema and attributes required to represent AI agents (digital workers) and agentic applications within the SCIM framework.

Description: Provision Agents / MCP Servers through a SCIM extension.

Discussion:

Requirement	Met or Partially Met	Not Met
Dynamic Identity for Agents and MCP Servers	Partially Met (SCIM support is limited and with constraints, example for Azure: “Subsequent syncs are triggered every 20-40 minutes” .)	
Interoperability Across Domains	Partially Met (SCIM is supported and schema is well defined)	
Flexible Definition of Claims	Met (Defined by the schema)	
Dynamic Management of Authorization Information	Partially Met (SCIM support is limited and with constraints, example for Azure: “Subsequent syncs are triggered every 20-40 minutes” .)	
Security and Privacy	Partially Met (A schema is defined but Selective Disclosure not supported)	
Compliance and Auditability	Partially Met (Claims are shared, APIs are auditable, but no regulatory standards are defined for the schema)	

Table 2

6.3. Solution Approach 3: Client-ID Metadata

Description: Client ID Metadata [I-D.ietf-oauth-client-id-metadata-document] is a method for OAuth clients to identify themselves using a URL, which points to a JSON file containing their OAuth metadata. Instead of requiring pre-registration, an authorization server can fetch this JSON document at the client’s provided URL to provision the identity.

Discussion:

Requirement	Met or Partially Met	Not Met
Dynamic Identity for Agents and MCP Servers	Met	
Interoperability Across Domains	Met	
Flexible Definition of Claims		Not Met* (client_id is common, all the rest is limited to OAuth2 claims)
Dynamic Management of Authorization Information	Met	
Security and Privacy	Partially Met** (A schema is defined but Selective Disclosure not supported)	
Compliance and Auditability	Partially Met** (Claims are shared, APIs are auditable, but no regulatory standards are defined for the schema)	

Table 3

**Flexible Definition of Claims:* The OAuth claims in the draft follow the claims exposed and defined by IANA OAuth Parameters. Hence, the metadata fields are limited to a fixed set of claims. This makes it difficult to convey richer features about the client, including provenance details, compliance attestations, or contextual trust scores without introducing vendor-specific extensions, thereby reducing portability and interoperability.

***Security and Privacy, Compliance and Auditability:* All metadata is published to a publicly accessible endpoint. There is no native ability to keep certain attributes private, apply selective disclosure, or restrict visibility based on trust relationships. Sensitive operational data or compliance-related data must be handled outside the metadata framework, leading to fragmented trust models. There is no mechanism for revocation and expiration.

6.4. Solution Approach 4: Client-ID Metadata + W3C Verifiable Credentials Extension

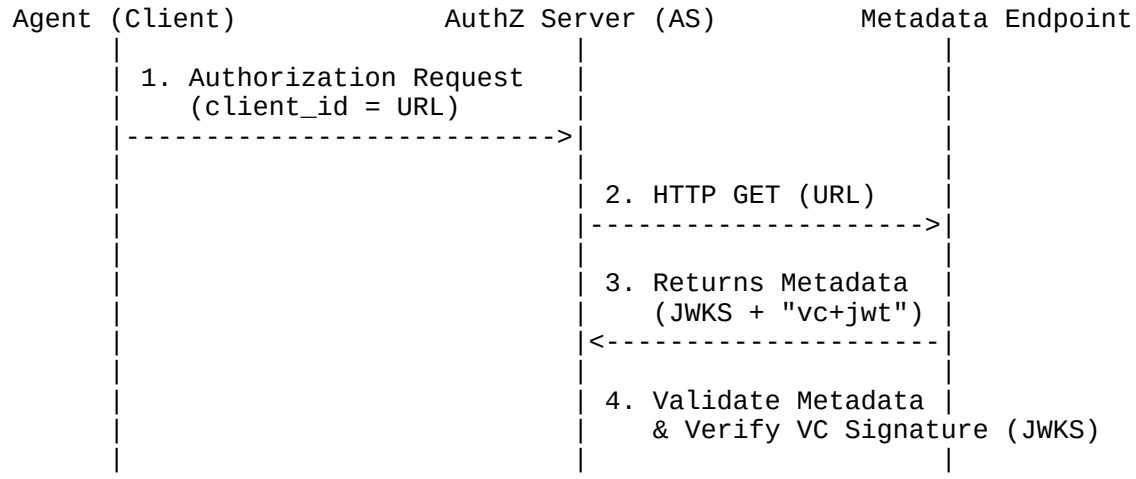

Figure 1: Client-ID Metadata with VC Verification Flow

Description: The proposed solution is based on Solution 3, but enhances the supported claims with a new claim, "vc+jwt", based on the Verifiable Credentials Data Model 2.0 from W3C.

