Network Working Group
Internet-Draft
Intended status: Informational
Expires: 16 September 2026

D. King
Lancaster University
R. Ramdhany
BBC
C. Liu
Huawei Technologies

15 March 2026

A Trust and Authentication Framework for Cross-Domain Agent-to-Agent Communications

draft-kiliram-agent-trust-auth-framework-00

Abstract

AI-based agent-to-agent communication increasingly occurs across trust domains (e.g., between enterprises, service providers, SaaS platforms, and application third parties). While many agent protocols and platforms can provide transport security and local permission models, deployments lack a coherent, interoperable baseline for verifiable agent identity, credentialing, cross-domain authorisation, delegation, revocation, and auditability.

This document defines an architectural framework for a cross-domain trust substrate for AI-based agent ecosystems. The framework is intended to be agent protocol-agnostic and to provide a consistent trust baseline that existing and emerging AI agent protocols can build upon.

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 16 September 2026.

Table of Contents

1. Introduction

AI agent ecosystems are moving from single-agent-service-operator deployments to multi-party environments, where independently operated AI agents interact across organisational and administrative boundaries. In these environments, trust is partial and dynamic, and is constrained by policy, contracts, and regulation.

Existing AI agent protocols and platforms often provide transport security and local permission models. They do not, by themselves, provide a consistent cross-domain baseline for verifiable agent identity, accountable delegation, interoperable credential handling, and audit evidence. Interoperability is therefore frequently implemented through bespoke integrations with uneven security properties and unclear lifecycle semantics.

This document composes established IETF security and identity building blocks into a cross-domain trust substrate for AI agent communications. It also identifies remaining interoperability gaps.

This document defines a Trust and Authentication Framework and key requirements for Agent-to-Agent Communications across domain boundaries. It specifies an architectural model, threat model, and a set of requirements for interoperable trust establishment, including:

trust domains and trust anchors; agent gateways and their security roles; identity and credential models; authentication and authorisation patterns (including delegation); and operational requirements for policy enforcement, lifecycle management, and auditability.

1.1. Scope

This framework targets deployments where AI-based agents interact across trust domains, including cloud providers and other agent service operators hosting agent runtimes, agent gateways, and third-party AI agent services. In these environments, the agent-to-agent transport protocol alone does not provide a sufficient basis for verifiable agent identity, cross-domain authorisation, delegation, and audit. In this document, a trust domain is a policy and governance scope, not simply the set of agents that implement the same communication protocol.

This document does not define a new agent communication protocol. It also does not standardise a global naming or discovery mechanism; instead, it defines the trust requirements that any discovery or registry interface must satisfy in order to bind endpoints and capabilities to verifiable identities and trust anchors across domains. Non-AI software agents are out of scope.

1.2. Background

1.2.1. Agentic Systems Context

This document focuses on agentic systems in which software agents can plan, invoke tools or services, and execute multi-step tasks across organisational boundaries. These interactions require interoperable trust, identity, delegation, and authorisation controls that remain consistent across trust domains.

1.2.2. Trust Domain Definition

In this document, a Trust Domain is a bounded policy and governance scope in which agents, gateways, identities, and credentials are governed under common administrative control or federated trust agreements.

A Trust Domain can align with network boundaries, application/content boundaries, protocol ecosystem boundaries, or combinations of these.

Membership in the same trust domain does not imply universal trust between agents; authentication and authorisation are evaluated per interaction according to policy. Therefore, "cross-domain" includes interactions across any of these trust-domain scopes.

2. Conventions and Terminology

Even though this document is not a protocol specification, it makes use of upper case key words to define requirements unambiguously.

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

Agent: An AI-based software entity that can initiate actions, invoke other services, and exchange messages to accomplish tasks.

Agent Service Operator: The entity responsible for operating an AI agent runtime, agent gateway, or related control-plane service. This is distinct from a network infrastructure operator.

Domain Boundary: A logical boundary between two domains such that membership, credentials, and policy assertions from one side are not assumed to be valid on the other side without explicit trust establishment and enforcement. Domain boundaries often coincide with agent gateways or mediation components.

Agent Gateway: A policy-enforcing intermediary that mediates agent-to-agent communications across trust domains.

Calling Agent: The initiating agent in a cross-domain interaction.

Gateway Coordinator: Gateway-side logic that performs policy enforcement, token exchange, and context propagation for cross-domain interactions.

Resource Agent: The target agent or service in the destination trust domain.

Subject Token: A token presented by the calling side as input for token exchange.

Exchanged Access Token: A domain-scoped, policy-constrained token issued for use in the destination trust domain (i.e., the issued token in OAuth 2.0 Token Exchange [RFC8693]).

Transaction Token: A token that carries workflow and task-intent context across delegated or multi-step interactions.

Relying Party: An entity that evaluates credentials, policy, and evidence before accepting agent requests.

Trust Domain: A bounded policy and governance scope that may align with network boundaries, application/content boundaries, protocol ecosystem boundaries, or combinations of these. This document uses the term in the same OAuth federation sense as token exchange between independently governed domains (see [RFC8693] and [I-D.ietf-oauth-identity-chaining]).

Trust Domain Authority: The policy authority within a trust domain that defines trust anchors, identity acceptance criteria, and cross-domain policy constraints.

3. Problem Statement

Early AI agent deployments are often single-domain: one agent service operator controls runtime, identity lifecycle, and policy enforcement. In that setting, proprietary identity, authorisation, and audit mechanisms may be operationally sufficient.

Cross-domain deployments are different. They span independent agent service operators (for example, enterprises, cloud providers, SaaS platforms, and third-party AI agent services) with different issuers, policy semantics, and lifecycle processes. In these settings, the transport protocol alone is not sufficient for interoperable trust establishment, delegation control, lifecycle handling, and auditability.

Three recurring deployment cases illustrate the problem space:

* Same protocol, different providers: an enterprise AI agent hosted by one cloud provider or agent service operator invokes a partner or supplier AI agent hosted by a different agent service operator using the same agent-to-agent protocol. Despite protocol interoperability, the parties still require a consistent approach to verifying agent identity and agent service operator accountability, evaluating credentials from different issuers, applying policy constraints (e.g., least privilege and step-up requirements), and handling credential lifecycle events across domains.

* Different protocols across domains: an agent service operator uses one protocol internally while an external AI agent ecosystem uses another. An agent gateway (or protocol bridge) mediates between

protocols, but trust and authorisation decisions must remain coherent end-to-end. In particular, identity and claims need stable representation across the boundary, delegation semantics must survive translation, and resulting actions must remain attributable and auditable.

