Network Working Group
Internet-Draft
Intended status: Informational
Expires: 15 November 2026

C. Munoz
Keel API, Inc.

14 May 2026

Signed Authorization-Evidence Records for WIMSE-Authorized AI Agent Actions

draft-munoz-wimse-authorization-evidence-00

Abstract

This document specifies a companion profile to the AI Agent Authentication and Authorization draft [I-D.klrc-aiagent-auth], defining a signed authorization-evidence record produced by WIMSE-authorized AI agent actions. The evidence record (referred to as a Permit) commits cryptographically to the canonical request bytes dispatched after authorization, satisfies the audit minimum requirements enumerated in Section 11 of the companion draft, and composes with HTTP Message Signatures, OAuth access tokens, and Shared Signals Framework eventing without requiring modifications to those existing standards. The profile is anchored in the SCITT profile defined in [I-D.munoz-scitt-permit-profile].

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 15 November 2026.

Copyright Notice

Table of Contents

1. Introduction

The AI Agent Authentication and Authorization draft [I-D.klrc-aiagent-auth] composes existing standards (SPIFFE, WIMSE, OAuth 2.0, HTTP Message Signatures) to specify how AI agents obtain identity, authenticate, and acquire runtime authorization for invocations against tools, services, and large language models. That draft explicitly states it does not define new protocols.

A consequence of this scoping decision is that the draft does not define a signed evidence record of the authorization decision itself. Section 11 of the draft enumerates audit minimum requirements (authenticated agent identifier, delegated subject, resource or tool, action requested and authorization decision, timestamp and correlation identifier, attestation or risk state, remediation or revocation events) but leaves the format of the evidence record to implementations.

This document defines such a record. The record is a Permit, as specified in the companion SCITT profile [I-D.munoz-scitt-permit-profile], with this document adding the WIMSE-specific integration guidance: how Permits represent SPIFFE identities, how they satisfy the Section 11 audit minimum requirements, how they integrate with HTTP Message Signatures and OAuth access tokens, and how Permit lifecycle states bridge to Shared Signals Framework eventing.

1.1. Scope

This profile specifies:

* How to populate Permit subject fields with SPIFFE URIs
* A crosswalk from Section 11 of [I-D.klrc-aiagent-auth] to Permit fields
* Optional integration with HTTP Message Signatures [RFC9421] for request-level signature coverage that extends Permit’s canonical-body binding to header coverage
* Optional integration with OAuth access tokens through a Permit-ID claim that allows runtime tokens to reference the signed authorization-evidence record
* A descriptive (non-normative) bridge between Permit lifecycle state transitions and Shared Signals Framework / Continuous Access Evaluation Profile / Risk Incident Sharing eventing

This profile does not specify:

* Modifications to [I-D.klrc-aiagent-auth] or any standard it builds upon

* A new authorization protocol or token format

* Identity management beyond what SPIFFE/WIMSE define

* The Permit object itself; that specification is in [KEEL-PERMIT] and the SCITT profile in [I-D.munoz-scitt-permit-profile]

1.2. Relationship to Other Specifications

This document is a companion to [I-D.munoz-scitt-permit-profile]. That document defines the Permit object as a SCITT Signed Statement and specifies the COSE_Sign1 envelope, Receipt format, and verification rules. This document extends that profile with WIMSE-specific integration guidance. An implementation that conforms to both profiles produces evidence that is consumable by SCITT-aware verifiers and that satisfies the audit minimum requirements of [I-D.klrc-aiagent-auth].

1.3. Terminology

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 terminology from [I-D.klrc-aiagent-auth]: Agent Identifier, Agent Credential, Workload Identity Token (WIT), Workload Proof Token (WPT), Transaction Token, Agent Authentication, Agent Authorization, Audit Event.

This document uses terminology from [KEEL-PERMIT] and [I-D.munoz-scitt-permit-profile]: Permit, Closure Record, binding_request_hash, dispatch_request_digest_v1, Signed Statement, Receipt, Transparent Statement.

