Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 22 November 2026

D. Lu
Futurewei

May 2026

Federated Infrastructure Layer for Cross-Platform Device and Application Attestation

draft-lu-rats-federated-attestation-infrastructure-00

Abstract

This document defines a federated infrastructure layer for cross-platform device and application attestation.

The proposed architecture extends the Remote ATtestation ProcedureS (RATS) architecture [RFC9334] by introducing privacy-preserving federation, decentralized endorsement distribution, decentralized normalization of attestation claims, and software supply-chain provenance integration.

The architecture enables interoperable attestation across heterogeneous operating systems, application ecosystems, and trust providers without requiring centralized platform ownership.

This document defines architectural roles, protocol interactions, endorsement distribution semantics, provenance integration models, and privacy requirements.

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 2 November 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

Remote ATtestation procedureS (RATS) Architecture [RFC9334] defines a generalized architecture for remote attestation systems. However, existing attestation deployments remain highly platform-centric and rely on vertically integrated trust infrastructures.

Examples include proprietary mobile integrity services, hardware vendor-specific endorsement chains, cloud-provider-specific attestation APIs, and disconnected software supply-chain verification systems.

These models create several limitations including fragmentation across ecosystems, inability to interoperate across trust domains, centralized endorsement bottlenecks, privacy leakage, opaque claim semantics, and weak integration with software provenance systems.

This document defines a Federated Infrastructure Layer for cross-platform attestation interoperability.

The architecture enables interoperable attestation verification, decentralized trust distribution, portable claim semantics, composable trust evaluation, and provenance-aware application integrity assessment.

2. 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].

2.1. Federated Verifier

A verifier capable of consuming attestation evidence from multiple heterogeneous attestation providers.

2.2. Claim Normalizer

A logical role that transforms platform-specific attestation claims into normalized semantic claims.

2.3. Endorsement Distributor

A decentralized service responsible for distributing endorsements, reference values, and trust anchors.

2.4. Provenance Authority

An authority responsible for validating software supply-chain provenance metadata.

3. Requirements

3.1. R1: Cross-Platform Interoperability

The system MUST support heterogeneous attestation providers and operating systems.

3.2. R2: Decentralized Trust Distribution

The architecture MUST avoid mandatory centralized endorsement authorities.

3.3. R3: Privacy Preservation

The architecture MUST minimize device-identifying information leakage.

3.4. R4: Claim Portability

The architecture MUST support normalized semantic claims across ecosystems.

3.5. R5: Provenance Awareness

The architecture MUST support integration with software supply-chain provenance systems.

3.6. R6: Federated Extensibility

The architecture MUST permit independently governed attestation domains.

3.7. Discussion

The requirements defined in this section are motivated by increasing fragmentation in modern attestation ecosystems. Current industry deployments are typically vertically integrated and optimized for a single vendor trust domain.

Mobile ecosystems commonly bind attestation semantics to proprietary platform APIs, cloud providers expose provider-specific confidential computing attestation formats, and enterprise endpoint systems frequently rely on closed posture-assessment protocols.

Cross-platform interoperability is therefore required to support heterogeneous device fleets and multi-cloud execution environments.

Decentralized trust distribution is motivated by operational and geopolitical concerns surrounding centralized trust anchors.

Privacy preservation reflects growing industry movement toward minimizing globally correlatable identifiers.

Claim portability is motivated by the operational burden created by incompatible attestation semantics.

Provenance awareness reflects the growing importance of software supply-chain security following widespread adoption of SBOMs, SLSA, Sigstore, and in-toto frameworks.

4. Architectural Overview

4.1. Core Roles

* Attester
* Verifier
* Relying Party

* Reference Value Provider

4.2. Extended Roles

* Federated Claim Normalizer

* Endorsement Distributor

* Provenance Authority

* Federation Coordinator

5. Federated Trust Architecture

5.1. Trust Domains

Each ecosystem MAY operate an independent trust domain.

Examples include mobile platform vendors, cloud providers, enterprise PKIs, open-source foundations, and hardware manufacturers.

5.2. Cross-Domain Verification

Federated verifiers MAY validate evidence originating from multiple trust domains.

Trust relationships MAY be established through cross-signing, transparency logs, policy registries, threshold trust models, and decentralized trust registries.

5.3. Trust Portability

A relying party SHOULD be able to evaluate trust equivalence between ecosystems using normalized semantic claims.