vc+jwt A standard method used in decentralized identity to package and secure a digital credential. The Verifiable Credential (VC)—the actual data proving a fact (e.g., identity or degree)—is encapsulated within a JSON Web Token (JWT), which provides a cryptographic signature from the issuer. This signature allows any third party (the Verifier) to instantly confirm the credential's authenticity and integrity. Modern formats like SD-JWT VC further enhance this by enabling the holder to share only specific parts of the credential (Selective Disclosure) to protect privacy.

Discussion:

Requirement	Met or Partially Met	Not Met
Dynamic Identity for Agents and MCP Servers	Met	
Interoperability Across Domains	Met (1)	
Flexible Definition of Claims	Met (2)	
Dynamic Management of Authorization Information	Met	
Security and Privacy	Met (3)	
Compliance and Auditability	Met (4)	

Table 4

Detailed explanations:

The W3C Verifiable Credentials Data Model 2.0 addresses the shortcomings of the previous Solution 3, by introducing a cryptographically verifiable and standards-based way to assert identities, capabilities, and compliance attestations. It supports selective disclosure to protect sensitive information, enables time-bound and revocable authorizations, and ensures provenance tracking for audit and regulatory compliance. They are part of a broader decentralized identity (DID) specification.

Using the W3C Verifiable Credentials Data Model 2.0 for Agents, MCP Clients, and MCP Servers provides a standardized, cryptographically secure way to establish trust, delegate authority, and ensure interoperability across different systems.

It enables Agents, MCP Clients, and MCP Servers to prove, among other things, their identities, identity issuers and provenance, supported skills and permitted actions through verifiable proofs, including support for fine-grained and time-bound access control, while maintaining transparency enabling audit trails for compliance.

Additionally, it enhances privacy through selective disclosure and privacy-preserving proofs (e.g., allowing Agents to prove skills and authorized actions without exposing unnecessary details).

The Verifiable Credentials Data Model 2.0 is a W3C standard that fits the needs expressed above:

(1) Enhanced Interoperability: VCDM 2.0 provides a standardized, extensible way to express identities, capabilities, and compliance data across different systems. Its use of DIDs and schema-governed claims ensures that Agents, MCP Clients, and MCP Servers can verify and exchange trust information, enabling consistent interoperability across domains and platforms.

(2) Flexible Definition of Claims: VCDM 2.0 allows arbitrary, schema-governed claims that can represent complex concepts such as provenance, software supply-chain attestations (e.g., SLSA), security posture, audit certifications (SOC2, ISO-27001), or dynamic trust signals. These claims can be extended without breaking interoperability, thanks to widely adopted JSON-LD and schema registries.

(3) Security and Privacy: Credentials are signed using verifiable cryptographic proofs (e.g., JSON Web Signatures, Data Integrity proofs). This ensures that Agents, MCP clients, and MCP servers can verify the authenticity of claims without relying solely on HTTPS endpoints or centralized registries. Through techniques like BBS+ signatures, holders can disclose only the subset of claims required for a given authorization flow. This is essential for scenarios where Agents must prove capabilities (e.g., "I’m allowed to execute this skill") without revealing unnecessary or sensitive identity attributes.

(4) Compliance and Auditability: VCDM 2.0 enables cryptographically verifiable audit trails by binding each credential to an issuer, timestamp, and revocation status. Compliance attestations can be embedded directly in credentials, and real-time revocation checks ensure they remain valid. This provides trustworthy provenance, regulatory-grade accountability, and reliable lifecycle tracking across distributed systems.

6.4.1. Delegation Semantics for Cross-Domain Multi-Agent Workflows

Multi-agent systems frequently involve chains of delegation where Agent A invokes Agent B, which may invoke Agent C—potentially spanning multiple administrative domains. This section defines authorization semantics for such delegation chains, addressing the novel challenges of cross-domain delegation.

