

SCIM Agentic Schema

Draft Consolidation Progress
IETF 125, March 2026

History

- SCIM was been widely adopted, particularly by identity providers and applications to enable those systems to exchange data about **users** and **groups**. SCIM schemas also defined for resources such as **devices**.
- People also want to use SCIM to exchange data and access of AI **agents** and AI adjacent artifacts (e.g., MCP servers)
- Two different drafts were proposed at Montreal (IETF 124) to represent agents as a resource type in SCIM
 - <https://datatracker.ietf.org/doc/draft-abbey-scim-agent-extension/>
 - <https://datatracker.ietf.org/doc/draft-wahl-scim-agent-schema/>

Goals

- **Standardized endpoint(s) that**
 - Can represent broad conceptions of an “agent”
 - Can integrate into a greater workload landscape
 - Facilitates a rapidly growing ecosystem
 - Can support all phases of agent lifecycle across many implementations and at massive scale
- **Agent-specific Metadata & Schema that**
 - Is compatible with authentication methods agents might use including OAuth client registration specifications
 - Is extensible to support other authentication & management specifications
 - Enables identity providers for agents to provide agent identities to systems that need them
- **Non-goal: OAuth client self-registration**
 - Specifications like CIMD and DCR handle this use case for OAuth

Context / Use Case

- Agent “fleet” management/synchronization
 - Central Authorities and/or Distributed synchronized authorities
- Agent to Cloud Resource
 - Agent queries for entitlements available to it, learning what it can do
- Agentic Protocol Discoverability
 - How does an agent know what other agents are available and what interfaces they have?
 - Eg: A2A protocol, SPIFFE SVID
- Cross-Protocol Abstraction Layer
 - OAuth, MCP, A2A, SPIFFE

Base Agent Lifecycle - similar to User Lifecycle?

Similar to SCIM user resource management:

- Removing an agent from use
- Listing active agents
- Creating an agent via an approved governance workflow that cannot be completed via a real-time dynamic provisioning process

Beyond the User Lifecycle model:

- See agents that are pending approval or known by an Enterprise but not active (or disallowed by policy)

SPIFFE Provisioning

- This is the real test of “extensible to support other auth specs”
- Two AuthN/Z Models for Agent Identity
 - OAuth covers agents authorized via tokens;
 - SPIFFE covers agents identified by their runtime context
- Use Cases
 - Agent B's identity provisioned into Agent A's trust domain; discovery is a consequence of provisioning
 - Bulk-register 200 agents with attestation selectors (k8s labels, cloud metadata, image hashes)
 - Update selectors fleet-wide: same identities, new attestation criteria
 - One SCIM-integrated IdP pipeline for users, OAuth clients, and SPIFFE entries

Next steps

We will bring a draft to the mailing list

Additional Discussion

Public agent identity and private workload identities

Did we discuss the concept of "public agent identity" --1:N--> "ephemeral private workload identities" recently?

Imagine there is a giant public facing agent that needs to interact with resource in many different systems and security boundaries, "Maudlin."

The providers of "Maudlin" do not want to expose details of their internal workloads to the outside world, but do want to allow enterprise customers to recognize "Maudlin" as an agent identity in other systems they are integrating with "Maudlin."

Were we resolved that SCIM fits for the "Public Agent Identity" but not for the related "ephemeral private workload identities"? Or do we want to not take a position and have the interface be okay if the providers of "Maudlin" do want to leak the details of the "ephemeral private workload identities."

SPIFFE Provisioning

- SCIM provisions the right to an identity, not the identity itself; for SPIFFE, the credential is obtained separately. This is the real test of “extensible to support other auth specs”
- Two Agent Identity Models
 - OAuth covers agents authorized via tokens; SPIFFE covers agents identified by their runtime context
 - SPIFFE: agent runs as a workload; SPIFFE implementation issues a credential only after the workload proves it matches a registered identity entry
- Trust Domain = Discovery Boundary
 - Federation is bilateral and human-authorized; federated domains share agent identities via SCIM, non-federated see nothing
 - One boundary, one decision
- One Protocol, No Sprawl
 - SCIM drives both OAuth client registration and SPIFFE registration entries through the same IdP operators already use for user provisioning
- Use Cases
 - Agent B's identity provisioned into Agent A's trust domain therefore discovery is a consequence of provisioning
 - Bulk-register 200 agents with attestation selectors (k8s labels, cloud metadata, image hashes)
 - Update selectors fleet-wide on infrastructure change i.e same identities, new attestation criteria
 - One SCIM-integrated IdP pipeline for users, OAuth clients, and SPIFFE entries