* Multi-operator media agent pipelines: in Object-Based Media (OBM) deployments, AI agents dynamically assemble discrete media objects (video segments, audio layers, and associated metadata) across independently operated infrastructure, compute, and content delivery services. Agents acting on behalf of different service operators must authenticate their task invocations and obtain authorisation to access or manipulate media objects, with delegation semantics and audit evidence preserved across operator boundaries. This use case is described in [I-D.rrk-object-based-media-usecase].

These cases require a protocol-agnostic trust substrate that can be applied consistently across providers, gateways, and heterogeneous protocol ecosystems. A primary interoperability challenge is preserving identity, delegation semantics, and policy intent when requests are translated between different agent communication protocols.

Cross-domain AI agent deployments face the following recurring gaps:

* Verifiable identity: recipients need to authenticate an agent and bind that identity to an accountable owner/agent service operator.

* Because agents frequently act on delegated authority, recipients need a verifiable delegation chain and a way to constrain scope and purpose.

* Heterogeneous systems — including gateways and intermediaries — require consistent policy enforcement across trust domains.

* Credential lifecycle: rotation and revocation must be supported with clear operational semantics.

* Actions and delegation chains must be auditable, with tamper-evident evidence for regulated environments.

4. Threat Model

This document considers (non-exhaustive):

* Impersonation of agents or agent providers.

* Compromise of agent execution environment or keys.

* Confused-deputy and privilege escalation via delegation chains.

* Replay, token substitution, and context injection.

* Cross-domain policy bypass at intermediaries.

* Downgrade of trust signals and audit/evidence manipulation.

5. Design Goals and Non-Goals

Goals:

* A protocol-agnostic trust baseline for cross-domain agent communications (CDAC).

* Interoperable identity binding (agent <-> owner/agent service operator <-> keys).

* Independent agent identities with support for on-behalf-of delegation to distinguish agent and principal.

* Short-lived, rapidly-rotated credentials that avoid long-term static secrets.

* Authorisation with constrained scope/purpose, fine-grained access tokens, and task-triggered issuance based on least-privilege principles.

* Delegation chain preservation and verification to prevent confused-deputy and privilege escalation attacks.

* Standardised lifecycle semantics (issue/rotate/revoke/suspend) with automated, zero-touch rotation.

* Capability registration, advertisement, and parsing mechanisms that reduce the risk of malicious discovery behaviour, spoofed announcements, or unsafe consumption of registry data.

* Evidence and audit hooks suitable for transparency services, regulatory compliance, and operational troubleshooting.

Non-Goals (initially):

* Defining a new agent communication protocol.

* Standardising underlying secure transport or message protection mechanisms. These are expected to be provided by existing transports (for example, HTTPS/TLS or IPsec) and, where needed, application-layer message protection mechanisms such as JOSE/COSE [RFC7515] [RFC9052]. This framework builds on top of such mechanisms rather than redefining them.

* Standardising a global naming or discovery system (the framework defines trust requirements for discovery, but not discovery protocols themselves).

* Defining runtime safety, model alignment mechanisms, or content filtering (these are application-layer concerns outside the scope of cross-domain trust establishment).

6. Architecture

This section describes the reference architecture used in this document: the entities in each trust domain, the cross-domain mediation points, and the control points for authentication, authorisation, credential lifecycle, and evidence generation. The architecture is protocol-agnostic and can be applied to multiple agent messaging protocols and deployment models.

Figure 1 shows the high-level components and trust relationships.

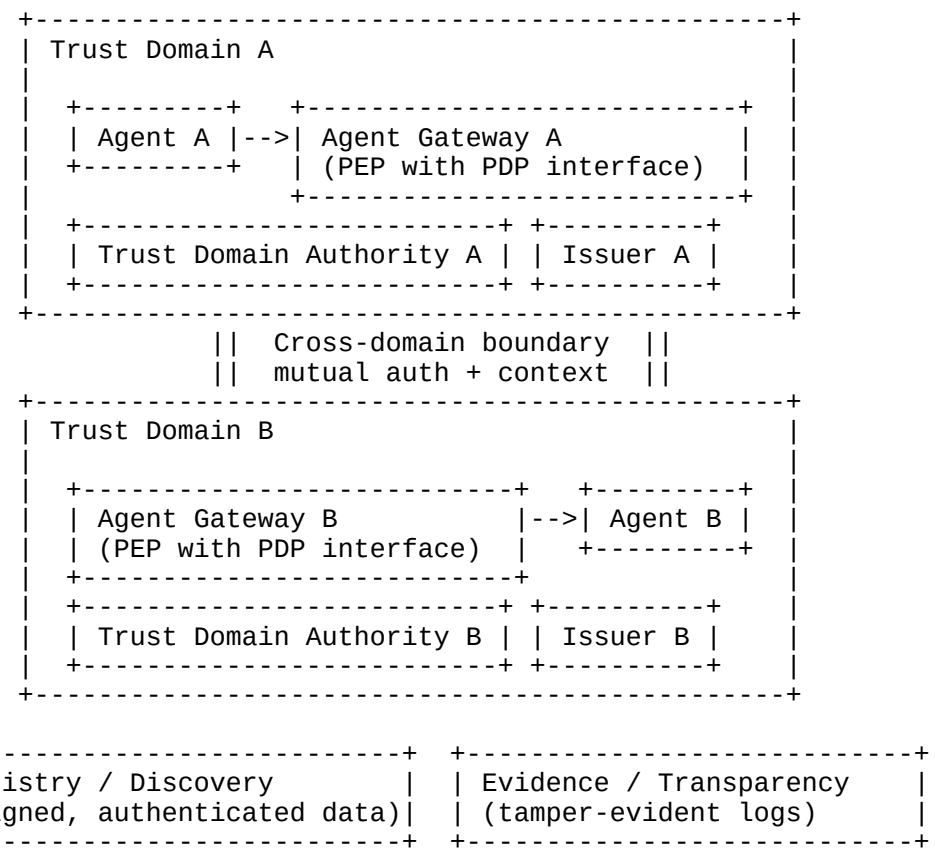

Figure 1: Reference Cross-Domain Trust Architecture

6.1. Entities and Roles

* Agent (caller, callee)
* Agent Provider / Operator
* Trust Domain Authority (policy + trust anchors)
* Agent Gateway (cross-domain mediation)
* Credential Issuer (PKI, token issuer, or equivalent)
* Transparency / Evidence Service (optional)

6.2. Trust Domains and Trust Anchors

A trust domain MUST define:

* One or more trust anchors used to validate agent credentials.
* Policy governing acceptable credential types and assurance levels.
* Operational requirements for issuance, rotation, and revocation.
* Audit and evidence requirements for cross-domain interactions.

6.3. Agent Gateways

Gateways SHOULD support:

* Mutual authentication on both sides of the gateway.
* Verification of delegation chain and policy constraints.
* Policy enforcement (PEP) with a decision interface (PDP).
* Evidence emission (logs/receipts) for audited actions.