Humans rarely delegate authority through multiple levels across organizational boundaries. Agents routinely do so. Existing delegation mechanisms ([RFC8693] Token Exchange 'act' claims) assume single-domain operation and do not address cross-domain VC-based delegation with verifiable accountability chains.

6.4.1.1. Delegation Models

This specification supports two delegation models:

Direct Invocation Agent A directly invokes MCP Server M using A's own credentials. No delegation occurs; authorization is based solely on A's VC.

Delegated Invocation (Cross-Domain) Agent A in Domain X invokes Agent B in Domain Y to perform a task. B acts on A's behalf (and transitively, on behalf of A's originating principal) when accessing resources. The authorization decision considers both A's delegation grant and B's own credentials, applying Domain Y's policies.

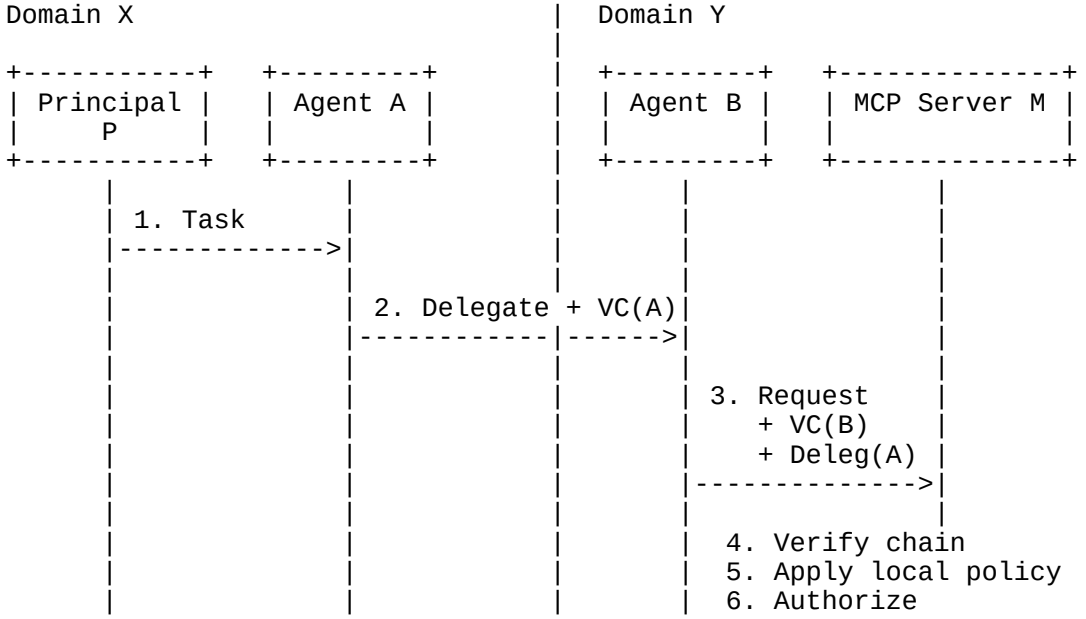

Figure 2: Cross-Domain Delegation Flow

6.4.1.2. Scope Derivation Rules

When Agent A delegates to Agent B (potentially across domains), B's effective scope MUST be computed as the intersection of:

1. A's delegatable scope: The subset of A's permissions that A is authorized to delegate
2. B's intrinsic scope: The permissions B holds in its own VC
3. Explicit delegation grant: What A actually delegated to B for this invocation
4. Receiving domain policy: The maximum scope the receiving domain permits for this cross-domain delegation

```
Effective_Scope(B) = A.delegatable_scope
                     INTERSECT B.intrinsic_scope
                     INTERSECT Delegation.granted_scope
                     INTERSECT Receiving_Domain_Policy
```

Figure 3: Scope Derivation Formula

This ensures:

* B cannot exceed A's authority (no privilege amplification)
* B cannot exceed its own authority (intrinsic limits apply)
* A can further restrict B for specific tasks (explicit grant)
* The receiving domain retains sovereignty over its resources (local policy)

Implementations MUST enforce monotonic scope narrowing: at each delegation step, effective scope can only decrease or remain the same, never increase.