2. Background

2.1. The Audit Gap in WIMSE-Style Authorization

A WIMSE-authorized AI agent action proceeds as follows: the agent authenticates via mTLS or WPT, obtains an OAuth access token from an authorization server, presents the token at a resource server, and invokes the resource. Each step produces transient authentication evidence (channel binding, signed proof-of-possession tokens, validated access tokens) but the authorization decision itself does not produce a durable signed record.

Section 11 of [I-D.klrc-aiagent-auth] enumerates the minimum fields an audit event MUST record:

* authenticated agent identifier
* delegated subject (user or system) when present
* resource or tool being accessed
* action requested and authorization decision
* timestamp and transaction or request correlation identifier
* attestation or risk state influencing the decision
* remediation or revocation events and their cause

These are evidence requirements without a defined evidence format. The draft is explicit that the format is left to implementations.

2.2. The Permit as Evidence Record

A Permit, as specified in [I-D.munoz-scitt-permit-profile] and [KEEL-PERMIT], is a Signed Statement that records the pre-execution authorization decision for an AI agent action. It satisfies the Section 11 audit minimum requirements directly, binds to the dispatched request bytes cryptographically, and is verifiable independently of the issuer. It serves as the audit evidence record the WIMSE draft requires.

3. The WIMSE Authorization-Evidence Profile

3.1. SPIFFE Subject Typing

The Permit object’s subject_type and subject_id fields together identify the actor that an authorization decision applies to. When the agent identity is established through SPIFFE/WIMSE:

* subject_type MUST be set to "spiffe".

* subject_id MUST be the agent's SPIFFE URI, in the format spiffe://{trust-domain}/{path}.

Implementations MAY also populate a delegated subject identifier in the Permit's decision_details.delegated_subject field when the agent is acting on behalf of another principal. The format of delegated_subject SHOULD identify the trust domain and identifier of the delegated principal.

When the agent's WIT is presented at authorization time, the WIT's serial number or thumbprint MAY be recorded in decision_details.credential_thumbprint as additional binding context. This is descriptive metadata; the Permit's binding_request_hash remains the cryptographic commitment to the dispatched request.

3.2. Audit Minimum Requirements Crosswalk

The following table maps Section 11 of [I-D.klrc-aiagent-auth] to Permit fields specified in [KEEL-PERMIT]:

Section 11 Requirement	Permit Field
authenticated agent identifier	subject_type + subject_id (SPIFFE URI)
delegated subject	decision_details.delegated_subject
resource or tool being accessed	resource_provider + resource_model + action_name
action requested	action_name + actions_json
authorization decision	decision (allow / deny / challenge) + decision_details
timestamp	created_at
transaction or request correlation identifier	idempotency_key + request_fingerprint
attestation or risk state	decision_details.code + routing
remediation or revocation events	status lifecycle (evaluated, bound, dispatched, closed, expired, revoked) + chain entries

Table 1

A Permit emitted by a conforming Issuer is intended to satisfy the currently enumerated Section 11 audit minimum requirements without additional structural extension.

3.3. HTTP Message Signature Integration

Section 9.2.2 of [I-D.klrc-aiagent-auth] describes signing of HTTP request components via HTTP Message Signatures [RFC9421]. The mandatory signed components include method, request-target, content digest, and the WIT itself.

The Permit’s binding_request_hash covers the canonical bytes of the request body but does not directly cover request method, request-target, or selected headers. For deployments requiring header coverage:

* The Issuer MAY compute an HTTP Message Signature over the full request and record the signature input string and signature value in the paired Closure Record's http_message_signature field.

* The Closure Record's dispatch_request_digest_v1 continues to commit to the canonical request body bytes; the HTTP Message Signature commits to the full request envelope.

* Verifiers of the Permit profile MAY use the HTTP Message Signature as an additional verification step beyond the binding_request_hash equality check.

