Workload Identity in Multi System Environments
Internet-Draft
Intended status: Informational
Expires: 1 September 2026

Y. Ni
C. P. Liu
Huawei

28 February 2026

WIMSE Applicability for AI Agents
draft-ni-wimse-ai-agent-identity-02

Abstract

This document discusses WIMSE applicability to Agentic AI, so as to establish independent identities and credential management mechanisms for AI agents.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ni-wimse-ai-agent-identity/.

Discussion of this document takes place on the Workload Identity in Multi System Environments Working Group mailing list (mailto:wimse@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/wimse/. Subscribe at https://www.ietf.org/mailman/listinfo/wimse/.

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 1 September 2026.

Table of Contents

1. Introduction

AI agents are autonomous software entities that receive an intent, process contextual information, and execute decisions at machine speed with minimal human intervention. Without appropriate guardrails, they may give rise to significant risks:

* Blurred Network Boundaries: AI agents may operate across systems and platforms, which expands attack surface and amplifies security risks.

* Arbitrary and Unpredictable Access Patterns: AI agents may perform unexpected actions or access sensitive resources susceptible to malicious manipulation or logical errors.

* Lack of Accountability: Tracing an AI agent's actions is inherently difficult, leading to difficulty to detect erroneous behaviors.

* Context Rot: A gradual degradation of their ability to maintain relevant and coherent call contexts over time.

Therefore, for AI agents, the traditional perimeter-based security model has to transform into the identity-based security model, which is a prequisite to implementing precise access control and ensuring security visibility.

To realize this goal, a mechanism should be designed considering the following requirements:

* Independent, Trustworthy Identities: AI agents should have independent and trustworthy identities and credentials, distinct from those of devices and users. This allows the AI agent to act either on its own behalf or as a delegation of a user, have its own access tokens and workflows.

* Automated Credential Management: An automated mechanism is necessary for managing credentials with reduced validity period to minimize security exposure.

* Minimal Privileged Access Tokens: AI agents should have task-oriented, fine-grained access tokens with short validity periods.

* Explicit Workflows: AI agents need explicit workflow management in order to avoid random agentic access. The workflow could be long-termed and static, or could be short-termed and task-triggered, but the call context must always be visible and preserved.

This document discusses possibility of using WIMSE architecture to provide AI agent identities and credentials. It accords with the original WIMSE use case in Section 3.3.1 Bootstrapping Workload Identifiers and Credentials of [I-D.ietf-wimse-arch-06]. We also discuss requirements of extending WIMSE architecture to bind workload/AI agent identity to its owner's identity.

2. Conventions and Definitions

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses terms and concepts defined by WIMSE architecture. For a complete glossary please refer to [I-D.ietf-wimse-arch-06]:

* Trust Domain: A logical grouping of systems that share a common set of security controls and policies. Agent credentials are issued under the authority of a trust domain.

* AI Agent: The autonomous software entity that initiates the credential request. This document may refer to it as the "agent", but is is essentially the workload instead of the agent in the WIMSE architecture.

* Identity Server: A trusted entity issuing agent identity and credentials. For simplicity, this document may refer to this component as the "server".

* Identity Proxy: An intermediary component that can request, inspect, replace or augment agent identity credentials. It exposes an Agent API locally to agents. For simplicity, this document may refer to this component as the "proxy".

In addition, this document introduces the following new terms: * Owner: An entity (individual or organization) responsible for the an agent, which can provide a cryptographic signature to bind the agent identity to a specific principal. Logically, an owner may manifest in various forms, including a natural person (e.g., via a manual confirmation process), a physical device (e.g., a hardware security module), or an automated policy engine that grants approval based on pre-defined security policies. * Dual-Identity Credential: A credential that contains the identifiers and associated public keys of both an agent and its owner. The credential is cryptographically bound to both entities.

3. Architecture

3.1. Bootstrapping AI Agent Identity and Credentials

This document presumes that the identity server has already been issued a signing certificate which has set keyCertSign in the key usage extension. The server and the proxy are assumed to have established a secure channel. A basic workflow is shown in Figure 1.

1. As an intermediary between the server and the agents, the proxy provides an agent API that agents can use to initiate identity credential requests. These requests include a public key and a signature as proof-of-possession to demonstrate control of the corresponding private key.

2. The proxy forwards these requests, along with the attestation evidence for verifing the operational status of the agent, to the server for processing.