This intersection semantics aligns with capability-based security principles (see: Mark Miller's Principle of Least Authority, Google Macaroons, Biscuit authorization tokens) and extends [RFC8693] Token Exchange delegation semantics to cross-domain VC-based scenarios.

6.4.1.3. Accountability Chain Construction

Each delegation step MUST produce a verifiable record enabling cross-domain forensic reconstruction.To support revocation and scope-change handling, delegation records SHOULD enable downstream verifiers to identify and validate upstream delegation elements so that authorization decisions do not depend on revoked or superseded delegation claims. The delegation record MUST include:

Field	Description	Cross-Domain Relevance
delegator_id	Agent A's id from its VC	Enables tracing to originating domain
delegator_domain	Domain that issued A's VC	Identifies trust relationship to verify
delegatee_id	Agent B's id from its VC	Identifies receiving agent
delegatee_domain	Domain that issued B's VC	Identifies receiving domain
timestamp	Time of delegation	Temporal ordering for forensics
granted_scope	Scopes delegated	Audit of intended authorization
expiration	Delegation validity period	MUST NOT exceed delegator's VC expiration
chain_hash	Cryptographic binding to prior chain	Ensures chain integrity across domains

Table 5: Delegation Record Fields

The complete accountability chain enables:

1. Cross-domain attribution: Determining which principal originated a request that traversed multiple domains

2. Forensic reconstruction: Rebuilding the sequence of delegations across domain boundaries

3. Coordinated revocation: Identifying downstream delegations in other domains that must be invalidated

6.4.1.4. Delegation Depth Limits

Implementations SHOULD enforce maximum delegation depth. Cross-domain delegations SHOULD count toward depth limits regardless of whether they're same-domain or cross-domain.

Receiving domains MAY impose stricter depth limits than the originating domain. When a delegation would exceed the receiving domain's depth limit, the AS MUST reject the request with an error indicating the depth violation.

6.4.1.5. Delegation Claims in VCs

To express delegation authority in VCs, the following claims are defined for credentialSubject:

```
{
  "credentialSubject": {
    "id": "https://example.com/agents/agent-a",
    "delegation": {
      "permitted": true,
      "maxDepth": 2,
      "crossDomainDelegation": {
        "permitted": true,
        "allowedDomains": ["partner.example.org"],
        "deniedDomains": ["competitor.example.com"]
      },
      "allowedDelegateeTypes": ["tool-agent", "workflow-agent"],
      "scopeRestrictions": {
        "nonDelegatable": ["admin:*", "pii:write"],
        "crossDomainNonDelegatable": ["internal:*"]
      }
    }
  }
}
```

Figure 4: Delegation Claims Example

delegation.permitted Boolean indicating whether this agent may delegate. Default: false.

delegation.maxDepth Maximum additional delegation hops this agent may initiate. Default: 0.

delegation.crossDomainDelegation.permitted Whether cross-domain

delegation is allowed at all.

delegation.crossDomainDelegation.allowedDomains Domains to which delegation is permitted (wildcards allowed).

delegation.scopeRestrictions.crossDomainNonDelegatable Scopes that may be delegated within the same domain but MUST NOT be delegated across domain boundaries.

6.4.2. Agent Authorization Verifiable Credential Profile

This section defines a Verifiable Credential profile that enables interoperable cross-domain agent authorization. The profile specifies claims that Authorization Servers in different domains can reliably interpret.

W3C VCDM 2.0 is intentionally flexible. For cross-domain interoperability, domains must agree on claim semantics. This profile provides that common language.

6.4.2.1. Profile Objectives

The Agent Authorization VC Profile (AAVC) enables:

* Cross-domain interoperability: ASs in different domains can parse and evaluate the same VC
* Semantic clarity: Claim meanings are defined, reducing policy skew risk
* Extensibility: Organizations can add custom claims without breaking base interoperability
* Selective disclosure compatibility: Claims are structured to support SD-JWT VC presentation

6.4.2.2. Required Claims for Cross-Domain Interoperability

The credentialSubject MUST include these claims for a VC to be recognized across domains:

id (required) Globally unique identifier for the agent. SHOULD be a dereferenceable URI pointing to client-id metadata.

agentType (required) Classification of the agent. Receiving domains use this to apply type-specific policies. Implementations MUST support:

* tool-agent: Provides specific tool/skill capabilities
* orchestrator-agent: Coordinates other agents
* workflow-agent: Executes multi-step workflows
* system-agent: Long-running infrastructure agent
* ephemeral-agent: Short-lived, single-task

authorizedScopes (required) Authorization scopes this agent may request. Receiving domains map these to local authorization policies.

issuerDomain (required) Administrative domain of the credential issuer. Receiving ASs use this to look up trust relationships and policy mappings.

