Supply Chain Integrity, Transparency, and Trust
Internet-Draft
Intended status: Informational
Expires: 3 August 2026

T. Kamimura
VeritasChain Standards Organization

30 January 2026

Verifiable AI Refusal Events using SCITT
draft-kamimura-scitt-refusal-events-02

Abstract

This document defines a claim set for recording AI content refusal events. The claim set specifies the semantic content and correlation rules for refusal audit trails, independent of any particular serialization format. The claims are designed to be carried within SCITT Signed Statements and verified using SCITT Receipts.

This specification addresses claim semantics and verification requirements; it does not mandate a specific encoding. A CDDL definition is provided for CBOR-based implementations, and equivalent JSON representations are shown in an appendix for illustration.

This specification provides auditability of logged refusal decisions. It does not define content moderation policies, classification criteria, or what AI systems should refuse.

About This Document

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

The latest version of this document, along with implementation resources and examples, can be found at <https://github.com/veritaschain/cap-spec>.

Discussion of this document takes place on the SCITT Working Group mailing list (scitt@ietf.org).

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/.

This Internet-Draft will expire on 3 August 2026.

Copyright Notice

Table of Contents

1. Introduction

AI systems capable of generating content increasingly implement safety mechanisms to refuse requests deemed harmful, illegal, or policy-violating. However, these refusal decisions typically leave no verifiable audit trail. When a system refuses to generate content, the event vanishes—there is no receipt, no log entry accessible to external parties, and no mechanism for third-party verification.

This document defines a claim set for recording such refusal events. The claim set specifies:

* The semantic meaning of each claim

* Correlation rules linking attempts to outcomes

* A completeness invariant for audit trail integrity

* Privacy-preserving approaches using cryptographic hashes

The claims are designed to be carried within SCITT Signed Statements [I-D.ietf-scitt-architecture] and verified using SCITT Receipts. This document does not mandate a specific serialization; implementations may use CBOR, JSON, or other formats as appropriate for their deployment context.

1.1. Motivation

The January 2026 Grok incident demonstrated the need for verifiable refusal records. The AI system generated millions of harmful images while the provider claimed moderation systems were functioning. External parties had no mechanism to verify whether refusals actually occurred or whether claimed safeguards were enforced.

This creates several problems:

* Regulators cannot independently verify that AI providers enforce stated policies
* Providers cannot prove to external auditors that specific requests were refused
* Third parties investigating incidents have no way to establish refusal without trusting provider claims

SCITT provides the infrastructure—Signed Statements, Transparency Services, and Receipts—to address this gap. This document defines the claim semantics for AI refusal events that can be carried within that infrastructure.

1.2. Scope

This document defines:

* A claim set for ATTEMPT and Outcome events
* Semantic definitions for each claim
* A completeness invariant for correlation
* A CDDL grammar for CBOR representation
* Integration guidance for SCITT Transparency Services

This document does NOT define:

* Content moderation policies
* Classification algorithms or risk scoring
* Mandatory serialization formats
* Transparency Service behavior beyond standard SCITT

* Regulatory compliance requirements

This specification is an application profile for SCITT, defining domain-specific claim semantics. It does not extend or modify SCITT architecture.

1.3. Limitations

This specification provides auditability of refusal decisions that are logged, not cryptographic proof that no unlogged generation occurred. An AI system that bypasses logging entirely cannot be detected by this mechanism alone.

Transparency Services do not enforce the completeness invariant defined in this document; such checks are performed by verifiers at the application level using the claim semantics defined herein.

1.4. Requirements Language

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.

2. Terminology

This document uses terminology from [I-D.ietf-scitt-architecture]. The following terms are specific to this specification:

Generation Request A request submitted to an AI system to produce content.

Refusal Event A decision by an AI system to decline a generation request.

ATTEMPT A claim set recording that a generation request was received.

DENY A claim set recording that a generation request was refused. References the corresponding ATTEMPT.

GENERATE A claim set recording successful content generation. References the corresponding ATTEMPT.

ERROR A claim set recording system failure during processing. References the corresponding ATTEMPT.

Outcome Any of DENY, GENERATE, or ERROR.

Completeness Invariant The property that every logged ATTEMPT has exactly one corresponding Outcome.

