Workload Identity in Multi System Environments
Internet-Draft
Intended status: Informational
Expires: 23 April 2026

Y. Ni
C. P. Liu
Huawei

20 October 2025

WIMSE Applicability for AI Agents
draft-ni-wimse-ai-agent-identity-01

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 23 April 2026.

Copyright Notice

Copyright (c) 2025 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 2
2. Conventions and Definitions 3
3. Architecture 4
3.1. Bootstrapping AI Agent Identity and Credentials 4
3.2. Attestation 5
4. Extensions to the WIMSE Architecture-- Binding Workload/AI Agent Identity to Its User Identity 5
5. Comparison with CHEQ 7
6. Initial Trust Establishment 8
7. Security Considerations 8
8. IANA Considerations 8
9. References 8
9.1. Normative References 8
9.2. Informative References 9
Acknowledgments 9
Authors' Addresses 9

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:

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

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

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

4. 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:

1. 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.

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

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

4. 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 user 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].

1. 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.

2. 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.

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

4. 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".

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.

2. The proxy forwards these requests, along with the 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.

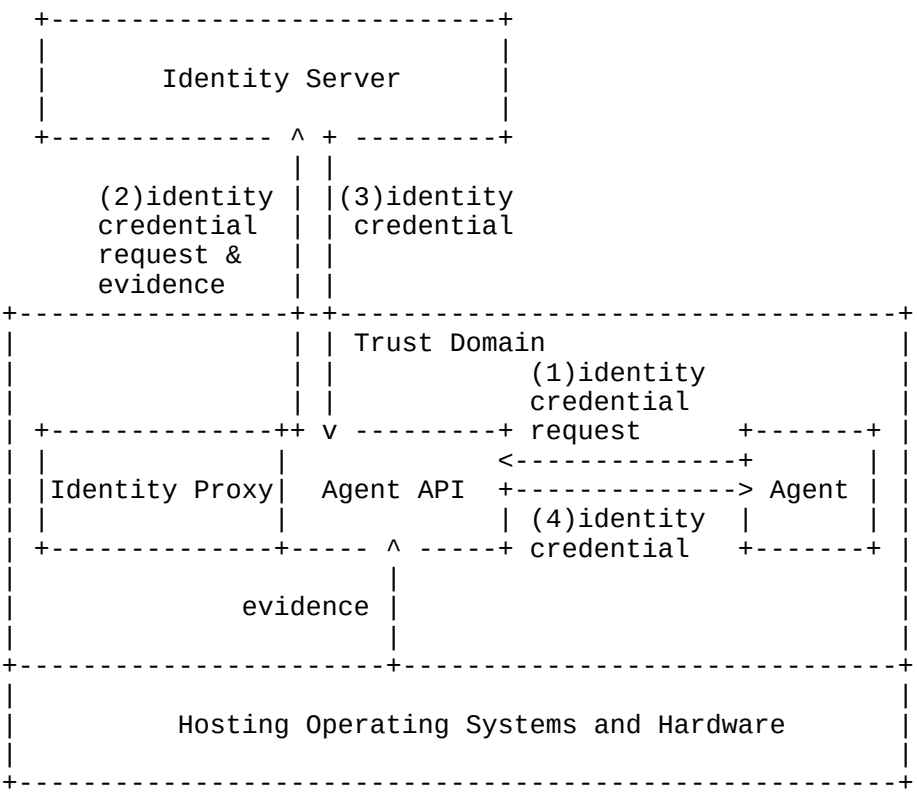

Figure 1: Basic Architecture and the Workflow

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 decide whether or not to issue the identity credential of an agent, whether it is a bootstrapping or a renewal request.

4. Extensions to the WIMSE Architecture-- Binding Workload/AI Agent Identity to Its User Identity

AI Agent identity has the full complexity of user identity, since agents may act on behalf of human, organization, etc. Therefore, agent should have a credential that both denotes its identity and its human owner identity, which will provide convenience for future access controls. Cryptographic assurances must be provided that the user approves the credential request.

Figure 2 illustrates the extended architecture, which binds user identity to agent identity. This architecture extends the basic workflow described in Section 2.2.

The core process remains largely unchanged from steps 1 to 4. However, a critical enhancement is introduced between steps 2 and 3:

4.1. Upon receiving an identity credential request, the server forwards it to the user on whose behalf the requesting agent acts. This initiates the user confirmation flow. 4.2. The user validates the received information. Upon approval, the user should provide a cryptographic signature, binding the user’s identity to the requested agent credential.

Open Question: How can users effectively provide cryptographic signatures for agent credential requests? Is leveraging hardware security features in user devices a viable and practical approach?

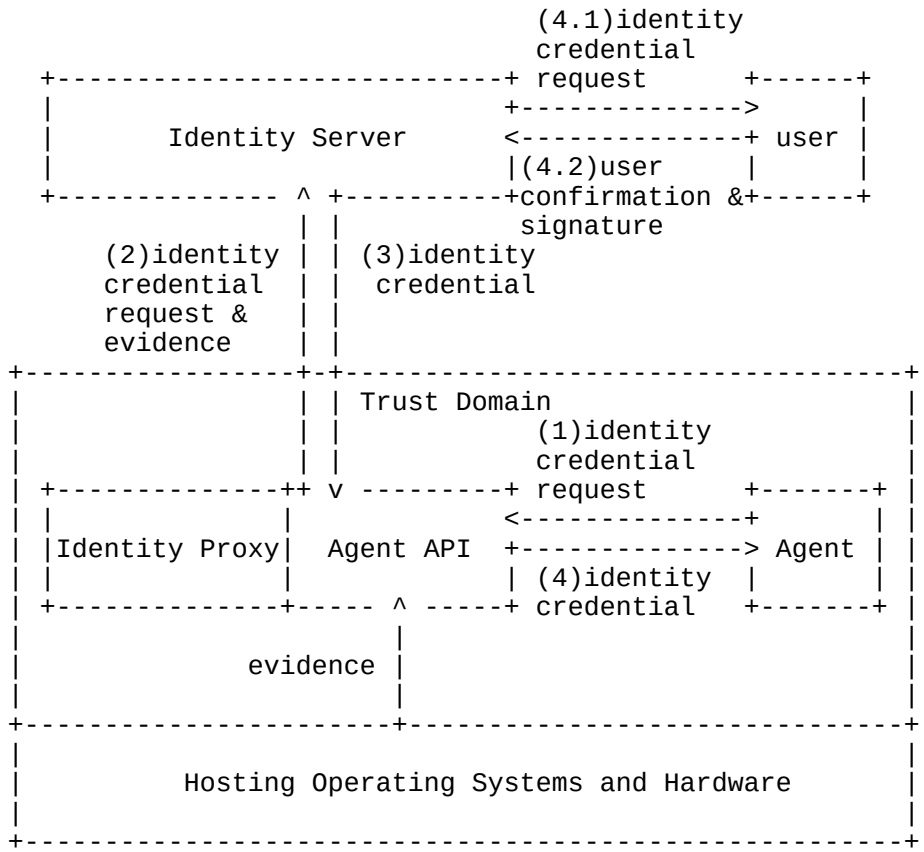

Figure 2: Extended Architecture and the Workflow

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

[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>.

[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>.

[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>.

[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>.

9.2. Informative References

[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>.

[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>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Yuan Ni
Huawei
Email: niyuan1@huawei.com

Chunchi Peter Liu
Huawei
Email: liuchunchi@huawei.com