6.4.2.3. Recommended Claims for Production Deployments

dataSensitivityClearance (recommended) Maximum data sensitivity level this agent may access. Standard values:

* public - Publicly available information
* internal - Organization-internal, non-sensitive
* confidential - Business-sensitive information
* restricted - Highly sensitive, need-to-know basis
* regulated - Subject to regulatory controls (PII, PHI, PCI)

complianceAttestations (recommended) Compliance certifications relevant to cross-domain trust decisions. Should include standard, scope, validUntil, and optionally certificationBody.

operationalConstraints (recommended) Runtime constraints that receiving domains MAY enforce, including maxConcurrentSessions, maxRequestsPerMinute, allowedTimeWindows, and geofence restrictions.

delegation (recommended) Delegation permissions as defined in Section 6.4.1.5.

6.4.2.4. Extension Mechanism

Organizations MAY extend the profile with additional claims by:

1. Defining a JSON-LD context document for the extension
2. Including the extension context URI in the VC's @context array
3. Adding extension claims under a namespace prefix

Implementations receiving VCs with unrecognized claims SHOULD:

* Ignore unknown claims (do not reject the VC)
* Log unrecognized claims for operational visibility
* NOT make authorization decisions based on claims they cannot interpret

6.4.2.5. Interoperability Requirements

For cross-domain interoperability, implementations:

MUST:

* Support required claims (id, agentType, authorizedScopes, issuerDomain)
* Recognize the AgentAuthorizationCredential type
* Verify the credential proof before processing claims
* Reject VCs missing required claims

SHOULD:

* Support recommended claims
* Implement selective disclosure for cross-domain presentations
* Maintain logs of cross-domain VC presentations

MAY:

* Define and process organization-specific extensions
* Implement additional validation beyond profile requirements

7. Example

This section provides illustrative examples of how Client-ID Metadata can be extended with Verifiable Credentials for agent identity. These examples demonstrate the recommended Approach 4 and show how the concepts apply to scenarios similar to the travel booking use case described earlier. These examples are non-normative and intended to demonstrate the concepts discussed in this document.

7.1. Client-ID Metadata Example

The following example shows OAuth client metadata that includes a Verifiable Credential in the "vc+jwt" field:

```
{
  "client_id": "https://agents.example.com/agent-001/metadata.json",
  "client_name": "Customer Service Agent 001",
  "jwks_uri": "https://agents.example.com/agent-001/jwks.json",
  "token_endpoint_auth_method": "private_key_jwt",
  "grant_types": ["client_credentials"],
  "scope": "read:orders write:tickets",
  "vc+jwt": "eyJhbGciOiJFZERTQSIsInR5cCI6IkpXVCJ9.eyJAY29udGV4dCI6W..."
}
```

The "vc+jwt" field contains a JWT-encoded Verifiable Credential. The actual JWT would be significantly longer and contain the complete credential structure shown in the next subsection.

7.2. Verifiable Credential Example

The following example shows the decoded structure of a Verifiable Credential for an agent (before JWT encoding):

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://example.org/agent-credentials/v1"
  ],
  "type": ["VerifiableCredential", "AgentIdentityCredential"],
  "issuer": "did:web:issuer.example.com",
  "issuanceDate": "2025-01-15T10:30:00Z",
  "expirationDate": "2026-01-15T10:30:00Z",
  "credentialSubject": {
    "id": "https://agents.example.com/agent-001/metadata.json",
    "agentName": "Customer Service Agent 001",
    "agentVersion": "2.1.0",
    "agentType": "customer-service",
    "owner": "did:web:example.com",
```