Verifiable Refusal Record An ATTEMPT claim set, a DENY claim set, and Receipts proving their registration with a Transparency Service.

2.1. Mapping to SCITT

This specification maps directly to SCITT primitives:

This Document	SCITT Primitive
ATTEMPT claim set	Signed Statement payload
Outcome claim set	Signed Statement payload
AI System	Issuer
Inclusion proof	Receipt

Table 1

3. Claim Set Definition

This section defines the claims for refusal events. Claims are specified by their semantic meaning; encoding is deployment-specific. Section 4 provides a CDDL grammar for CBOR representation.

3.1. Common Claims

The following claims appear in all event types:

event-type (REQUIRED) One of: "ATTEMPT", "DENY", "GENERATE", "ERROR". Identifies the event category.

event-id (REQUIRED) Unique identifier for this event. UUIDv7 [RFC9562] is RECOMMENDED for temporal ordering.

timestamp (REQUIRED) Time of event occurrence. In CBOR, MUST be either tag 0 (RFC 3339 date-time string) or tag 1 (epoch-based integer/float). Untagged integers representing Unix epoch seconds are also permitted.

issuer (REQUIRED) Identifier of the AI system that created this event. SHOULD be a URI.

3.2. ATTEMPT Claims

An ATTEMPT records receipt of a generation request. It MUST be created before any safety evaluation begins.

prompt-hash (REQUIRED) Cryptographic hash of the prompt content. MUST use SHA-256 or stronger. The original prompt MUST NOT be stored.

input-type (REQUIRED) Type of input: "text", "image", "text+image", "audio", "video", or "multimodal".

reference-input-hashes (OPTIONAL) Array of hashes for non-text inputs (images, audio, etc.).

session-id (OPTIONAL) Session identifier for correlation across requests.

actor-hash (OPTIONAL) Pseudonymized hash of the requesting entity.

model-id (OPTIONAL) Identifier of the AI model processing the request.

policy-id (OPTIONAL) Identifier of the content policy being applied.

The requirement to create ATTEMPT before safety evaluation prevents selective logging where only "safe" requests are recorded.

3.3. DENY Claims

A DENY records refusal of a generation request.

attempt-id (REQUIRED) The event-id of the corresponding ATTEMPT. This establishes the correlation required by the completeness invariant.

risk-category (OPTIONAL) Category of policy violation. Values are deployment-specific; Appendix A provides a non-normative taxonomy.

risk-score (OPTIONAL) Confidence score, 0.0 to 1.0.

refusal-reason (OPTIONAL) Human-readable reason. SHOULD NOT contain prompt content.

human-override (OPTIONAL) Boolean indicating human reviewer

involvement.

3.4. GENERATE Claims

A GENERATE records successful content generation.

attempt-id (REQUIRED) The event-id of the corresponding ATTEMPT.

output-hash (OPTIONAL) Hash of the generated content, for correlation with downstream provenance (e.g., C2PA manifests).

3.5. ERROR Claims

An ERROR records system failure, not a policy decision.

attempt-id (REQUIRED) The event-id of the corresponding ATTEMPT.

error-code (OPTIONAL) Machine-readable error identifier.

error-message (OPTIONAL) Human-readable error description.

ERROR does not indicate policy enforcement. A high ERROR rate may indicate operational issues or adversarial inputs.

3.6. Completeness Invariant

The completeness invariant is the core integrity property:

For every ATTEMPT with event-id A that is registered with a Transparency Service, there MUST exist exactly one Outcome (DENY, GENERATE, or ERROR) with attempt-id equal to A.

Formally:

$$|\{ATTEMPT\}| = |\{DENY\}| + |\{GENERATE\}| + |\{ERROR\}|$$

where each Outcome references exactly one ATTEMPT, and no ATTEMPT is referenced by multiple Outcomes.

Violations indicate:

* ATTEMPT without Outcome: incomplete logging or in-progress request
* Outcome without ATTEMPT: fabricated outcome or logging failure
* Multiple Outcomes per ATTEMPT: data integrity failure

Verifiers SHOULD check this invariant when auditing event streams. Transparency Services do not enforce this invariant; it is an application-level semantic constraint.