6.4. Registries and Discovery Interfaces

Discovery mechanisms (DNS, directories, registries, APIs) enable agents to locate and interact with other agents or services. In deployments where discovery information influences trust or routing decisions, discovery and registration mechanisms MUST support verifiable bindings between discovered endpoints and associated identities.

Registration Security:

* Capability and endpoint registrations MUST be authenticated. Only agents or operators with valid credentials and appropriate authorisation SHOULD be able to register or update entries in a discovery registry.
* Registrations SHOULD be signed or otherwise cryptographically bound to the registering agent’s identity to enable verification by consumers.
* Registries SHOULD implement rate limiting, abuse detection, and validation checks to prevent flooding, enumeration, or injection of malicious entries.

Broadcast and Advertisement Security:

* Where capability advertisements are broadcast (e.g., via mDNS, DNS-SD, or multicast protocols), recipients MUST verify the authenticity and authorisation of the broadcaster before trusting advertised endpoints or capabilities.

* Broadcast mechanisms SHOULD include anti-spoofing protections (e.g., cryptographic signatures, nonce-based freshness, or network-level source authentication).

Parsing and Consumption Security:

* Agents consuming discovery information MUST implement robust parsing and validation to prevent exploitation via malformed or malicious registry entries (e.g., buffer overflows, injection attacks, or resource exhaustion).

* Discovered endpoints and capabilities SHOULD be treated as untrusted until verified through authentication and authorisation (see Section 8).

Identity and Trust Anchor Binding:

* Discovery mechanisms SHOULD enable consumers to bind discovered endpoints or capabilities to verifiable agent identities and relevant trust anchors. For example, a registry entry MAY include or reference the agent’s credential, public key fingerprint, or trust domain identifier.

* This document defines requirements for such bindings, but does not standardise specific discovery protocols or registry formats.

6.5. Reference Flow Labels and Token Transitions

This subsection introduces flow labels and token terms used across Sections 6, 8, 9, and 10.

* Calling Agent: the initiating agent in the source trust domain.

* Gateway Coordinator: gateway-side logic that performs policy enforcement, token handling, and cross-domain mediation.

* Resource Agent: the target agent or service in the destination trust domain.

* Subject Token: token presented by the calling side as input to token exchange.

* Exchanged Access Token: policy-constrained token issued for the destination trust domain (i.e., the issued token in OAuth 2.0 Token Exchange [RFC8693]).

* Transaction Token: token carrying workflow and task-intent context across delegated steps.

Figure 2 shows the reference sequence and control points.

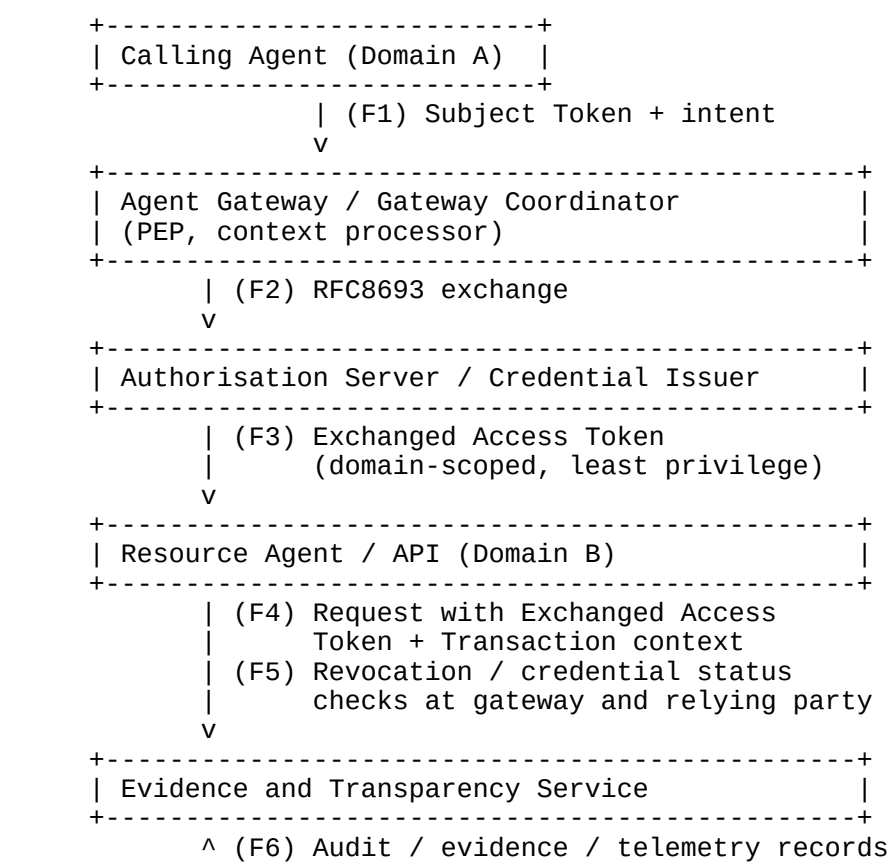

Figure 2: Reference Flow Labels for Context-Aware Cross-Domain Access

7. Identity and Credential Model

Cross-domain agent interactions require a consistent approach to identity representation, credential binding, and cryptographic proof. This section defines requirements for agent identifiers, credential formats, key management practices, and lifecycle operations that support short-lived credentials and avoid long-term static secrets.

7.1. Agent Identifiers

An agent identifier MUST unambiguously identify an agent instance within and across trust domains. Agent identifiers SHOULD:

* Be globally unique or unique within a well-defined namespace.

* Be stable across credential rotation events (i.e., the agent identity persists even as cryptographic keys are rotated).

* Support binding to an accountable agent service operator or owner entity (e.g., an organisation, cloud provider, or user principal).

Agent identifiers MAY be represented using URIs, DIDs (Decentralised Identifiers), NAI-style identifiers [RFC7542], or domain-scoped naming schemes, provided that the chosen format supports verifiable binding to credentials and trust anchors.

Use of NAI-style identifiers is optional and deployment-specific; this document does not require or prefer NAI over other identifier formats.

In delegation scenarios (see Section 7.4), it MUST be possible to distinguish the agent’s own identity from the identity of the principal (user or service) on whose behalf it is acting.

Identity Schemes and Namespace Considerations:

Agent identity is fundamentally tied to well-defined namespaces and trust contexts. Different deployment scenarios employ different identity management approaches:

* Enterprise and Federated Identity: In enterprise consortia and federated environments, agent identity is often managed through Single Sign-On (SSO), federation, and related identity-provider infrastructure. Provisioning and synchronisation mechanisms such as SCIM [RFC7643] [RFC7644] MAY be used operationally, but do not by themselves define runtime trust or authorisation semantics across domains.