3. The server validates the evidence received from the proxy, and issues the corresponding identity credentials.

4. Once issued, the proxy forwards the agent identity credentials.

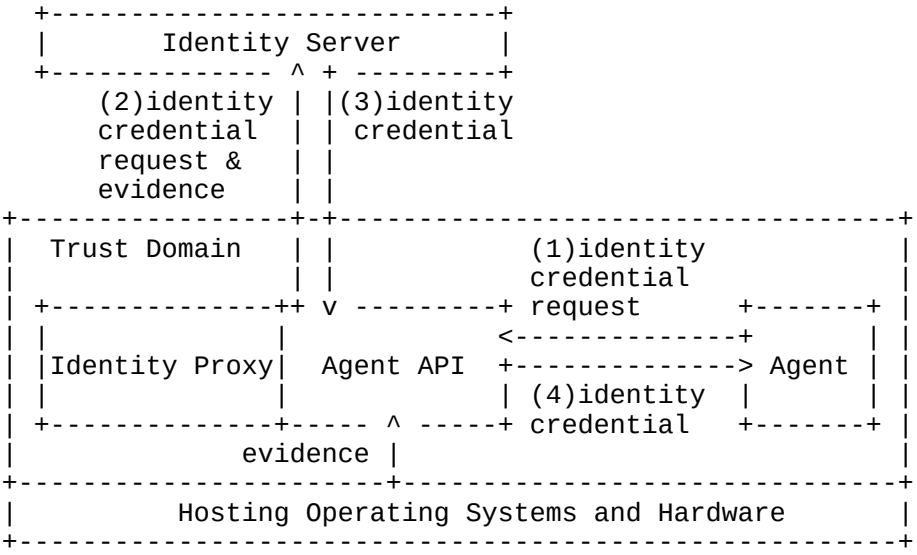

Figure 1: Basic Architecture and the Workflow

3.2. Attestation

During the request and issuance of identity credentials, the proxy should gather attestation evidence from the operating system and hardware to verify the operational status of the agent. This information is used by a RATS Verifier (could be the server) to support identity server's decision of whether or not to issue the identity credential of an agent, whether it is a bootstrapping or a renewal request. The structure and claims of this evidence may refer to the Entity Attestation Token (EAT) profile for Autonomous AI Agents [I-D.draft-messous-eat-ai-01], which defines specialized claims for AI agent integrity, training provenance, and runtime authorization.

4. Identity Binding Extensions for WIMSE

The basic WIMSE architecture ensures a trusted entity owns a trusted workload identity that is secure to connect to. However, the agent often operates on behalf of a human user or an organization. Consequently, an agent requires a credential that cryptographically binds its identity to its owner' s identity. This dual-identity credential provides a necessary foundation for access control and accountability. This section describes the necessity of dual-identity credential through two representative use cases and defines three operational models to issue it.

4.1. Use Cases

Network Access Control in Campus

Campus administrators require authentication of agents before granting access to internal network via authentication protocols such as IEEE 802.1X (e.g., using EAP-TLS). In this scenario, the Authentication Server must verify the agent's credential and determine its network privileges. A single-identity credential only allows the Authentication Server to identify the agent itself. Without further information, Authentication Server may treat the agent as a guest, providing only limited public access or rejecting the request to protect sensitive zones. A dual-identity credential allows the Authentication Server to verify both the agent's identity and the specific user or department it represents. For instance, an agent representing Alice from the R&D department can be identified as a trusted entity, and can receive proper access privileges according to existing IAM information (like RBAC). This enables the network to implement user-specific segmentation, such as automatically assigning the agent to the R&D VLAN rather than a guest network.

4.1.1. Cross-organization Interaction

In collaborative enterprise environments, it is essential to ensure that any agent requesting services is explicitly approved by its organization. This requirement spans the entire credential lifecycle, from issuance to interaction.

* Issuance: When an agent requests an identity credential, the identity server may require organizational oversight. By binding the agent's identity credential request to its corresponding organization, the server can verify the organizational approval before issuing the credential. In other words, dual-identity credential could be a manifestation of Human-in-the-Loop (HITL) mechanisms.