3.7. Timing Requirements

To maintain audit trail integrity:

* ATTEMPT MUST be created before safety evaluation begins
* Outcome SHOULD be created promptly after decision
* Outcome timestamp MUST be greater than or equal to ATTEMPT timestamp

Some scenarios require delayed outcomes (human review, system recovery). Implementations SHOULD document expected latency bounds.

4. CDDL Definition

This section provides a CDDL [RFC8610] grammar for CBOR-based representation of the claim set. This grammar is RECOMMENDED for implementations using CBOR encoding within SCITT Signed Statements.

Implementations using other encodings (e.g., JSON) SHOULD maintain semantic equivalence with this definition.

```
; Refusal Event Claim Set
; draft-kamimura-scitt-refusal-events-02

refusal-event = attempt-event / outcome-event

; Common claims (appear in all events)
common-claims = (
  event-type: event-type-value,
  event-id: uuid,
  timestamp: time,
  issuer: tstr
)

event-type-value = "ATTEMPT" / "DENY" / "GENERATE" / "ERROR"

; ATTEMPT event
attempt-event = {
  common-claims,
  prompt-hash: hash-value,
  input-type: input-type-value,
  ? reference-input-hashes: [+ hash-value],
```

```
    ? session-id: uuid,
    ? actor-hash: hash-value,
    ? model-id: tstr,
    ? policy-id: tstr,
    * tstr => any   ; extension point
  }

  input-type-value = "text" / "image" / "text+image" /
                     "audio" / "video" / "multimodal"

  ; Outcome events
  outcome-event = deny-event / generate-event / error-event

  deny-event = {
    common-claims,
    attempt-id: uuid,
    ? risk-category: tstr,
    ? risk-score: float16 .le 1.0,
    ? refusal-reason: tstr,
    ? human-override: bool,
    * tstr => any
  }

  generate-event = {
    common-claims,
    attempt-id: uuid,
    ? output-hash: hash-value,
    * tstr => any
  }

  error-event = {
    common-claims,
    attempt-id: uuid,
    ? error-code: tstr,
    ? error-message: tstr,
    * tstr => any
  }

  ; Supporting types
  uuid = bstr .size 16 / tstr       ; binary or string representation
  hash-value = bstr / tstr          ; raw bytes or prefixed string
  time = #6.0(tstr) / #6.1(number) / uint  ; tag 0 (date-time string),
                                           ; tag 1 (epoch), or raw uint
```

Notes on the CDDL definition:

* Extension points (* tstr => any) allow deployment-specific claims without breaking interoperability

* UUID may be binary (16 bytes) or string (RFC 9562 format)

* Hash values may be raw bytes or prefixed strings (e.g., "sha256:...")

* Timestamps support CBOR tag 0 (RFC 3339 date-time string), tag 1 (epoch-based number), or untagged epoch seconds

The claim names defined in this specification (e.g., "event-type", "prompt-hash") represent semantic labels. Implementations MAY use alternative key representations (such as short integer labels for CBOR efficiency) provided the semantic meaning is preserved and documented.

5. SCITT Integration

This section describes how refusal event claim sets integrate with SCITT infrastructure.

5.1. Encoding as Signed Statements

A refusal event claim set is carried as the payload of a SCITT Signed Statement:

* The claim set is serialized (CBOR RECOMMENDED)

* The serialized bytes become the COSE_Sign1 [RFC9052] payload

* The Issuer is the AI system’s signing identity

* Content-Type SHOULD indicate the claim set format

For CBOR-encoded claim sets, the Content-Type "application/cbor" or a more specific media type MAY be used.

5.2. Registration

After creating a Signed Statement, the Issuer SHOULD register it with a SCITT Transparency Service via SCRAPI [I-D.ietf-scitt-scrapi].

The Transparency Service returns a Receipt proving inclusion. Issuers SHOULD store Receipts for future verification requests.

