Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 19, 2026

E. Yalcinkaya
Luminesce Limited

March 19, 2026

An ASIL-M Profile for Multi-Root Evidence Synthesis in RATS
draft-yalcinkaya-rats-asil-m-00

Abstract

This document defines an application profile for Remote ATtestation procedureS (RATS) deployments in which authorization decisions depend on evidence from multiple independent trust roots. The profile is intended for AI inference systems that require explicit cross-root appraisal before execution is authorized.

The document specifies three related artifacts:

* an Attestation Evidence Synthesis Protocol for combining multiple evidence sets into one deterministic appraisal outcome;

* the Twin Attestation Policy Language (TAPL), a constrained policy language for deterministic multi-root appraisal; and

* the Canonical Attestation Record (CAR), an audit-oriented envelope that binds evidence references, appraisal outputs, freshness information, and replay-verification metadata.

CAR is not a replacement for Evidence or Attestation Results as used in the RATS architecture. Instead, it is a higher-layer profile for packaging multi-root appraisal state and application-specific bindings for replay and audit.

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at https://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at https://www.ietf.org/shadow.html

This Internet-Draft will expire on September 19, 2026.

Copyright Notice

respect to this document.

1. Introduction

Some AI inference systems are intentionally deployed across heterogeneous trust roots in order to reduce common-mode failure, implementation monoculture, and correlated compromise. In such systems, a single attestation-verification outcome is not sufficient for a Relying Party decision. Instead, the Relying Party requires a synthesized appraisal over multiple evidence sets and an auditable binding between that synthesized appraisal and the inference decision being authorized.

The RATS architecture already defines the relevant roles and message classes for this environment: Attesters produce Evidence, Verifiers appraise that Evidence according to policy, and Relying Parties use Attestation Results as inputs to authorization. What is not defined by the base architecture is an application profile for deterministic synthesis across multiple independent evidence sources for one AI inference authorization event.

This document defines such a profile. It does not introduce new RATS roles. It defines:

* a deterministic synthesis procedure across multiple evidence sets;
* a constrained appraisal-policy language for multi-root decisions; and
* an audit-oriented container for binding the resulting appraisal state to freshness and replay metadata.

2. Terminology

The key words "MUST", "MUST NOT", "SHOULD", and "MAY" in this document are to be interpreted as described in BCP 14.

This document uses the RATS terms "Attester", "Evidence", "Verifier", "Attestation Results", "Relying Party", and "Appraisal Policy" as defined by RFC 9334.

This document additionally uses the following profile terms:

* Multi-root evidence set: the collection of Evidence objects that correspond to the same authorization event but originate from two or more independent trust roots.
* Synthesized appraisal: the deterministic outcome produced after individual evidence appraisal and cross-root policy evaluation.
* TAPL: a constrained language for expressing threshold checks, predicate checks, and decision outputs over a multi-root evidence set.
* CAR: an audit-oriented envelope that binds evidence references, synthesized appraisal state, freshness information, and replay metadata.
* Semantic binding: application-specific metadata that binds the attested execution context to the inference event that the Relying Party is authorizing.
* Independence metrics: policy-level attestation-plane measures that verify the absence of shared vendor, cryptographic stack, or operational control plane across evidence sources. These are appraisal-layer predicates and do not depend on graph-theoretic

extensions.

3. Problem Statement

In a conventional single-root deployment, a Relying Party may accept Attestation Results derived from one Evidence source and authorize execution accordingly. In a multi-root AI deployment, that is insufficient for two reasons.

First, the authorization decision may depend on policy predicates that compare properties across roots rather than within a single root. For example, the Relying Party may require evidence that two execution paths are not derived from the same vendor, cryptographic stack, or operational control plane.

Second, the Relying Party may need a replayable record showing which combination of evidence, policy version, and freshness context caused a specific authorization outcome for a specific inference event.

Existing attestation workflows do not define how multiple heterogeneous evidence sources should be combined into one deterministic appraisal outcome for a single authorization decision. This profile makes those synthesis and packaging rules explicit for AI inference deployments.

4. Attestation Evidence Synthesis Protocol

4.1. Overview

The Attestation Evidence Synthesis Protocol defined here is a profile procedure executed by a Verifier, or by a Verifier-associated policy service, before a Relying Party authorizes an inference event.