* Consumer and Citizen Identity: In consumer or citizen-facing scenarios, deployments MAY use verifiable credential ecosystems and related work building on the W3C Verifiable Credentials Data Model [W3C.VC]. The exact role of such credential formats, including CBOR- or JSON-based representations, is deployment-specific.

* Workload and Application Identity: For application and workload identities (including AI agents), emerging work in areas such as workload identity and credential management may be relevant to representing both the agent's identity and the identity of its owner or principal (see Section 7.4).

These examples are illustrative rather than exhaustive, and the relationship between these identity ecosystems and cross-domain agent trust is to be discussed in more detail in a later version of this document.

Deployments SHOULD select identity representation formats appropriate to their operational context and trust model. Universal identity schemes face practical challenges in namespace governance, trust anchor distribution, and cross-context interoperability; agent service operators SHOULD design for namespace-scoped identity with explicit trust relationships between domains.

7.2. Credential Formats

Agent credentials bind an agent identifier to cryptographic proof material (public keys or equivalent) and MAY include additional claims such as capabilities, agent service operator identity, assurance level, and validity constraints.

Credentials MUST support:

* Cryptographic binding between agent identity and key material, so that possession of the credential implies control of the corresponding private key.

* Tamper-evident representation (e.g., signed by a trusted issuer).

* Time-bounded validity, with explicit expiration constraints.

* Revocation or suspension mechanisms (see Section 9).

Credentials SHOULD support:

* Structured claims for authorisation context (e.g., scope, purpose, or allowed actions).

* Where the agent acts on delegated authority, representation of the full delegation chain (see Section 7.4 and Section 8.2).

Suitable credential formats include X.509 certificates, JSON Web Tokens (JWT), CBOR Web Tokens (CWT), Verifiable Credentials (W3C VC), or equivalent formats that meet the requirements above. The choice of format SHOULD align with the operational and performance constraints of the deployment environment.

7.3. Key Management and Proof-of-Possession

Agent credentials MUST be bound to cryptographic keys under the control of the agent or its runtime environment. To mitigate key compromise and support operational agility, deployments SHOULD:

* Use short-lived credentials with validity periods measured in hours or days, rather than months or years.

* Rotate keys and credentials frequently and automatically.

* Avoid embedding long-term static secrets in agent code, configuration, or environment variables.

Agent authentication MUST include proof-of-possession of the private key corresponding to the public key in the credential. Acceptable proof-of-possession mechanisms include:

* Digital signatures over authentication challenges or request content (e.g., using JOSE or COSE signature schemes).

* Key agreement or key confirmation protocols.

* Cryptographic binding in token-based schemes (e.g., DPoP for OAuth 2.0 [RFC9449]).

Where hardware security modules (HSMs), trusted execution environments (TEEs), or secure enclaves are available, private keys SHOULD be protected within these environments to reduce exposure.

7.4. On-Behalf-Of Delegation

Agents frequently act on behalf of a user, service, or organisation. In such cases, credentials and authorisation tokens MUST clearly represent both:

* The agent’s own identity (the immediate actor).

* The identity of the principal on whose behalf the agent is acting (the authorising party).

This separation enables relying parties to:

* Attribute actions to the correct principal for audit and accountability purposes.

* Apply policy based on both agent and principal identity (e.g., "agent X acting for user Y may access resource Z").

* Enforce constraints on delegation depth and scope (see Section 8.2).

Delegation relationships SHOULD be cryptographically verifiable, either through structured token claims (e.g., JWT "act" or "may_act" claims) or through chained credentials that establish a delegation path from the principal to the agent.

Dual Identity for AI Agents:

One possible model for AI agents is a dual-identity representation comprising the agent’s own identity and the identity of its owner or invoking principal [I-D.ni-wimse-ai-agent-identity]. Where such a model is used, it may provide additional context for access control decisions. Potential benefits include:

* Reuse of existing Role-Based Access Control (RBAC) policies defined for human principals, while also applying agent-specific constraints.

* Differentiation between agent capabilities (what the agent itself is authorised to do) and principal authorisation (what the owner is authorised to do), allowing security engines to make better-informed decisions.

* Clear separation of accountability: the agent’s virtual identity remains stable across different principals, while the principal’s identity varies based on who invoked the agent.

Where deployments implement such a dual-identity model, both identities SHOULD be represented in credentials, tokens, or equivalent policy inputs as appropriate. The detailed representation and processing model is to be discussed in more detail in a later version of this document.

7.5. Credential Lifecycle and Rotation

To support operational resilience and security hygiene, agent credential issuance and rotation MUST be automated and SHOULD NOT require manual intervention under normal operating conditions.

Deployments SHOULD implement:

* Automated credential issuance upon agent initialisation or registration.

* Proactive credential renewal before expiration (e.g., renewing when 50-75% of the validity period has elapsed).

* Graceful handling of overlapping validity periods to avoid service disruption during rotation.

Credential rotation events SHOULD be logged for audit and troubleshooting. Revocation and suspension mechanisms are described in Section 9.

8. Authentication and Authorisation Patterns

This section defines authentication and authorisation requirements for cross-domain agent interactions. The patterns described here are intended to support least-privilege access control, time-bound authorisation, task-specific scoping, and auditability across trust domain boundaries.

8.1. Mutual Authentication

Agent-to-agent interactions across domain boundaries MUST use mutual authentication, in which both the calling agent and the receiving agent (or gateway) authenticate each other before exchanging sensitive data or performing actions.

Mutual authentication ensures that:

* The caller can verify the identity and trustworthiness of the service or agent it is invoking.

* The callee can verify the identity and authorisation of the caller before granting access or performing requested actions.

Mutual authentication SHOULD be established at the transport layer (for example, mutual TLS) or, where appropriate, at the application layer using signed requests, proof-of-possession mechanisms, or equivalent message protection schemes (for example, HTTP Message Signatures [RFC9421]).

Where agent gateways mediate cross-domain communication, mutual authentication MUST be performed on both sides of the gateway: between the calling agent and the gateway, and between the gateway and the target agent or service.

8.2. Delegation and Call-Chain Context

When an agent acts on behalf of a user or invokes another agent as part of a multi-step workflow, the call chain context MUST be preserved and made available to downstream relying parties.

In Figure 2 terms, call-chain context enters at (F1), is evaluated and transformed during (F2), and is propagated downstream at (F4) (typically via transaction tokens or equivalent structured context).

Call-chain context includes:

* The identity of the original requesting principal (user or service).

* The identity of each intermediary agent in the delegation chain.

* Constraints on delegation depth (e.g., "may delegate at most one level further").

* Scope and purpose restrictions that apply to delegated authority.

Relying parties MUST be able to inspect and verify the call-chain context in order to:

* Attribute actions to the correct principal for audit and compliance purposes.