* Interaction: When an agent accesses another agent or a service across organizational boundaries, authentication is necessary to ensure the request is from a valid entity, as illustrated in the A2A protocol[A2A-SPEC] and WIMSE architecture[I-D.ietf-wimse-arch-06]. A dual-identity credential carries an organizational approval, which provides a strong basis for trust, ensuring both accountability and traceability for cross-organization interactions.

4.2. Issuance Models

Identity binding can be integrated into the WIMSE workflow in several ways. we introduce the following three models according to the mediation point where the agent's identity and organizational authority are cryptographically bound. Before initiating the dual-identity issuance flow, a pre-established trust relationship must exist, where the identity server is provisioned with trust anchors (e.g., public keys, CA certificates, or hardware-backed credentials) to verify the owner' s signature. The mechanism by which these trust anchors are established, distributed, or updated is out of scope of this document.

4.2.1. Agent-Mediated (Owner-Pre-Signed)

In this model, the owner acts as a local offline approvar, which provides a signature on the agent's request before it is submitted to the proxy. The identity binding phase, consisting of the following two steps, is added prior to the standard issuance flow defined in Figure 1:

a. The agent generates an identity credential request and sends it to the owner.

b. The owner signs the request and returns the signature to the agent. The agent then combines the original request and the owner's signature into a new composite request.

The following steps are similar to the basic architecture, that is, the agent send the new composite request to the server via the proxy (steps 1 and 2), then the server verifies the request and issues the dual-identity credential to the agent via the proxy (steps 3 and 4).

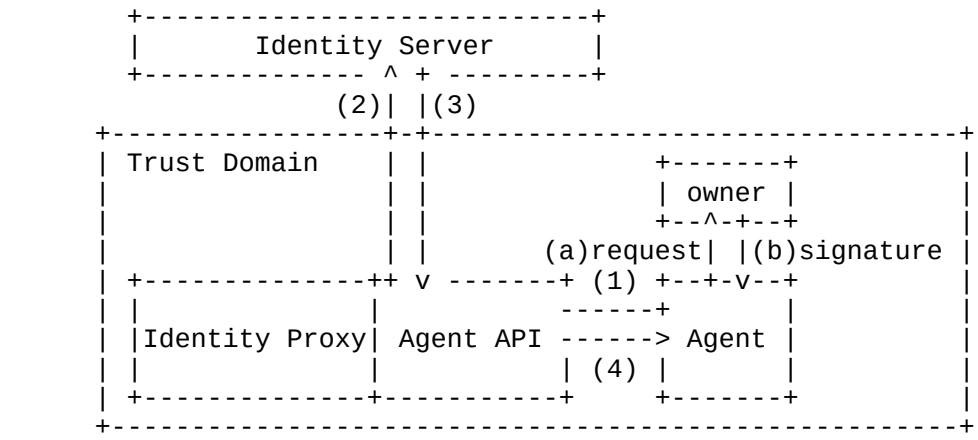

Figure 2: Agent-Mediate Model

* Typical Application Scenarios: This model is ideal for local identity binding where the owner and agent operate on the same host . It supports asynchronous and offline owner confirmation, allowing the owner use local signing module (e.g., a hardware security key) to pre-sign the agent's request. For instance, a developer using a personal hardware security key to sign an agent's request directly on their local machine. Therefore, this model obviously supports hardware-based identity binding, cryptographically anchoring the agent' s dual-identity to the specific physical device managed by the owner.

* Attack Surface: The primary attack surface lies at the local environment, focusing on two main risks. First is human-agent trust exploitation: an attacked or rogue agent may manipulate human users to approve malicious request. Second is the compromise of signing keys: if the owner' s local private keys are not protected by hardware (e.g., TPM), an attacker can forge confirmations.

4.2.2. Owner-Mediated (Gateway Mode)

In this model, the owner acts as the supervisory gatekeeper between the proxy and the server. It inspects requests relayed by the proxy to ensure compliance with organizational policies, providing cryptographic binding only after approval.

Such a mechanism is intergrated in the basic architecture as shown in Figure 3. Firstly, the agent generates an identity credential request and sends it to the proxy(step 1), then：

a. The proxy intercepts the request and relays it to the owner for administrative inspection.

b. The owner reviews and signs the request. It then combines the original request and the owner’s signature into a new composite request to be submitted to the server, along with additional organizational materials, such as an oragnizational credential.

c. The server validates the received information and issues the dual-identity credential back to the owner, who then dispatches it to the proxy.

Finally, the proxy send the credential to the corresponding agents(step 4).