The procedure is:

1. Collect Evidence objects for the same authorization event from two or more Attesters.

2. Verify each Evidence object according to its applicable Appraisal Policy for Evidence.

3. Reject stale, incomplete, or mismatched evidence sets before synthesis begins.

4. Evaluate cross-root predicates and thresholds over the verified evidence set.

5. Produce one synthesized appraisal outcome.

6. Bind that outcome to freshness context, policy identity, and the application-specific inference event identifier.

7. Emit Attestation Results and, optionally, a CAR for audit and replay.

4.2. Inputs

The minimum synthesis inputs are:

* one or more Evidence objects per authorization event;

* a policy identifier and policy digest;

* freshness inputs sufficient to detect replay or mixed-session reuse; and

* the application event identifier to which the synthesized appraisal

will apply.

4.3. Outputs

The synthesis procedure produces:

* a deterministic decision such as ALLOW, DEGRADE, BLOCK, or QUARANTINE;

* a list of evaluated predicates or thresholds sufficient for audit; and

* Attestation Results suitable for use by a Relying Party.

When audit and replay support are needed, the procedure SHOULD also emit a CAR.

5. TAPL as Constrained Appraisal Policy

5.1. Design Goal

TAPL is a constrained policy language for deterministic multi-root appraisal. It is intended to express profile-specific Appraisal Policy logic, not to define a general-purpose policy engine.

TAPL is deliberately limited to:

* threshold checks,
* predicate checks,
* explicit decision outputs, and
* side-effect-free evaluation.

5.2. Example Grammar

```
PolicyDef        ::= "policy" Identifier "{" Directives Rules "}"
Directives       ::= Directive*
Directive        ::= "enforce_profile" ProfileLevel ";"
                   | "set_threshold" Identifier "=" FloatLiteral ";"

Rules            ::= Rule+
Rule             ::= "on" ContextTrigger "{" LogicBlock "}" "->" Decision ";"

ContextTrigger   ::= "attestation" | "semantic_binding" | "freshness"
LogicBlock       ::= Statement*

Statement        ::= "if" Condition "then" Decision ";"
                   | "require" Condition ";"

Condition        ::= Expression
Expression       ::= Term (("AND" | "OR") Term)*
Term             ::= "NOT" Term | "(" Expression ")" | Predicate | Comparison

Comparison       ::= MetricIdentifier Operator NumericLiteral
Operator         ::= ">=" | "<=" | "==" | ">" | "<"

Predicate        ::= "verify_evidence(" RootIdentifier ")"
                   | "check_predicate(" PredicateIdentifier ")"

RootIdentifier   ::= Identifier
PredicateIdentifier ::= Identifier
MetricIdentifier ::= Identifier

Decision         ::= "ALLOW" | "DEGRADE" | "BLOCK" | "QUARANTINE"
ProfileLevel     ::= "L0" | "L1" | "L2" | "L3" | "L4"

Identifier       ::= Letter (Letter | Digit | "_")*
```

```
FloatLiteral     ::= Digit+ "." Digit+
NumericLiteral   ::= FloatLiteral | Digit+
```

5.3. TAPL Result Binding

The result of TAPL evaluation MUST be bound to:

* the exact multi-root evidence set being appraised;
* the policy digest used for appraisal;
* the freshness context used to reject replay; and
* the application event identifier.

A TAPL outcome without that binding MUST NOT be reused as if it were current authorization state.

6. Canonical Attestation Record (CAR)

6.1. Role of CAR

CAR is an audit-oriented envelope for multi-root appraisal. It is not itself a new RATS role and it does not replace Evidence or Attestation Results. Its purpose is to bind together:

* references to the Evidence appraised,
* synthesized appraisal outputs,
* policy identity,
* freshness context, and
* replay-verification metadata.

6.2. Conceptual Structure

```json
{
  "metadata": {
    "car_version": "1.0",
    "event_id": "uuid",
    "profile_id": "L2",
    "issued_at": "2026-03-19T12:00:00Z"
  },
  "attestation_evidence": [
    {
      "root_id": "root_a",
      "evidence_digest": "sha-384:...",
      "evidence_type": "evidence-reference"
    },
    {
      "root_id": "root_b",
      "evidence_digest": "sha-384:...",
      "evidence_type": "evidence-reference"
    }
  ],
  "policy_evaluation": {
    "tapl_policy_digest": "sha-384:...",
    "synthesized_decision": "ALLOW",
    "failing_predicates": []
  },
  "independence_metrics": {
    "vendor_separation": true,
    "crypto_stack_separation": true,
    "control_plane_separation": true
  },
  "freshness": {
    "nonce": "base64url...",
    "evidence_window": "session-epoch-123"
  },
  "semantic_binding": {
```

```
    "subject_digest": "sha-384:...",
    "binding_type": "application-event"
  },
  "semantic_consensus": {
    "consistency_check": "passed",
    "checked_properties": ["model_version", "input_schema"]
  },
  "cryptographic_binding": {
    "car_digest": "sha-384:...",
    "timestamp_token_digest": "sha-384:...",
    "signer": "verifier.example"
  }
}
'''
```