* Detect and prevent confused-deputy attacks and privilege escalation.

* Enforce policy based on the full delegation path (e.g., "user A via agent B via agent C may not access resource X").

Call-chain context SHOULD be represented using structured tokens (e.g., JWT with nested or chained claims) or as a sequence of verifiable credentials that establish the delegation path. Cryptographic signatures MUST bind each step in the chain to prevent tampering or substitution.

Transaction Tokens [I-D.ietf-oauth-transaction-tokens] may provide one useful mechanism for encoding and protecting workflow-related context across trust domains in multi-step interactions. Depending on deployment requirements, such tokens may carry task, transaction, or purpose-related context that can assist downstream policy evaluation. The exact representation of delegation chains, workflow state, and related semantics is deployment-specific and to be discussed in more detail in a later version of this document.

8.3. Policy Enforcement and Decision Points

Access control decisions in cross-domain agent ecosystems require dynamic evaluation of policy based on agent identity, principal identity, call-chain context, resource attributes, and environmental conditions.

Deployments SHOULD implement a clear separation between:

* Policy Enforcement Points (PEPs): components that intercept requests and enforce access decisions (e.g., agent gateways, API gateways, or runtime enforcement layers).

* Policy Decision Points (PDPs): components that evaluate policy rules and return authorisation decisions to PEPs.

PDPs SHOULD support policy evaluation based on:

* Agent and principal identity.

* Delegation chain and call-chain depth.

* Resource being accessed and requested operation.

* Contextual attributes (e.g., time of day, source network, assurance level).

Policy languages and decision engines are not standardised by this framework; however, policies MUST be enforceable consistently across gateways and relying parties within a trust domain, and SHOULD be auditable and versioned for governance and compliance purposes.

8.4. Least-Privilege and Fine-Grained Access Tokens

Access tokens issued to agents MUST implement the principle of least privilege by constraining:

* Scope: the set of resources or operations the token grants access to (e.g., "read messages in project X", "invoke workflow Y").

* Validity period: tokens SHOULD have short lifetimes (measured in minutes to hours) to limit exposure in the event of compromise.

* Purpose or task context: tokens MAY be bound to a specific task or workflow instance, and MUST NOT be reusable across unrelated operations unless explicitly permitted by policy.

Fine-grained scoping reduces the impact of token leakage or misuse and supports auditability by creating a clear link between tokens and the tasks they authorise.

Access tokens SHOULD be issued dynamically in response to specific task triggers (e.g., user initiation of a workflow, scheduled job execution, or event-driven invocation) rather than being provisioned statically at agent deployment time.

Token formats SHOULD support structured scope representation (e.g., OAuth 2.0 scopes, RBAC roles, or attribute-based claims) and MUST include expiration timestamps. Deployments MUST support one or more revocation or status-validation mechanisms (e.g., token introspection, revocation endpoints, issuer-side deny lists, or equivalent controls) to enable lifecycle management (see Section 9).

Cross-Domain Token Exchange:

When agents interact across trust domains, access tokens issued in one domain often cannot be used directly in another domain. OAuth 2.0 Token Exchange [RFC8693] provides one established mechanism for exchanging a token from Domain A for a token acceptable in Domain B, subject to policy and trust relationships between the domains.

Related OAuth extensions may be relevant in specific cross-domain scenarios:

* Identity Chaining [I-D.ietf-oauth-identity-chaining] may be applicable where identity information must be conveyed across trust domains in a form acceptable to a downstream authorisation system.

* Identity Assertion Authorisation Grant [I-D.ietf-oauth-identity-assertion-authz-grant] may be relevant where identity assertions are used as inputs to obtain access tokens in federation scenarios.

Deployments using OAuth-based authorisation SHOULD consider these existing standards before creating bespoke token exchange or assertion mechanisms. Detailed profiling guidance for agent-specific use of these extensions is to be discussed in more detail in a later version of this document.

In the reference sequence (Figure 2), this corresponds to:

* (F1) Subject Token presented by the calling side to the gateway.

* (F2) Token exchange request by the gateway/authorisation component.

* (F3) Issuance of a destination-domain exchanged access token.

* (F4) Use of the exchanged access token (and workflow context) to access destination resources.

8.5. Explicit Workflow and Step-Up Authentication

In multi-step agent workflows, explicit workflow identifiers and step-up authentication MAY be required to ensure that sensitive actions are authorised at an appropriate assurance level.

Explicit workflow context includes:

* A workflow or session identifier that links related agent interactions.

* The current step or phase within the workflow.

* Constraints on permissible transitions between steps.

Step-up authentication is a mechanism in which an agent or user is required to re-authenticate or provide additional proof before performing a high-risk or sensitive operation. For example:

* A low-assurance token may permit read-only operations, while write or delete operations require a higher-assurance token obtained through step-up.

* Delegation to a third-party agent may require explicit user consent or re-authentication.

Step-up requirements SHOULD be expressed in policy and enforced by PDPs and PEPs. Workflow and step-up context SHOULD be included in call-chain tokens to ensure that downstream relying parties can verify compliance with workflow constraints.

9. Revocation, Rotation, and Lifecycle

Credential lifecycle management is critical to the security and operational resilience of cross-domain agent ecosystems. This section defines requirements for credential issuance, rotation, revocation, and suspension, with an emphasis on automation, short validity periods, and clear operational semantics.

Relative to Figure 2, lifecycle controls govern tokens and credentials produced in (F2)/(F3), consumed in (F4), and continuously validated via (F5).

9.1. Automated Credential Issuance

Agent credentials SHOULD be issued automatically as part of agent initialisation, registration, or onboarding processes. Manual credential provisioning introduces operational overhead, increases the risk of misconfiguration, and delays agent deployment.

Automated issuance mechanisms MUST:

* Authenticate the requesting agent or agent service operator before issuing credentials.

* Bind the issued credential to the agent’s identity and cryptographic keys (see Section 7).

* Apply policy constraints (e.g., validity period, scope, trust domain) at issuance time.

* Log issuance events for audit and compliance purposes (see Section 10).

Credential issuers MAY implement attestation or proof-of-identity requirements (e.g., requiring the agent to demonstrate control of a pre-registered public key, or to provide a bootstrapping token issued by a trust domain authority).

9.2. Credential Rotation

To limit exposure in the event of key compromise and to support cryptographic agility, agent credentials SHOULD be rotated frequently. Rotation intervals depend on the threat model and operational context, but deployments SHOULD target validity periods measured in hours or days rather than months or years.

Automated rotation MUST be supported without manual intervention. Rotation mechanisms SHOULD:

* Begin renewal before credential expiration (e.g., when 50-75% of the validity period has elapsed) to avoid service disruption.

* Support overlapping validity periods for old and new credentials during the rotation window, allowing graceful transition without breaking in-flight requests.

