agent2agent
Internet-Draft
Intended status: Informational
Expires: 6 May 2026

Y. Ni
C. P. Liu
N. Geng
Q. Gao
Z. R. Li
Huawei

2 November 2025

Security Requirements for AI Agents
draft-ni-a2a-ai-agent-security-requirements-00

Abstract

This document discusses security requirements for AI agents, covering different stages of security interactions. These include provisioning, registration, cross-domain interconnection, and access control.

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 6 May 2026.

Copyright Notice

Table of Contents

1. Introduction

With the widespread application of agentic AI technology across various business scenarios, its security issues have become increasingly prominent.

This document aims to provide an architecture addressing security requirements across different stages of interactions of Agentic AI use cases. These includes provisioning, registration, cross-domain interconnection, and access control. This document establishes a starting point to guide Agentic AI security design, development, and implementation consideration discussions.

2. Architecture

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.

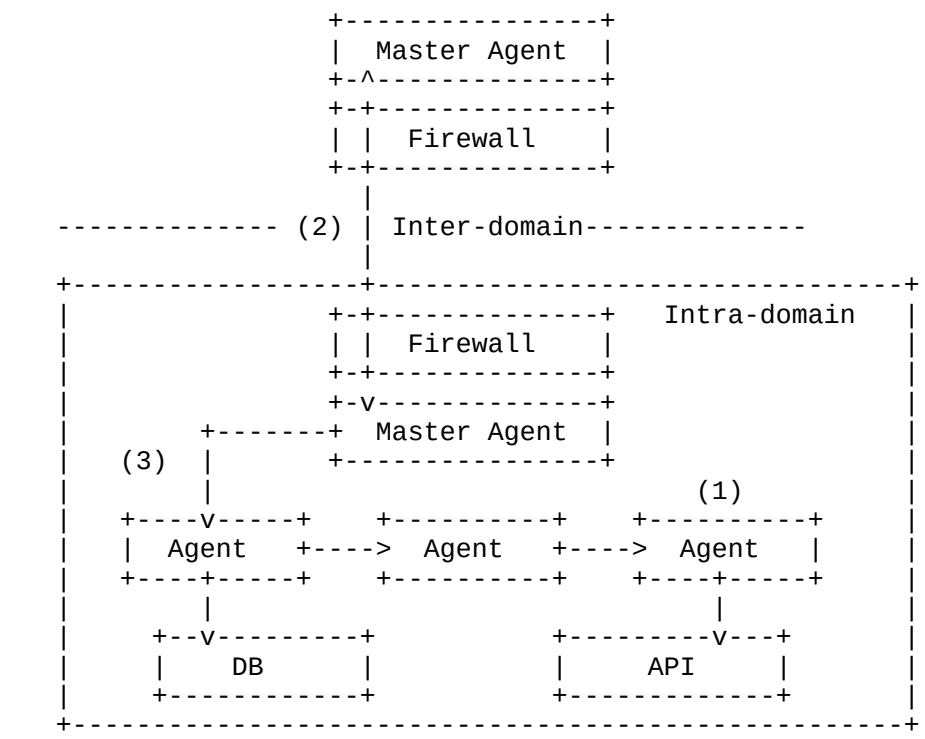

Figure 1. Architecture of Agent Security Control and Management

The architecture of agent security control and management is illustrated in Figure 1. There are four types of security interactions, in a sequential order:

1. Provisioning and Registration: Creating agent identity, establishing initial trust, provisioning agent secrets and credentials.

2. Cross-domain Interconnection: Enabling secure, authenticated communication between agents across different trust domains.

3. Access Control: The Master Agent validates both intra-domain and inter-domain access tokens, creates internal workflow and manages different credentials for heterogeneous systems.

Therefore, the architecture includes four components:

1. Firewall: A network security device designed to monitor, filter, and control incoming and outgoing network traffic based on predetermined security rules.

2. Master Agent: The central orchestrating entity that manages multi-agent operations, including cross-domain communication, workflow coordination, credential management, and security policy management.