This composition layers RFC 9421 signature coverage on top of Permit's canonical-body binding without requiring expansion of the canonicalization function in [KEEL-PERMIT].

3.4. OAuth Access Token Composition

OAuth access tokens issued under the [I-D.klrc-aiagent-auth] model carry standard claims (client_id, sub, aud, scope) and MAY carry profile-specific claims.

For deployments where a runtime access token should reference the signed pre-execution authorization-evidence record:

* The Authorization Server MAY include a claim named permit_id whose value is the Permit's id (a UUID).

* A Resource Server validating the access token MAY retrieve the referenced Permit and verify the relationship between the in-token claims (client_id, sub, aud, scope) and the Permit's subject, resource, and decision fields.

* The permit_id claim MUST NOT be treated as a substitute for Permit verification. Verifiers requiring strong evidence MUST retrieve and verify the Permit's COSE_Sign1 signature and Receipt per [I-D.munoz-scitt-permit-profile].

This composition keeps the OAuth access token in its runtime role (short-lived authorization grant) while making the durable signed authorization-evidence record retrievable from the token.

Profile-specific claim registration for permit_id is out of scope for this document. Implementations MAY use a private claim under the issuer's namespace until registration is sought.

3.5. Transaction Token Linkage

Section 10.4 of [I-D.klrc-aiagent-auth] permits Transaction Tokens (RFC 8693 [RFC8693]) for downscoped per-call context. A Transaction Token's transaction identifier MAY appear in:

* The Permit's idempotency_key field

* The paired Closure Record's correlation_id field

Both fields permit a Verifier consuming Permits and Transaction Tokens to correlate the runtime per-call context with the durable signed evidence record. This profile does not require Transaction Token use; it specifies the correlation pattern for deployments that adopt them.

3.6. Permit Chains for Multi-Step Authorization

The guidance in this document treats a Permit as the authorization-evidence record for a single authorized AI agent action. Many WIMSE-authorized deployments involve multi-step agent workflows or delegated authority, where one authorization decision leads to a sequence of dependent actions across tools, services, or models.

The Permit Chains construction specified in [KEEL-PERMIT] extends a single Permit into an ordered, hash-linked sequence of Permits, each committing to the preceding Permit's identifier and digest. WIMSE-style runtime authorization composes with both forms: a single Permit is sufficient evidence for a single authorized action, and a Permit Chain is the natural extension for multi-step or delegated authority, binding the per-step Permits into one verifiable record of the whole authorized sequence.

The SPIFFE subject typing, audit minimum requirements crosswalk, HTTP Message Signature integration, and OAuth access token composition guidance in this document applies per-Permit, and therefore applies without modification to each Permit in a chain. This document does not specify the Permit Chain construction itself; that specification is in [KEEL-PERMIT].

3.7. Lifecycle State and Eventing Bridge

The Permit object carries a status lifecycle ([KEEL-PERMIT]):

* evaluated

* bound

* dispatched

* closed

* expired

* revoked

State transitions emit chain entries with severity and outcome metadata. [I-D.klrc-aiagent-auth] expects authorization-state changes (revocation, suspension, attestation degradation) to propagate as Shared Signals Framework (SSF), Continuous Access Evaluation Profile (CAEP), or Risk Incident Sharing (RISC) signals.

A bridge from Permit chain events to SSF / CAEP / RISC signals SHOULD be provided by deployments requiring real-time authorization-state propagation. The bridge transforms a chain entry of severity "warning" or "error" affecting a Permit's revoked or expired status into an outbound SSF event of the appropriate type. This profile does not specify the bridge normatively; it documents the integration pattern.

A future revision of this profile MAY specify normative bridge event formats once production deployments demonstrate stable patterns.