* Re-authenticate the agent and verify continued authorisation before issuing renewed credentials.

* Log rotation events and preserve a historical record of credential issuance for forensic analysis (see Section 10).

Rotation SHOULD be triggered by time-based expiration, but MAY also be triggered by policy events (e.g., change of agent role, detected anomaly, or administrative action).

9.3. Revocation

Revocation is the permanent invalidation of a credential before its natural expiration. Revocation is necessary in response to key compromise, agent decommissioning, policy violation, or other security events.

Revocation mechanisms MUST:

* Provide timely propagation of revocation status to relying parties. Acceptable mechanisms include Online Certificate Status Protocol (OCSP), Certificate Revocation Lists (CRLs), token revocation endpoints, or equivalent real-time revocation services.

* Support query-based revocation checking by relying parties (i.e., relying parties MUST be able to verify whether a credential is revoked before accepting it).

* Generate audit evidence when credentials are revoked, including the reason for revocation and the time of revocation.

Deployments using short-lived credentials (e.g., validity periods of hours or days) MAY rely on expiration rather than explicit revocation for routine lifecycle management, reserving revocation for emergency or high-severity events. However, a revocation mechanism MUST still be available for such events.

Revocation SHOULD be irreversible. If an agent requires new credentials after revocation, it MUST re-authenticate and obtain a fresh credential through the issuance process.

9.4. Suspension and Resumption

Suspension is the temporary invalidation of a credential, typically in response to a transient condition (e.g., suspected anomaly, pending investigation, or administrative hold). Unlike revocation, suspension MAY be reversible.

Suspension mechanisms SHOULD:

* Use the same propagation and checking mechanisms as revocation (e.g., OCSP, CRLs, or token introspection endpoints).

* Distinguish suspended credentials from revoked credentials in status responses, enabling relying parties to apply appropriate policy (e.g., temporary denial vs. permanent rejection).

* Log suspension and resumption events for audit purposes.

Resumption (re-activation of a suspended credential) SHOULD require explicit administrative action or policy evaluation, and MUST be logged. Resumed credentials retain their original expiration time; suspension does not extend validity.

9.5. Operational Semantics and Grace Periods

To avoid service disruption during lifecycle transitions, deployments SHOULD implement grace periods and operational best practices:

* Clock skew tolerance: relying parties SHOULD tolerate small clock skew (e.g., +/- 5 minutes) when validating credential expiration times.

* Caching and refresh intervals: relying parties that cache revocation or suspension status SHOULD refresh cached data frequently (e.g., every few minutes for high-assurance environments, or hourly for lower-risk contexts).

* Fallback and failover: if revocation status cannot be checked (e.g., due to network failure or service outage), deployments SHOULD implement a fail-safe policy (e.g., reject credentials when revocation status is unavailable in high-assurance environments, or accept credentials with logging and delayed verification in lower-risk environments).

Operational semantics for lifecycle events (issuance, rotation, revocation, suspension, resumption) SHOULD be documented and consistent across trust domains to ensure interoperability.

10. Auditability, Transparency, and Evidence

Cross-domain agent interactions often involve sensitive data, high-value transactions, or regulated decision-making processes. To support accountability, compliance, and operational troubleshooting, this framework requires that agent interactions produce verifiable evidence available to operators and auditors.

10.1. Audit Requirements

Agent platforms, gateways, and relying parties MUST generate audit logs for security-relevant events, including:

* Authentication and authorisation decisions (successful and failed).

* Credential issuance, renewal, revocation, and suspension events.

* Delegation and call-chain construction.

* Access to sensitive resources or execution of high-risk operations.

* Policy evaluation decisions and any policy violations or anomalies.

Audit logs MUST include sufficient context to support forensic analysis and compliance reporting, including:

* Timestamp (with time zone or UTC offset).

* Agent identity and principal identity (if applicable).

* Call-chain context (see Section 8.2).

* Resource or operation being accessed.

* Authorisation decision and policy identifier.

* Reference flow label(s) for the transaction path (e.g., F1-F6 from Figure 2), where available.

Audit logs SHOULD be tamper-evident (e.g., signed, hash-chained, or committed to an append-only transparency log) to ensure integrity and non-repudiation.

10.2. Transparency and Evidence Services

In deployments where auditability and public accountability are critical (e.g., regulated industries, cross-organisational collaborations, or high-assurance environments), operators MAY integrate transparency services such as:

* Certificate Transparency (CT) logs for agent credentials.

* Verifiable data structures (e.g., Merkle trees) for tamper-evident audit trails.

* Third-party attestation or notarisation services that provide independent verification of agent actions or credential lifecycle events.

Transparency services enable:

* Detection of mis-issued or unauthorised credentials.

* Independent audit of agent behaviour and delegation chains.

* Compliance with regulatory requirements for auditability and traceability (e.g., GDPR Article 22 for automated decision-making, or financial services audit requirements).

This framework does not mandate specific transparency log formats or protocols, but recommends that any transparency service integrated into an agent ecosystem support cryptographic proof of inclusion and consistency.

10.3. Decryption and Inspection for Compliance

In some regulatory or enterprise environments, operators or compliance teams may require the ability to decrypt and inspect agent-to-agent traffic for security monitoring, data loss prevention (DLP), or regulatory compliance purposes.

Where decryption and inspection are required, deployments SHOULD:

* Implement inspection at controlled policy enforcement points (e.g., gateways or proxies) rather than passively intercepting encrypted traffic.

* Use explicit trust relationships and key escrow or key sharing arrangements that are disclosed to all parties and governed by policy.

* Log and audit all decryption and inspection events to ensure accountability and prevent misuse.

Decryption and inspection mechanisms MUST NOT undermine the end-to-end integrity and authenticity guarantees provided by agent credentials and signed messages. In particular:

* Inspection points MUST re-encrypt traffic after inspection to maintain cryptographic protection downstream.

* If an intermediary applies a new downstream signature, it MUST preserve verifiable evidence of the original upstream sender signature or attestation context so that accountability and provenance are not lost.

* Inspection MUST be authorised by policy and MUST NOT occur without the knowledge and consent of the trust domain authorities on both sides of the interaction.

Where appropriate, operators MAY consider privacy-preserving inspection techniques, such as selective field decryption or other deployment-specific approaches. Advanced techniques and their applicability are to be discussed in more detail in a later version of this document.

11. Operational Considerations

11.1. Relationship to OAuth Ecosystem

This framework aligns with OAuth working group specifications for cross-domain authorisation and identity federation. Deployments using OAuth 2.0 or OpenID Connect SHOULD prioritise existing extensions and profiles over proprietary mechanisms.

OAuth specifications and drafts that may be relevant to this framework include:

* RFC 8693 [RFC8693] for token exchange between security domains.