6.3. CAR Processing Rules

A CAR emitted under this profile MUST:

* identify the specific evidence set that was appraised;

* identify the policy digest used to synthesize the result;

* identify freshness inputs sufficient to reject stale reuse;

* identify the application event to which the result applies; and

* include cryptographic binding sufficient to detect tampering after issuance.

7. Relationship to RFC 9334 and EAT

7.1. Relationship to the RATS Architecture

This profile is designed to fit within the RATS architecture described by RFC 9334.

In particular:

* Attesters continue to produce Evidence;

* Verifiers continue to appraise Evidence and produce Attestation Results;

* Relying Parties continue to consume Attestation Results; and

* TAPL is profile-specific Appraisal Policy logic for multi-root synthesis.

This document does not redefine those roles.

7.2. Relationship to EAT

This profile does not require a new token format. EAT-based claims can be used where deployments already use EAT to encode Evidence, Attestation Results, or related claims.

CAR is therefore best understood as a higher-layer audit and replay envelope. It can carry references to EAT-based artifacts, or to other evidence and result encodings, together with the policy digest, freshness context, and application-event binding needed for multi-root replay.

8. CAR-to-RATS Mapping

| CAR field | RATS / EAT concept

Role primarily responsible	Purpose
‘attestation_evidence[]‘	Evidence or references to Evidence	Attester, Verifier	Identifies the evidence set appraised for the authorization event
‘policy_evaluation.synthesized_decision‘	Attestation Results outcome	Verifier	Captures the deterministic result of multi-root appraisal
‘policy_evaluation.tapl_policy_digest‘	Appraisal Policy identifier or digest	Verifier or policy service	Shows which policy instance produced the result
‘independence_metrics.*‘	Application-specific appraisal predicates	Verifier or policy service	Records whether cross-root separation predicates passed at appraisal time
‘freshness.*‘	Freshness inputs associated with appraisal	Attester, Verifier	Supports replay rejection and same-session binding
‘semantic_binding.*‘	Application-specific claim binding	Verifier or application policy service	Binds the appraisal to the specific inference event being authorized
‘semantic_consensus.*‘	Application-specific consistency checks	Verifier or application policy service	Records whether cross-root semantic consistency checks passed
‘cryptographic_binding.*‘	Integrity protection around evidence references and result package	Verifier	Detects tampering after issuance

9. Security Considerations

A deployment using this profile MUST ensure that synthesized appraisal outputs cannot be detached from the specific evidence set, policy digest, freshness context, and application event to which they apply.

In particular:

* replayed evidence from an earlier authorization event MUST be rejected;

* mixed-session evidence sets MUST be rejected before synthesis;

* quote or evidence substitution after appraisal MUST be detectable;

* the freshness window for synthesis MUST be explicit;

* policy drift MUST be detectable through policy identity or digest; and

* any timestamp token used for long-term replay MUST be bound to the exact CAR digest being archived.

Where the profile carries application-specific semantic bindings, the binding SHOULD use digests or references rather than raw application payloads unless disclosure is operationally required.

10. Privacy Considerations

This profile SHOULD minimize disclosure of application-specific data. A CAR need not contain raw Evidence bytes, raw inference inputs, or raw model outputs when references or digests are sufficient for replay and audit.

Deployments using EAT-based claims or equivalent encodings SHOULD disclose only those claims required by the Relying Party’s policy.

11. IANA Considerations

This document has no IANA actions.

12. References

12.1. Normative References

[RFC9334] Birkholz, H., Thaler, D., Richardson, M., Tsao, T., and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", RFC 9334, January 2023.

12.2. Informative References

[RFC3161] Adams, C., Cain, P., Pinkas, D., and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", RFC 3161, August 2001.

Author’s Address

Erkan Yalcinkaya
Luminesce Limited
United Kingdom
Email: yalcinkaya@luminescelimited.com
URI: https://luminescelimited.com