4. Composition with the SCITT Profile

A Permit emitted under this WIMSE companion profile and signed under [I-D.munoz-scitt-permit-profile] satisfies both:

* SCITT verifiers (via the COSE_Sign1 Signed Statement envelope and chain-entry Receipt)

* [I-D.klrc-aiagent-auth] Section 11 audit minimum requirements

Implementations MAY produce Permits that conform only to this companion profile (without the SCITT COSE_Sign1 envelope) if SCITT compatibility is not required. In that case, the Permit's legacy signature envelope [KEEL-PERMIT] satisfies the integrity requirement for Section 11 evidence purposes, but the artifact is not consumable by SCITT-aware verifiers.

A future version of this profile MAY deprecate the legacy-only mode in favor of SCITT-compatible emission. This profile-00 does not.

5. Security Considerations

5.1. Token Lifetime versus Evidence Persistence

OAuth access tokens, WITs, and WPTs are short-lived runtime credentials. Permits are long-lived signed evidence records. The two are linked by the permit_id claim (when present) but their verification lifetimes are distinct.

Compromise of a runtime credential does not retroactively invalidate the Permit (because the Permit was signed by the Issuer at decision time, not by the runtime credential). Compromise of an Issuer signing key, however, permits forgery of Permits and SHOULD be addressed through key-rotation procedures specified in the Issuer's key manifest.

5.2. Subject Identifier Disclosure

A Permit's subject_id, when a SPIFFE URI, identifies the agent's trust domain and workload path. When a delegated subject is present, the delegated principal is also identified. Verifiers processing Permits SHOULD apply access controls appropriate to the sensitivity of the subject identifiers.

5.3. Transaction Token Replay

Transaction Tokens carry per-call context that should not appear in long-lived hashes. The Permit's binding_request_hash is computed over canonical request bytes after volatile-key stripping ([KEEL-PERMIT]); Transaction Token identifiers are stripped if they appear in the request body. Transaction Token correlation recorded in the Permit's idempotency_key field is not subject to the stripping rules; this is intentional, since the correlation ID is the same artifact that downstream eventing references.

5.4. HTTP Message Signature Composition

When an HTTP Message Signature is recorded in the Closure Record, the signature's signed components include the WIT itself (per [I-D.klrc-aiagent-auth] Section 9.2.2). The WIT carries the agent's public key reference; the signature commits to the agent's authentication context at dispatch time. Verifiers validating both the Permit and the HTTP Message Signature obtain two independent cryptographic commitments to the dispatched request.

5.5. SSF Event Integrity

The descriptive bridge from Permit chain events to SSF events is not normative in this profile. Deployments that adopt the bridge MUST ensure that bridge-generated SSF events are produced by a component with appropriate trust relative to the Permit Issuer and the SSF transmitter. Bridge components SHOULD sign or otherwise authenticate the events they generate.

6. Privacy Considerations

6.1. Delegated Subject Identification

When a delegated subject is identified in the Permit's decision_details.delegated_subject, the privacy considerations of the delegated principal apply. Issuers SHOULD consider whether direct identifier inclusion is appropriate or whether a pseudonymized identifier is required, depending on the audience consuming the Transparent Statement.

6.2. Trust Domain Disclosure

A SPIFFE URI in subject_id discloses the agent's trust domain. The trust domain may identify an organization, a deployment environment, or a particular system. Verifiers and Transparency Service operators SHOULD treat trust domain disclosure with appropriate sensitivity.

6.3. Long-Lived Identifiers

The combination of subject_id, policy_id, and resource fields across many Permits supports re-identification of agents and correlation across requests. Operators SHOULD apply data minimization and access control to the audit-export delivery mechanism for Permits and their chain entries.

7. IANA Considerations

This document has no immediate IANA requests. A future revision MAY request registration of the permit_id OAuth claim and the http_message_signature Closure Record extension.

8. Implementation Status

This section is to be removed before publication as an RFC.

A reference Issuer implementation is published at [KEEL-PERMIT] under Apache 2.0. SPIFFE subject typing is supported by the open subject model. The Section 11 audit minimum requirements crosswalk is

satisfied by the Permit fields as deployed. HTTP Message Signature integration, OAuth permit_id claim, and SSF bridge are work in progress.