Registration timing affects audit trail integrity:

* Prompt registration provides stronger temporal guarantees

* Batched registration may be acceptable for non-time-critical audits

5.3. Verification

A Verifiable Refusal Record consists of:

1. ATTEMPT Signed Statement and Receipt

2. DENY Signed Statement and Receipt

3. Verification that attempt-id in DENY equals event-id in ATTEMPT

Verifiers confirm:

* Both Receipts are valid for the Transparency Service

* Both Signed Statements have valid Issuer signatures

* The attempt-id correlation is correct

* Timestamps are consistent (Outcome >= ATTEMPT)

This demonstrates that a refusal was logged. It does not prove that no unlogged generation occurred.

5.4. Registration Policy Considerations

A Transparency Service MAY apply a Registration Policy that validates:

* Required claims are present

* Timestamp is within acceptable bounds

* Issuer is authorized (if restricted)

Transparency Services SHOULD NOT attempt to enforce the completeness invariant; this is an application-level verification performed by auditors.

6. IANA Considerations

This document has no IANA actions at this time.

Future revisions may request:

* Registration of a media type for refusal event claim sets

* Establishment of a registry for event-type values

* Establishment of a registry for risk-category values

The working group should consider whether standardized registries would benefit interoperability, or whether deployment-specific vocabularies are sufficient.

7. Security Considerations

7.1. Threat Model

This specification assumes:

* The AI system (Issuer) is partially trusted: expected to log events but may have bugs or be compromised

* The Transparency Service is partially trusted: provides append-only guarantees but may be compromised

* Verifiers are trusted to perform completeness checks

This specification does NOT protect against:

* An AI system that bypasses logging entirely

* Collusion between Issuer and Transparency Service

7.2. Omission Attacks

An adversary controlling the AI system might omit events to hide policy violations. The completeness invariant provides detection for logged events: auditors can identify ATTEMPTs without Outcomes.

However, if an ATTEMPT is never logged, this specification cannot detect the omission. Complete prevention requires external mechanisms (trusted execution, attestation [RFC9334]) outside this specification’s scope.

7.3. Log Equivocation

A malicious Transparency Service might present different views to different parties. Mitigations include:

* Gossiping of Signed Tree Heads between verifiers

* Registration with multiple Transparency Services

* External anchoring to public ledgers

7.4. Replay Attacks

Replay of old events is mitigated by:

* UUIDv7 temporal ordering
* Timestamp verification against registration time
* Receipt binding to specific log position

7.5. Dictionary Attacks on Hashes

An adversary with a dictionary of prompts could attempt to identify which prompt was used by hash comparison. Mitigations include:

* Access controls on event queries
* Rate limiting
* Time-limited retention

8. Privacy Considerations

8.1. Harmful Content

Refusal events may be triggered by harmful content. To prevent the audit log from becoming a repository of harmful content:

* prompt-hash stores only a hash, not the original
* refusal-reason SHOULD NOT quote prompt content
* Reference images are stored as hashes only

8.2. Actor Privacy

actor-hash provides pseudonymization. Implementations SHOULD:

* Use pseudonymous identifiers by default
* Maintain separate access-controlled identity mappings
* Support crypto-shredding for data deletion

8.3. Correlation Risks

Event metadata may enable correlation attacks:

* Timestamps reveal activity patterns
* session-id links requests
* model-id reveals system usage

Implementations SHOULD apply appropriate access controls.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/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>.

[RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <https://www.rfc-editor.org/info/rfc8610>.

[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>.

[RFC9562] Davis, K., Peabody, B., and P. Leach, "Universally Unique IDentifiers (UUIDs)", RFC 9562, DOI 10.17487/RFC9562, May 2024, <https://www.rfc-editor.org/info/rfc9562>.

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

[I-D.ietf-scitt-scrapi]
Steele, O., "SCITT Reference APIs", Work in Progress, Internet-Draft, draft-ietf-scitt-scrapi, 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-scitt-scrapi>.

9.2. Informative References

[RFC6962] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", RFC 6962, DOI 10.17487/RFC6962, June 2013, <https://www.rfc-editor.org/info/rfc6962>.

[RFC9334] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", RFC 9334, DOI 10.17487/RFC9334, January 2023, <https://www.rfc-editor.org/info/rfc9334>.

[CAP-SRP] VeritasChain Standards Organization, "CAP-SRP: Content/Creative AI Profile - Safe Refusal Provenance", https://github.com/veritaschain/cap-spec, 2026.