Figure 3 shows a one-to-one mapping case between the owner and the proxy. In this case, the function of owner can be integrated directly into the proxy, collapsing the hierarchy into a single entity to simplify the deployment. Moreover, this model also supports a one-to-many topology, allowing a central owner to manage multiple proxies across various trust domains.

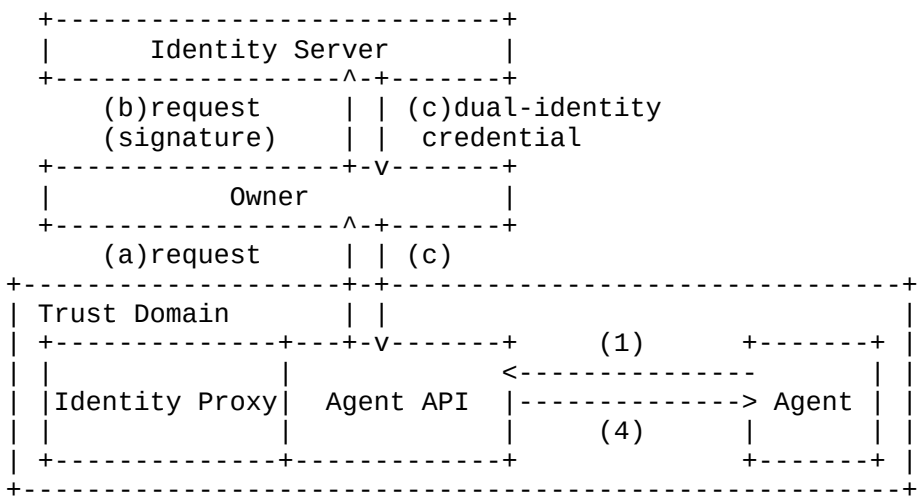

Figure 3: Owner-Mediate Model

* Typical Application Scenarios: This model is ideal for enterprise governance. Since the owner sits in the middle, acting like a middlebox to ensure no request reaches server unless it complies with enterprise security policies and compliance requirements. It is particularly suitable for hierarchical environments where the owner acts as a centralized gateway for multiple proxies.

* Attack Surface: The owner becomes a high-value target and a single point of failure. If it is compromised, an attacker can forge approvals for any agent across the managed proxies. To mitigate this, mutual authentication and cryptographic integrity are mandatory between proxies and the owner. Furthermore, as a centralized gateway, the owner is vulnerable to Denial-of-Service attacks. It is essential to implement rate limiting and request queuing.

4.2.3. Server-Mediated (Challenge-Response)

In this model, the owner acts as an independent verifier. The server orchestrates the binding phase by contacting the owner as a separate step in the issuance logic, decoupling the binding from the agent's request.

This mechanism is integrated into the basic architecture as shown in Figure 4. First, the agent sends an identity credential request to the server via the proxy (steps 1 and 2), then:

a. The server pauses the process and sends a binding challenge to the owner on whose behalf the requesting agent acts. This initiates the owner confirmation flow.

b. The owner reviews the challenge, signs it, and returns the response to the server.

After that, the server validates the owner's response, completes the identity binding, and issues the dual-identity credential to the agent via the proxy (steps 3 and 4).

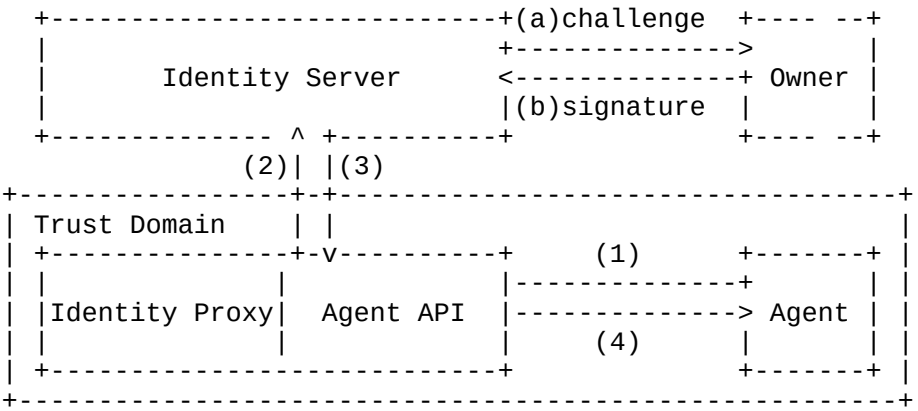

Figure 4: Server-Mediate Model

* Typical Application Scenarios: This model is suitable for scenarios requiring independent and real-time confirmation from an owner who is not involved in the initial request path. For instance, An agent is deployed by a service provider on behalf of an organization. When the agent requests an identity credential, the server initiates an out-of-band verification directly to the administrative center.