3. Agents: Autonomous software entities deployed in various domains to perform specific tasks.

4. Heterogeneous systems: API endpoints, microservices, tools, and databases.

3. Provisioning and Registration

Figure 2 shows the diagram of provisioning and registration, which includes Agent Certificate Authority(ACA) and Agent Registry Service(ARS):

1. ACA (Agent Certificate Authority): A trusted third party that issues and manages credentials for agents.

2. ARS (Agent Registry Service): A system responsible for agent identity registration and discovery-matching.

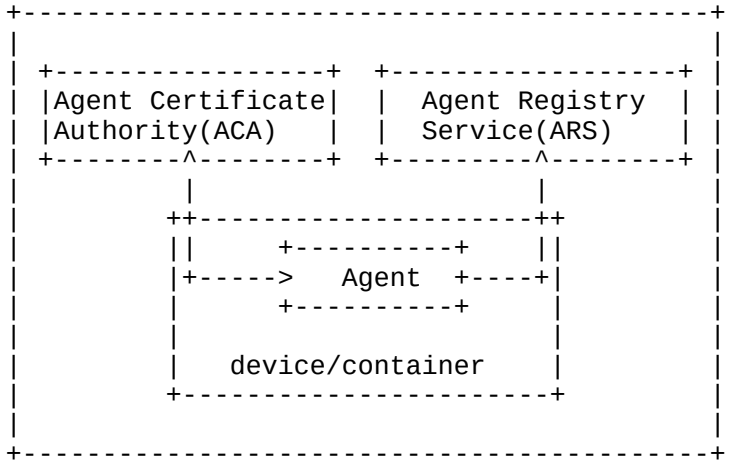

Figure 2. Diagram for Provisioning and Registration

3.1. Identity Provisioning and Management

Identity provisioning and management are the process of creating and assigning a verifiable digital identity to an agent.

* Initial Trust Establishment: Intial trust can be established through one or more of the following trust anchors, including, but are not limited to: a manufacturer-embedded immutable credential like an IDevID certificate; a hardware root of trust like a Trusted Platform Module (TPM) or Hardware Security Module (HSM); identity documents like an AWS Instance Identity Document or an Azure Managed Service Identity token. This step verifies the agent's execution environment (device, container, etc.) as trustworthy, allows the device or container to join the network, thereby enabling secure operations for all subsequent steps.

* Credential Request: During a credential request, the agent must provide multiple proofs of its legitimacy, such as a Certificate Signing Request or a Proof of Possession by signing with the corresponding private key, as well as remote attestation by collecting and submitting evidence to a RATS (Remote Attestation Procedures) Verifier. Additionally, to define the agent's operational scope, the request should incorporate user identity context, binding the credential to a specific human user or an organizational role.

* Credential Issuarance: The ACA validates proofs and requests from the above two steps, if passed, it issues an agent-specific credential that may include its owner or requester identity, capabilities, locator, acceptable validation methods for the ARS.

* Credential Lifecycle Mangement: The ACA not only issues credentials but also defines and enforces revocation policies. These policies are triggered by specific events, such as a detected security compromise, the agent’s scheduled decommissioning, or a key rotation.

3.2. Agent Registration

After receiving a credential from the ACA, the agent then sends it to the ARS to authenticate itself and start the registration process.

* Authentication: The ARS must verify the legitimacy of the credential submitted by the agent. It must be signed or otherwise endorsed by the ACA.

* Registration: The ARS then checks if the information signed by the ACA, such as the agent’s capabilities, exactly matches the registration request sent by the agent. Upon successful validation, the ARS assigns the agent a unique identifier and establishes an agent record that links the identifier to its attributes.

* Record Management: This step automatically removes expired credentials and synchronizes with the ACA to ensure timely revocation of credentials, preventing the use of invalid or compromised credentials.

3.3. Agent Onboarding

Agent onboarding differs between campus and cloud environments. On campus, agents use protocols like EAP-TLS for network access. In the cloud, the process involves injected sidecars, which register agents to the central service mesh registry automatically to enable communication and management.

4. Cross-Domain Interconnection

4.1. Cross-Domain Identifier Interoperability

Different domains may use distinct identifier schemas. Possible methods include:

* pre-configured schema translation

* cross-domain identifier synchronization

* a universal parsing framework or system

4.2. Secure Cross-Domain Transmission

Mutual TLS (mTLS) connection starts from the external requesting agent to the master agent. The master agent terminates the mTLS connection and parses the application layer requests. In this case, the master agent functions as an OAuth resource server, and manages internal task orchestration.