Appendix A. Risk Category Taxonomy (Non-Normative)

This appendix provides an example taxonomy for risk-category values. This taxonomy is non-normative; deployments MAY use different values appropriate to their context.

Category	Description
CSAM_RISK	Child sexual abuse material
NCII_RISK	Non-consensual intimate imagery
MINOR_SEXUALIZATION	Sexualization of minors
REAL_PERSON_DEEPFAKE	Unauthorized realistic depiction
VIOLENCE_EXTREME	Graphic violence
HATE_CONTENT	Discriminatory content
TERRORIST_CONTENT	Terrorism-related material
SELF_HARM_PROMOTION	Self-harm encouragement
COPYRIGHT_VIOLATION	Clear IP infringement
COPYRIGHT_STYLE_MIMICRY	Artist style imitation
OTHER	Other policy violations

Table 2

A future revision may define an IANA registry for risk categories if interoperability requires standardized values.

Appendix B. Deployment Considerations (Non-Normative)

This appendix provides guidance for deployments with varying assurance requirements. These are not conformance tiers; all deployments MUST satisfy the normative requirements in Section 3.

B.1. Basic Deployment

Suitable for voluntary transparency:

* Log ATTEMPT and Outcome events locally
* Register with Transparency Service periodically (daily)
* Retain events for 6+ months
* Provide Evidence Packs on request

B.2. Regulated Deployment

Suitable for regulatory compliance (e.g., EU AI Act):

* Register events promptly (within minutes)
* Implement continuous completeness monitoring
* Retain events for 2+ years
* Provide programmatic audit API
* Document latency bounds and retention policies

B.3. High-Assurance Deployment

Suitable for maximum accountability:

* Register events in real-time (within seconds)
* Use HSM for signing keys
* Register with multiple Transparency Services
* Implement external anchoring (blockchain, multiple TSAs)
* Retain events for 5+ years
* Support 24-hour incident response

Appendix C. JSON Representation Example (Non-Normative)

This appendix shows equivalent JSON representations of the claim sets defined in Section 3. These examples are for illustration; the normative definition is the CDDL in Section 4.

C.1. ATTEMPT Example

```
{
  "event-type": "ATTEMPT",
  "event-id": "019467a1-0001-7000-0000-000000000001",
  "timestamp": 1738159425,
  "issuer": "urn:example:ai-service:img-gen-prod",
  "prompt-hash": "sha256:7f83b1657ff1fc53b92dc18148a1d65d...",
  "input-type": "text+image",
  "reference-input-hashes": [
    "sha256:9f86d081884c7d659a2feaa0c55ad015..."
  ],
  "session-id": "019467a1-0001-7000-0000-000000000000",
  "actor-hash": "sha256:e3b0c44298fc1c149afbf4c8996fb924...",
  "model-id": "img-gen-v4.2.1",
  "policy-id": "content-safety-v2"
}
```

C.2. DENY Example

```
{
  "event-type": "DENY",
  "event-id": "019467a1-0001-7000-0000-000000000002",
  "timestamp": 1738159426,
  "issuer": "urn:example:ai-service:img-gen-prod",
  "attempt-id": "019467a1-0001-7000-0000-000000000001",
  "risk-category": "NCII_RISK",
  "risk-score": 0.94,
  "refusal-reason": "Content policy violation detected"
}
```

Appendix D. Evidence Pack Format (Non-Normative)

An Evidence Pack is a self-contained collection of events suitable for regulatory submission or third-party audit. This format is deployment-specific and not required for conformance.

A typical Evidence Pack contains:

* Manifest with metadata and checksums
* Serialized events (CBOR or JSON Lines)
* Receipts from Transparency Services
* Completeness verification results
* Public keys for signature verification

The CAP-SRP specification [CAP-SRP] defines a detailed Evidence Pack format for content AI applications.

Acknowledgements

The authors thank the SCITT Working Group for feedback on earlier revisions, particularly regarding SCITT charter scope and the distinction between claim semantics and payload formats.

This work builds upon transparency log concepts from Certificate Transparency [RFC6962] and the SCITT architecture.

Author’s Address

TOKACHI KAMIMURA
VeritasChain Standards Organization
Japan
Email: standards@veritaschain.org
URI: https://veritaschain.org