* Transaction Tokens [I-D.ietf-oauth-transaction-tokens] for carrying transaction-related context in multi-step workflows.

* Identity Chaining [I-D.ietf-oauth-identity-chaining] for cross-domain identity conveyance and related assertion patterns.

* Identity Assertion Authorisation Grant [I-D.ietf-oauth-identity-assertion-authz-grant] for using identity assertions as inputs to access-token issuance.

These specifications and drafts illustrate possible building blocks for cross-domain authorisation, delegation, and workflow context handling. Their precise application to agent-to-agent communication is to be discussed in more detail in a later version of this document.

11.2. OAuth Profiling for Agent-to-Agent Communication

Deployments using OAuth 2.0 for agent communications SHOULD adopt or profile existing OAuth mechanisms where possible. For example, [I-D.liu-oauth-a2a-profile] describes one approach to profiling OAuth 2.0 for agent-to-agent communications, including reuse of transaction-related context fields for agent-specific semantics.

Such profiles provide concrete guidance on:

* Token formats and claim structures for agent credentials and access tokens.

* Grant types and flows appropriate for agent initialisation, delegation, and cross-domain invocation.

* Integration with agent gateways and policy enforcement points.

Operators SHOULD evaluate applicable profiles and contribute implementation experience to ongoing standardisation work. Additional profiling considerations are to be discussed in more detail in a later version of this document.

11.3. Deployment Patterns

Cross-domain deployments commonly follow the patterns below.

Same-Protocol, Different Providers:

Agents hosted by different agent service operators (e.g., different cloud providers or enterprises) use the same agent-to-agent protocol but operate under different trust domains. In this scenario, agent

gateways at domain boundaries handle mutual authentication, credential verification, token exchange (via RFC 8693 or equivalent), and policy enforcement, while preserving protocol compatibility end-to-end.

Different Protocols with Gateway Mediation:

Agents using different protocols (e.g., one domain uses Protocol X internally, another uses Protocol Y) communicate through a protocol-translating gateway. The gateway acts as a relying party on both sides, performing mutual authentication, protocol translation, and policy enforcement. Call-chain context and delegation semantics MUST be preserved across the gateway (see Section 8.2), and audit logs MUST capture the translation and policy decisions.

Federated Multi-Domain Workflows:

A workflow spans multiple domains, with agents in each domain performing subtasks and passing results to agents in other domains. Implementations MAY use transaction tokens or equivalent mechanisms to carry workflow-related context across domain boundaries. Trust relationships MUST be established between domains (e.g., via pre-configured trust anchors or other agreed mechanisms), and each domain MUST enforce its own policy on inbound requests. Additional details are to be discussed in more detail in a later version of this document.

11.4. Gateway Configuration and Policy

Agent gateways are critical policy enforcement points in cross-domain deployments. Operators SHOULD:

* Configure gateways to enforce mutual authentication on both sides (inbound and outbound) as described in Section 8.1.

* Integrate gateways with Policy Decision Points (PDPs) that evaluate authorisation policies based on agent identity, principal identity, call-chain context, and resource attributes (see Section 8.3).

* Enable audit logging and evidence emission for all gateway decisions (see Section 10).

* Implement rate limiting, anomaly detection, and abuse prevention mechanisms to protect against malicious or misconfigured agents.

Gateway policies SHOULD be versioned, auditable, and subject to governance review. Policy updates SHOULD be tested in staging environments before deployment to production.

11.5. Interoperability and Testing

Interoperability between agent platforms and trust domains requires consistent implementation of credential formats, token structures, policy semantics, and lifecycle operations. Operators SHOULD:

* Participate in interoperability testing programs or working group plugfests to validate cross-domain interactions.

* Publish conformance statements describing supported credential types, token formats, policy languages, and lifecycle mechanisms.

* Use standardised test vectors and example flows (see Appendix A) to verify correct implementation of delegation chains, token exchange, and revocation checking.

Interoperability issues SHOULD be reported to relevant standards bodies (for example, the IETF OAuth working group and relevant agent protocol communities) to inform future updates.

12. Privacy Considerations

Cross-domain agent interactions can expose personal data, sensitive business context, and behavioural metadata across multiple operators. Deployments SHOULD apply privacy-by-design controls to identity, delegation, telemetry, and audit processing.

In particular, deployments SHOULD:

* Apply data minimisation. Credentials, tokens, and call-chain context SHOULD carry only the claims required for a specific transaction.

* Apply purpose limitation. Identity and delegation claims SHOULD be scoped to an explicit task or workflow and SHOULD NOT be reused for unrelated processing.

* Avoid unnecessary disclosure of principal identity to downstream domains where pseudonymous or pairwise identifiers are sufficient.

* Limit retention of audit logs and telemetry to what is necessary for security, compliance, and operational purposes, consistent with applicable law and policy.

* Logs, traces, and evidence services that may contain sensitive metadata SHOULD be protected by access controls, redaction policies, and compartmentalisation.

* Provide transparency and governance for cross-domain sharing, including documented legal basis, controller/processor roles, and cross-border transfer requirements where applicable.

When decryption and inspection are used (Section 10.4), agent service operators SHOULD ensure the minimum disclosure necessary for the compliance use case and SHOULD log access to inspected content.

Deployments that process personal data MUST comply with applicable privacy and data protection requirements in their jurisdictions.

13. Security Considerations

This section summarises baseline controls and residual risks for cross-domain agent interactions. The threat model is in Section 4.

Deployments MUST:

* Enforce mutual authentication across domain boundaries (Section 8.1), including both sides of any mediating gateway.

* Validate proof-of-possession for presented credentials and tokens (Section 7.3).

* Call-chain context for delegated requests MUST be preserved and verified to mitigate confused-deputy and privilege escalation attacks (Section 8.2).

* Apply least-privilege, short-lived tokens, and explicit lifecycle controls (issuance, rotation, revocation, suspension) as described in Sections 8.4 and 9.

* Protect trust anchors, issuer keys, and agent private keys against compromise (Section 7.3), including use of hardware-backed key protection where available.

* Tamper-evident, auditable evidence MUST be generated for all security-relevant events (Section 10).

Specific security risks include:

* Compromised gateways or intermediaries, which can become high-value targets and policy bypass points.

* Incomplete revocation propagation or stale status caches, which can allow temporary acceptance of invalid credentials.

* Unauthenticated endpoint or capability bindings create a surface for discovery or registry poisoning (Section 6.4).

* Weak trust-domain onboarding or misconfigured inter-domain trust, which can permit token substitution or unauthorised token exchange.

No single control is sufficient. Deployments should combine identity, cryptographic, policy, and lifecycle controls.

An example external risk taxonomy is provided in Appendix B for additional context.

14. IANA Considerations

This document has no IANA actions.

15. References

15.1. 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/info/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/info/rfc8174>.