* Attack Surface: The primary risk lies in the out-of-band channel. Without nonces or mutual authentication, an attacker could intercept this channel to perform response forgery or replay attacks.

5. Comparison with CHEQ

While both this document and CHEQ [I-D.draft-rosenberg-cheq-00] introduce a human element to enhance security, their goals and the underlying mechanisms are different.

CHEQ focuses primarily on controlling the actions of AI agents. It requires user double confirmation when an AI Agent invokes an OAuth access token request, preventing possible deviation from user expectations.

The purpose of this document is to provide distinct identity and credentials to AI agents, whether or not it is bound to an owner user of device's parent identity. Whether or not the agent inherits access permission privileges from its user is out of scope of this document.

6. Initial Trust Establishment

AI agents may operate in cloud or campus. In the cloud, the initial trust establishment between the proxy and the server has already been solved by solutions like SPIRE. However, in campus scenarios, the heterogeneity and limited manageability of devices make credential provisioning challenging, complicating initial trust establishment.

BRSKI [RFC8995] provides a feasible method by introducing a cryptographically signed artifact called "voucher".

In the BRSKI flow, the proxy (acting as a BRSKI pledge) discovers the server (acting as a BRSKI registrar), initiates a TLS handshake, and sends a voucher request including its immutable manufacturer credential-the IDevID (Initial Device Identifier). The server uses this IDevID to contact the manufacturer's service (MASA). After validating the request, the MASA issues a signed voucher.

The proxy then verifies the manufacturer's signature on the voucher, which securely transferring trust from the manufacturer to the local domain. This verified trust is a prerequisite for the server to issue a local domain device certificate (LDevID). This certificate enrollment step essentially follows the standard EST mechanism [RFC7030].

However, it should be noted that BRSKI is not necessarily the only way to achieve this secure integration. The core goal is to bridge the initial trust gap. If the proxy is pre-configured with the target server's public key or certificate and can securely locate it, the standard EST protocol alone may be sufficient to establish trust and obtain the LDevID certificate.

Open Question: What are the precise conditions and mechanisms for determining the use of various bootstrap methods (including but not limited to BRSKI and EST)?

7. Security Considerations

TODO Security

8. IANA Considerations

This document has no IANA actions.

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/rfc/rfc2119>.

[RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <https://www.rfc-editor.org/rfc/rfc7030>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <https://www.rfc-editor.org/rfc/rfc8995>.

9.2. Informative References

[A2A-SPEC] "Agent2Agent (A2A) Protocol Specifications", October 2025, <https://a2a-protocol.org/latest/topics/enterprise-ready/>.

[I-D.draft-messous-eat-ai-01] MESSOUS, A., Morand, L., and P. C. Liu, "Entity Attestation Token (EAT) Profile for Autonomous AI Agents", Work in Progress, Internet-Draft, draft-messous-eat-ai-01, 23 February 2026, <https://datatracker.ietf.org/doc/html/draft-messous-eat-ai-01>.

[I-D.draft-rosenberg-cheq-00] Rosenberg, J., White, P., and C. F. Jennings, "CHEQ: A Protocol for Confirmation AI Agent Decisions with Human in the Loop (HITL)", Work in Progress, Internet-Draft, draft-

rosenberg-cheq-00, 24 July 2025,
<https://datatracker.ietf.org/doc/html/draft-rosenberg-cheq-00>.

[I-D.ietf-wimse-arch-06]
Salowey, J. A., Rosomakho, Y., and H. Tschofenig, "Workload Identity in a Multi System Environment (WIMSE) Architecture", Work in Progress, Internet-Draft, draft-ietf-wimse-arch-06, 30 September 2025,
<https://datatracker.ietf.org/doc/html/draft-ietf-wimse-arch-06>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Yuan Ni
Huawei
Email: niyuan1@huawei.com

Chunchi Peter Liu
Huawei
Email: liuchunchi@huawei.com