```
      "authorizedCapabilities": [
        "order-lookup",
        "ticket-creation",
        "customer-notification"
      ],
      "delegatedCredentials":[
        {
          "id": "https://agents.partner.com/agent-111/metadata.json",
          "agentName":"Sec Agent 111",
          "delegatedBy":"https://agents.example.com/agent-001/metadata.json",
          "delegationChain": [
                  "urn:uuid:*****",
                  "urn:uuid:***"
          ],
          "scope": "read:sec-remediate:isolations",
          "authorizedCapabilities": [
                  "enrichment",
                  "isolation-request"
          ],
      "constraints": {
        "dpopNonceRequired": true
          }
        }
      ],
      "complianceFrameworks": ["SOC2-TypeII", "ISO-27001"],
      "delegationLevel": 1,
      "maxDelegationDepth": 2
    },
    "credentialStatus": {
      "id": "https://issuer.example.com/status/2024/12345",
      "type": "StatusList2021Entry",
      "statusPurpose": "revocation",
      "statusListIndex": "12345",
      "statusListCredential": "https://issuer.example.com/status/2024"
    }
  }
```

Note: The actual credential would include a "proof" section with cryptographic signatures. The proof section is omitted here for clarity, as it would add significant length to the example. The proof structure varies depending on the signature suite used (e.g., Ed25519Signature2020, JsonWebSignature2020).

8. Security Considerations

8.1. Security Considerations

This section addresses security considerations specific to cross-domain authorization information sharing. Implementations MUST also follow security guidance in [RFC6819], [W3C.VCDM2.0], and [I-D.ietf-oauth-client-id-metadata-document].

8.1.1. Threat Model for Cross-Domain Authorization Sharing

This section defines threats specific to sharing authorization information across administrative domain boundaries. General OAuth and VC security threats are addressed in [RFC6819] and [W3C.VCDM2.0]; this section focuses on threats that emerge or are amplified when credentials cross domain boundaries.

8.1.1.1. Protected Assets in Cross-Domain Scenarios

Beyond standard authorization assets, cross-domain sharing introduces additional assets requiring protection:

Cross-Domain Trust Relationships The established trust between domains that enables agents to operate across organizational boundaries. Compromise affects all agents relying on that trust relationship.

Shared Authorization Semantics The agreed-upon meaning of claims, scopes, and compliance attestations between domains. Semantic drift or manipulation can cause authorization decisions that violate one or both domains' policies.

Delegation Chains Spanning Domains When agent A in Domain X delegates to agent B in Domain Y, the chain itself becomes an asset requiring integrity protection across trust boundaries.

8.1.1.2. Cross-Domain Specific Threat Actors

Rogue Domain An entire domain that participates in the trust federation but issues fraudulent VCs or makes incorrect authorization decisions—either through compromise or malicious operation. Unlike a single rogue IdP, a rogue domain may control IdP, AS, and metadata endpoints coherently.

Cross-Domain Insider An administrator with legitimate access in Domain A who abuses cross-domain trust relationships to gain unauthorized access to Domain B's resources.

8.1.1.3. Cross-Domain Attack Categories

8.1.1.3.1. Trust Anchor Attacks

Metadata Endpoint Compromise An attacker gains control of the URL hosting an agent's client-id metadata document and substitutes malicious metadata or VCs. Because the metadata URL is the trust anchor for cross-domain identity, compromise enables complete identity takeover across all trusting domains.

Cross-domain amplification: In single-domain scenarios, the AS typically has out-of-band trust with clients. In cross-domain scenarios, the metadata endpoint IS the sole trust anchor— there's no fallback.

Metadata Poisoning Injection of false claims into legitimate metadata documents through supply chain attacks, DNS hijacking, or exploitation of hosting infrastructure.

Cross-domain amplification: Poisoned metadata propagates trust violations to all domains that fetch and cache the metadata.

8.1.1.3.2. Semantic Attacks

Policy Skew Exploitation An agent authorized under Domain A's policies operates in Domain B where different security policies apply. The agent performs actions that are within Domain A's authorized scope but violate Domain B's policies.