[RFC8693] Jones, M., Nadalin, A., Campbell, B., Ed., Bradley, J., and C. Mortimore, "OAuth 2.0 Token Exchange", RFC 8693, DOI 10.17487/RFC8693, January 2020, <https://www.rfc-editor.org/info/rfc8693>.

15.2. Informative References

[I-D.ietf-oauth-identity-assertion-authz-grant] Campbell, B., "Using OpenID Connect Identity Assertions as Authorization Grants", October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-identity-assertion-authz-grant-01>.

[I-D.ietf-oauth-identity-chaining]
Schwenkschuster, O. and P. Czapiewski, "OAuth 2.0 Identity Chaining across Trust Domains", September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-identity-chaining-08>.

[I-D.ietf-oauth-transaction-tokens]
Hardt, D. and O. Schwenkschuster, "Transaction Tokens", October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-transaction-tokens-07>.

[I-D.liu-oauth-a2a-profile]
Liu, D., "OAuth 2.0 Profile for Agent-to-Agent (A2A) Communications", September 2025, <https://datatracker.ietf.org/doc/html/draft-liu-oauth-a2a-profile-00>.

[I-D.ni-wimse-ai-agent-identity]
Ni, T., "AI Agent Identity", October 2025, <https://datatracker.ietf.org/doc/html/draft-ni-wimse-ai-agent-identity-01>.

[I-D.rrk-object-based-media-usecase]
Ramdhany, R., Race, N., and D. King, "Use Case and Challenges for the Deployment of Object-Based Media across the Internet", March 2026, <https://datatracker.ietf.org/doc/html/draft-rrk-object-based-media-usecase>.

[OWASP-ASI2026]
OWASP Gen AI Security Project - Agentic Security Initiative, "OWASP Top 10 for Agentic Applications for 2026, Version 2026", December 2025, <https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.

[RFC7542] DeKok, A., "The Network Access Identifier", RFC 7542, DOI 10.17487/RFC7542, May 2015, <https://www.rfc-editor.org/info/rfc7542>.

[RFC7643] Hunt, P., Ed., Grizzle, K., Wahlstroem, E., and C. Mortimore, "System for Cross-domain Identity Management: Core Schema", RFC 7643, DOI 10.17487/RFC7643, September 2015, <https://www.rfc-editor.org/info/rfc7643>.

[RFC7644] Hunt, P., Ed., Grizzle, K., Ansari, M., Wahlstroem, E., and C. Mortimore, "System for Cross-domain Identity Management: Protocol", RFC 7644, DOI 10.17487/RFC7644, September 2015, <https://www.rfc-editor.org/info/rfc7644>.

[RFC9052] Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", STD 96, RFC 9052, DOI 10.17487/RFC9052, August 2022, <https://www.rfc-editor.org/info/rfc9052>.

[RFC9421] Backman, A., Ed., Richer, J., Ed., and M. Sporny, "HTTP Message Signatures", RFC 9421, DOI 10.17487/RFC9421, February 2024, <https://www.rfc-editor.org/info/rfc9421>.

[RFC9449] Fett, D., Campbell, B., Bradley, J., Lodderstedt, T., Jones, M., and D. Waite, "OAuth 2.0 Demonstrating Proof of Possession (DPoP)", RFC 9449, DOI 10.17487/RFC9449, September 2023, <https://www.rfc-editor.org/info/rfc9449>.

[W3C.VC] Sporny, M., Longley, D., and D. Chadwick, "Verifiable Credentials Data Model v2.0", November 2024, <https://www.w3.org/TR/vc-data-model-2.0/>.

Appendix A. Example Flows (Informative)

This appendix provides non-normative examples illustrating how the framework can be applied in common deployment patterns.

A.1. Same-Protocol, Cross-Domain Invocation with Token Exchange

1. Agent A in Domain A receives a user task and obtains a local access token scoped to that task.

2. Agent A presents its token to Domain A’s Security Token Service, which performs OAuth 2.0 Token Exchange [RFC8693] for Domain B.

3. Domain B issues a short-lived token constrained to resource, purpose, and delegation depth.

4. Agent A calls Agent B using mutual authentication; Agent B validates token, call-chain context, and policy before execution.

5. Both domains emit audit records tied to the same workflow context.

A.2. Cross-Protocol Invocation Through a Gateway

1. Agent X (Protocol X) sends a delegated request to a gateway at the Domain boundary.

2. The gateway authenticates Agent X, validates delegation constraints, and enforces inbound policy.

3. The gateway translates protocol semantics for Protocol Y while preserving delegation and workflow context in signed form.

4. The gateway authenticates to Agent Y (Protocol Y side) and submits a policy-constrained request.

5. The gateway records translation, policy decision, and outbound request evidence for auditability.

Appendix B. Current Agentic Risk Landscape (Informative)

This appendix provides a non-normative, time-stamped snapshot of a current agentic risk taxonomy to support threat modelling and control prioritisation. Taxonomies evolve quickly; implementers SHOULD consult current source publications.

One recent external publication, the OWASP Top 10 for Agentic Applications for 2026 [OWASP-ASI2026], identifies the following risk categories:

* ASI01: Agent Goal Hijack

* ASI02: Tool Misuse and Exploitation

* ASI03: Identity and Privilege Abuse

* ASI04: Agentic Supply Chain Vulnerabilities

* ASI05: Unexpected Code Execution (RCE)

* ASI06: Memory and Context Poisoning

* ASI07: Insecure Inter-Agent Communication

* ASI08: Cascading Failures

* ASI09: Human-Agent Trust Exploitation

* ASI10: Rogue Agents

The following high-level mapping provides one illustrative view of how this framework's controls may relate to the categories above:

* Identity and privilege controls (Section 7, Section 8.3, Section 8.4) are particularly relevant to ASI01, ASI02, ASI03, and ASI09.

* Token exchange, delegation-chain integrity, and task-context binding (Section 8.2, Section 8.4, Section 8.5) are relevant to ASI01, ASI02, ASI03, ASI07, and ASI09.

* Lifecycle controls (Section 9) are relevant to ASI03, ASI07, and ASI10, including rapid response to compromise and credential misuse.

* Auditability and transparency controls (Section 10) are relevant to ASI01, ASI05, ASI08, ASI09, and ASI10 for detection, investigation, and accountability.

This appendix is informative only and reflects one external risk taxonomy at a point in time. Normative requirements remain in the main body of this document. The treatment of external taxonomies is to be discussed in more detail in a later version of this document.

Authors' Addresses

D. King
Lancaster University
Email: d.king@lancaster.ac.uk

R. Ramdhany
BBC
Email: rajiv.ramdhany@bbc.co.uk

Chunchi Peter Liu
Huawei Technologies
Email: liuchunchi@huawei.com