9. Acknowledgments

The author thanks the authors of [I-D.klrc-aiagent-auth] — Pieter Kasselman, Jeff Lombardo, Yaroslav Rosomakho, Brian Campbell, and Nick Steele — for the framing of the AI agent authentication and authorization model that this companion profile extends. The author also thanks the SCITT working group and the authors of adjacent profiles [I-D.emirdag-scitt-ai-agent-execution] and [I-D.veridom-omp] for related work.

10. References

10.1. Normative References

[I-D.klrc-aiagent-auth]
Kasselman, P., Lombardo, J., Rosomakho, Y., Campbell, B., and N. Steele, "AI Agent Authentication and Authorization", Work in Progress, Internet-Draft, draft-klrc-aiagent-auth-01, 30 March 2026, <https://datatracker.ietf.org/doc/html/draft-klrc-aiagent-auth-01>.

[I-D.munoz-scitt-permit-profile]
Munoz, C., "A SCITT Profile for Pre-Execution AI Action Authorization Records", Work in Progress, Internet-Draft, draft-munoz-scitt-permit-profile-00, 15 May 2026, <https://datatracker.ietf.org/doc/html/draft-munoz-scitt-permit-profile-00>.

[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>.

[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/rfc/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/rfc/rfc9421>.

10.2. Informative References

[I-D.emirdag-scitt-ai-agent-execution]
Emirdag, P., "AI Agent Execution Profile of SCITT", Work in Progress, Internet-Draft, draft-emirdag-scitt-ai-agent-execution-00, 11 April 2026, <https://datatracker.ietf.org/doc/html/draft-emirdag-scitt-ai-agent-execution-00>.

[I-D.ietf-scitt-architecture]
Birkholz, H., Delignat-Lavaud, A., Fournet, C., Deshpande, Y., and S. Lasker, "An Architecture for Trustworthy and Transparent Digital Supply Chains", Work in Progress, Internet-Draft, draft-ietf-scitt-architecture-22, 10 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-scitt-architecture-22>.

[I-D.veridom-omp]
Adebayo, T. and O. Apalowo, "Operating Model Protocol (OMP) Core -- Version 02: Invariant 3 -- Verifiable Delegation Binding", Work in Progress, Internet-Draft, draft-veridom-omp-02, 13 May 2026, <https://datatracker.ietf.org/doc/html/draft-veridom-omp-02>.

[KEEL-PERMIT]
Keel API, Inc., "Keel Permit Specification", 2026, <https://github.com/keelapi/keel-permit/blob/1818c3e04eddf9a2ab6231486ca2cdb2d250ec74/spec/permit-chain-v1.md>.

[RFC3161] Adams, C., Cain, P., Pinkas, D., and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", RFC 3161, DOI 10.17487/RFC3161, August 2001, <https://www.rfc-editor.org/rfc/rfc3161>.

[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/rfc/rfc8693>.

[RFC9068] Bertocci, V., "JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens", RFC 9068, DOI 10.17487/RFC9068, October 2021, <https://www.rfc-editor.org/rfc/rfc9068>.

Appendix A. Open Issues for -01 and Beyond

This section is to be removed before publication as an RFC.

Open issues:

1. Whether to specify the permit_id OAuth claim as a normative addition or to leave it as an implementation pattern.

2. Whether to define the http_message_signature field in the Closure Record normatively, including its exact serialization.

3. Whether to specify normative bridge event formats from Permit chain events to SSF / CAEP / RISC signals.

4. The exact registration mechanism for the permit_id claim (IANA OAuth Parameters registry, private vendor claim, or both).

5. Whether to specify a SPIFFE-required mode or to keep SPIFFE subject typing as optional.

Feedback on any of these is welcome on the WIMSE and SCITT mailing lists.

Author’s Address

Christian Munoz
Keel API, Inc.
Email: christian@keelapi.com
URI: https://keelapi.com