4.3. Authenticating External Calls

The master agent then verifies the identity of the requesting agent, and whether or not it has permission to the requested service or agent. Different authentication methods might be possible:

* API keys

* Username-password

* Pre-shared secrets

* Assertions (for example, JWT Authorization Grant[I-D.draft-ietf-oauth-identity-chaining-06])

which can even be combined with AND/OR logic. During this process, the master agent might be able to identify the caller endpoint type:

* human user via browser or app

* human user via API

* AI agents

* Hardware or equipment via an IoT API

4.4. IAM Integration

Since the agent may inherit its access rights from its owner or user, when authenticating requests, the validation might require integration of IAM systems for redirected verification.

5. Access Control

5.1. Authorization Handling

The master agent acts as the role of OAuth 2.1 resource server. It must validate access tokens as described in OAuth 2.1 Section 5.2. If validation fails, it must respond according to OAuth 2.1 Section 5.3 error handling requirements.

5.2. Authorization Chaining Across Domains

In an agentic AI use case, a request may traverse multiple resource servers in multiple trust domains before completing. It will be common that the requesting agent from domain A needs to access the resource server (master agent) of domain B. During this process, the following information should be preserved:

* Original requesting agent identity
* Authorization context
  - Scope
  - Resource
  - Audience
  - Grant type
  - Assertion

The current best practice is [I-D.draft-ietf-oauth-identity-chaining-06].

5.3. Converting to Internal Workflow

* Workflow Generation: Complex tasks often require multi-agent collaboration. The master agent receives, parses, and extracts the original job request from the external requesting agent, then create sequential workflows or parallel calls. This requires the master agent to have information of all callable internal API assets, agent capabilities, etc.
* Downscoping: If the master agent intends to use a workflow, it extracts the original caller’s identity and authorization context, and initiates a new internal workflow. It should follow the current least privilege best practice of downscoping-Transaction Tokens as specified in [I-D.draft-tulshibagwale-oauth-transaction-tokens-05]. The access rights to each downstream workload decrease.

5.4. Interoperability for Heterogeneous Systems

Within a domain, there might exist different types of heterogeneous systems or legacy systems that require different authentication methods. They could be API endpoints, microservices, tools or databases. The exact authentication methods are determined by the service itself, for example,

* bearer tokens
* API keys
* pre-shared secrets
* username-passwords
* X.509 certificates, etc.

As a result, the master agent also works as an intermediary credential manager that converts the formats, scopes, identity of the credential, bridging the gap between heterogeneous systems and platforms.

Examples include:

* Static secrets (API keys) to be exchanged to short-lived, on demand credentials (bearer tokens)

5.5. Zero Trust Analysis

The above information can be used as rich context that allow zero trust access control. Remote attestation results of the requesting agent could also be part of access policy decision point’s inputs. Remote attestation results of the requesting agent could include the following information:

* RoT and trust anchors
* Identifiers
* Affiliations
* Posture assessment results
* Capabilities

The overall information will be used as input of Policy Engine (PE) and Policy Decision Point (PDP).

5.6. Microsegmentation

Microsegmentation may be enforced to prevent lateral movement of security risks. Possible granularity of microsegmentation includes:

* per IP segment/subnet

* per each workload

* per tags and attributes (of workload), etc.

There should be policy enforcement points (PEP) at the perimeter of each segment. Each PEP can receive software-defined security policies issued by PE/PDP.

6. IANA Considerations

This document has no IANA actions.

7. References

7.1. 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>.

[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>.

7.2. Informative References

[I-D.draft-ietf-oauth-identity-chaining-06] Schwenkschuster, A., Kasselman, P., Burgin, K., Jenkins, M. J., and B. Campbell, "OAuth Identity and Authorization Chaining Across Domains", Work in Progress, Internet-Draft, draft-ietf-oauth-identity-chaining-06, 12 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-identity-chaining-06>.

[I-D.draft-tulshibagwale-oauth-transaction-tokens-05] Tulshibagwale, A., Fletcher, G., and P. Kasselman, "Transaction Tokens", Work in Progress, Internet-Draft, draft-tulshibagwale-oauth-transaction-tokens-05, 20 October 2023, <https://datatracker.ietf.org/doc/html/draft-tulshibagwale-oauth-transaction-tokens-05>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Yuan Ni
Huawei
Email: niyuan1@huawei.com

Chunchi Peter Liu
Huawei
Email: liuchunchi@huawei.com

Nan Geng
Huawei
Email: gengnan@huawei.com

Qiangzhou Gao
Huawei
Email: gaoqiangzhou@huawei.com

Zhenbin Robin Li
Huawei
Email: robinli314@163.com