Claim Semantic Drift Domains interpret the same claim differently, leading to authorization decisions that neither domain intended.

Compliance Attestation Mismatch Compliance attestations (SOC2, ISO-27001, GDPR) in VCs are accepted at face value without verifying that the issuing domain's certification scope covers the receiving domain's requirements.

8.1.1.3.3. Delegation Attacks Across Domains

Cross-Domain Privilege Amplification In multi-hop delegation (A to B to C) where agents belong to different domains, downstream agents in permissive domains accumulate effective permissions that exceed what the originating principal intended.

Accountability Chain Breakage Delegation chains that cross domain boundaries lose forensic continuity. When incidents occur, attribution cannot be reconstructed because each domain only has partial visibility.

Confused Deputy Across Trust Boundaries An attacker in Domain A

tricks a trusted agent in Domain B into performing actions on their behalf that the attacker couldn't perform directly—exploiting the cross-domain trust relationship. See [Hardy1988].

8.1.1.3.4. Cross-Domain Lateral Movement

Using legitimate cross-domain authorization to pivot from a compromised position in Domain A to resources in Domain B that would not otherwise be accessible. An attacker compromises a low-privilege agent in Domain A. That agent has cross-domain authorization to Domain B. The attacker uses the agent's credentials to establish a foothold in Domain B, then escalates within Domain B.

8.1.1.3.5. Correlation and Privacy Attacks

Linking agent activities across domains using stable identifiers in VCs to build profiles of organizational operations, potentially revealing sensitive business processes or security postures.

Cross-domain amplification: In single-domain scenarios, the domain controls what's logged. In cross-domain scenarios, each domain receiving the VC can correlate independently, and the agent's home domain cannot prevent this.

8.1.1.4. Trust Establishment Between Domains

8.1.1.4.1. Domain Trust Lists

Before accepting VCs from another domain, an Authorization Server MUST verify that the issuing domain is in an explicitly configured trust list. Implementations MUST NOT implicitly trust domains based solely on successful cryptographic verification of the VC.

Trust list entries SHOULD include:

* Domain identifier (issuerDomain claim value)
* Accepted VC types from that domain
* Scope of trust (which local resources may be accessed)
* Expiration or review date for the trust relationship

8.1.1.4.2. Cross-Domain Policy Mapping

Implementations MUST NOT assume semantic equivalence of claims across domains. Before accepting a VC from another domain, the AS SHOULD verify:

1. Explicit mapping exists between the foreign domain's claim semantics and local policy requirements
2. Compliance attestations in the VC meet local regulatory requirements (not just the issuing domain's)
3. Scope values have been mapped to local authorization semantics

8.1.1.4.3. Trust Relationship Lifecycle

Cross-domain trust relationships MUST have defined lifecycles:

* Periodic review and re-validation
* Documented procedures for trust relationship termination
* Immediate revocation capability when a trusted domain is compromised
* Notification mechanisms between domains for security events

8.1.1.5. Metadata Endpoint Security for Cross-Domain Trust

Because the client-id metadata endpoint is the sole trust anchor for cross-domain agent identity, it requires enhanced protection beyond typical endpoint security.

8.1.1.5.1. Endpoint Integrity Beyond Transport

Metadata documents SHOULD be signed independently of the transport layer, allowing verification even if TLS is compromised or if the document is cached by intermediaries.

8.1.1.5.2. Change Detection for Cached Metadata

Authorization Servers SHOULD:

* Cache fetched metadata documents with appropriate TTLs
* Detect and alert on unexpected changes to critical fields (jwks_uri, vc+jwt, token_endpoint_auth_method)

* Implement anomaly detection for metadata that changes more frequently than expected

8.1.1.5.3. Cross-Domain Availability Dependencies

When metadata endpoints are unavailable, cross-domain authorization fails. Implementations SHOULD:

* Cache validated metadata with TTLs appropriate to the sensitivity of protected resources

* Implement circuit breakers to prevent cascading failures

* Define graceful degradation policies

8.1.1.6. Delegation Security Across Domain Boundaries

8.1.1.6.1. Domain Boundary Checkpoints

When a delegation chain crosses a domain boundary, the receiving domain’s AS MUST:

* Fully verify the upstream delegation chain before accepting

* Apply local policy to the delegation

* Log the domain boundary crossing for audit purposes

Receiving domains SHOULD consider the status of delegated credentials and, where status information is available, avoid making authorization decisions that rely on revoked,suspended, or suspended delegation chain elements, according to local policy.

Deployments concerned with stale delgations MAY apply bounded caching and periodic revalidation of delegation status appropriate to the sensitivity of protected resources.

8.1.1.6.2. Monotonic Scope Narrowing Across Domains

Implementations MUST enforce that effective scope can only decrease (or remain the same) at each delegation step, including when crossing domain boundaries. The receiving domain MUST apply its local policy as an additional constraint on effective scope.

8.1.1.6.3. Accountability Chain Preservation

Delegation chains spanning domains MUST maintain cryptographically verifiable accountability. Each domain SHOULD:

* Preserve the incoming delegation chain in its audit logs

* Add its own delegation record when extending the chain

* Provide APIs for authorized forensic queries from upstream domains

8.1.1.7. Privacy Considerations for Cross-Domain Sharing

8.1.1.7.1. Selective Disclosure for Cross-Domain Presentations

Agents presenting VCs across domain boundaries SHOULD use selective disclosure mechanisms (SD-JWT VC, BBS+ signatures) to reveal only claims necessary for the specific cross-domain authorization request.

8.1.1.7.2. Correlation Risk Assessment

Organizations SHOULD assess cross-domain correlation risks before establishing trust relationships, including whether agent identifiers enable activity tracking by foreign domains.

9. IANA Considerations

This document introduces a new OAuth client metadata parameter "vc+jwt" as described in the Solution Analysis section (Approach 4). If this approach is standardized, the following registration would be required:

* *Registry:* IANA [‘OAuth Dynamic Client Registration Metadata‘](https://www.iana.org/assignments/oauth-parameters/oauth-parameters.xhtml#client-metadata) registry

* *Client Metadata Name:* vc+jwt

* *Client Metadata Description:* JWT-encoded Verifiable Credential

* *Change Controller:* TBD

* *Reference:* https://www.w3.org/TR/vc-data-model-2.0

This informational document does not make a formal request for IANA registration at this time. Registration would be appropriate if this approach is adopted and progresses to a standards track specification.

10. References

10.1. Informative References

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

[I-D.abbey-scim-agent-extension]
Abbey, J., "SCIM Extensions for Agents", 2025.

[I-D.wahl-scim-agent-schema]
Wahl, M., "SCIM Schema for Agents", 2025.

[I-D.ietf-oauth-client-id-metadata-document]
Jones, M., Sakimura, N., and J. Bradley, "OAuth 2.0 Client Identity Metadata Document", 2025.

[W3C.VCDM2.0]
W3C, "Verifiable Credentials Data Model v2.0", 2024, <https://www.w3.org/TR/vc-data-model-2.0/>.

[RFC6819] Lodderstedt, T., Ed., McGloin, M., and P. Hunt, "OAuth 2.0 Threat Model and Security Considerations", January 2013, <https://www.rfc-editor.org/info/rfc6819>.

[RFC8693] Jones, M., Nadalin, A., Campbell, B., Bradley, J., and C. Mortimore, "OAuth 2.0 Token Exchange", January 2020, <https://www.rfc-editor.org/info/rfc8693>.

[Hardy1988]
Hardy, N., "The Confused Deputy: (or why capabilities might have been invented)", ACM SIGOPS Operating Systems Review Volume 22, Issue 4, 1988.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", RFC 8174, 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

Authors' Addresses

Jean Diaconu
Cisco Systems, Inc.
Av. des Uttins 5
CH-1180 ROLLE
Switzerland
Email: jdiaconu@cisco.com

Marcelo Yannuzzi
Cisco Systems, Inc.
Av. des Uttins 5
CH-1180 ROLLE
Switzerland
Email: mayannuz@cisco.com

Herve Muyal
Cisco Systems, Inc.
Av. des Uttins 5
CH-1180 ROLLE
Switzerland
Email: hmuyal@cisco.com

Frank Brockners
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
40549 DUESSELDORF
Germany
Email: fbrockne@cisco.com

Nik Kale
Cisco Systems, Inc.
170 W Tasman Dr
SAN JOSE, CA, 95134
United States of America
Email: nikkal@cisco.com

Ankit Agarwal
Skyfire, Inc.
Kentfield, CA,
United States of America
Email: ankit@skyfire.xyz

Jeffrey Hickman
Ory Corp, Inc.
15169 N Scottsdale Rd Suite 205
Scottsdale, AZ, 85254
United States of America
Email: jeff.hickman@ory.com

Amritha Lal
Amazon Web Services
Seattle, WA,
United States of America
Email: amrithak@amazon.com