5.4. Discussion

The federated trust model defined in this section is motivated by the operational reality that modern computing environments are inherently multi-domain and multi-stakeholder.

Current industry practice already demonstrates partial federation models in adjacent domains including federated identity systems, public key infrastructures, certificate transparency ecosystems, and software package trust systems.

Trust domains are introduced to enable independently governed attestation ecosystems while still supporting interoperable verification.

Cross-domain verification is motivated by real-world scenarios where relying parties must assess devices or workloads originating outside their direct ecosystem.

Trust portability addresses one of the largest operational gaps in current attestation deployments: the inability to express security intent independently of vendor implementation.

6. Privacy-Preserving Federation

6.1. Selective Disclosure

Attesters SHOULD support selective disclosure of claims.

6.2. Minimization of Persistent Identifiers

Attestation evidence MUST NOT require globally stable device identifiers.

6.3. Verifier Isolation

Federated verifiers SHOULD be isolated from ecosystem-global tracking systems.

6.4. Discussion

Privacy-preserving federation is motivated by longstanding concerns that attestation systems can become de facto device tracking infrastructures.

Industry practice increasingly recognizes that attestation must balance trust establishment with user privacy.

Selective disclosure mechanisms are motivated by the principle of least privilege.

Minimization of persistent identifiers reflects broader industry migration away from stable hardware identity exposure.

Verifier isolation is motivated by concerns that centralized verification infrastructures can aggregate cross-service behavioral data.

7. Decentralized Endorsement Distribution

7.1. Distributed Endorsement Registries

Endorsements MAY be distributed using transparency logs, content-addressable storage, distributed ledgers, and signed metadata registries.

7.2. Endorsement Objects

An endorsement object MAY contain hardware model metadata, firmware measurements, OS build metadata, application signing metadata, revocation status, and provenance bindings.

7.3. Discussion

Decentralized endorsement distribution is motivated by operational scalability limitations and governance concerns associated with centralized endorsement infrastructures.

Industry practice increasingly favors distributed trust metadata systems such as certificate transparency logs, decentralized software repositories, and software transparency systems.

Distributed endorsement registries reduce single points of operational failure.

Transparency requirements reflect industry movement toward verifiable infrastructure governance.

Multi-authority endorsements recognize that no single entity fully controls modern computing stacks.

8. Decentralized Claim Normalization

8.1. Normalized Semantic Claims

Normalized claims abstract vendor-specific representations into interoperable semantics.

Normalized Claim	Vendor-Specific Sources
oemboot	Android Verified Boot, Apple Secure Boot
hw_backed_keys	StrongBox, Secure Enclave
app_integrity	Play Integrity, App Attest

Table 1: Example Normalized Claims

8.2. Discussion

Decentralized claim normalization is motivated by incompatible claim semantics across vendors and ecosystems.

Different ecosystems expose similar security properties using incompatible terminology, evidence structures, confidence models, and verification semantics.

Normalized semantic claims abstract implementation details into policy-relevant security properties.

Layered confidence models reflect the operational reality that attestation strength varies significantly depending on implementation characteristics.

9. Supply Chain Provenance Integration

9.1. Motivation

Traditional attestation systems validate runtime integrity but often ignore software provenance.

9.2. Provenance Sources

The architecture MAY integrate SBOM systems, SLSA provenance, in-toto attestations, Sigstore transparency records, reproducible build verification, and package registry signatures.

9.3. Discussion

Supply-chain provenance integration is motivated by the growing recognition that runtime integrity alone is insufficient for establishing trustworthy software execution.

Recent industry incidents involving compromised build systems, malicious dependencies, package repository attacks, and signed malware have demonstrated that valid runtime attestation may coexist with compromised software provenance.

Industry practice has increasingly shifted toward end-to-end software supply-chain verification.

Provenance binding enables attestation systems to connect runtime measurements with software lifecycle evidence.

Composite trust evaluation reflects common enterprise security practice where trust decisions incorporate multiple dimensions simultaneously.

10. Composite Attestation Results

10.1. Multi-Source Aggregation

Composite results MAY combine hardware attestation, OS attestation, application integrity, provenance verification, and enterprise posture assessment.

10.2. Policy Evaluation

Relying parties MAY define policies over normalized claims.

10.3. Discussion

Composite attestation results are motivated by the operational reality that modern trust evaluation is inherently multi-dimensional and risk-based.

Current industry practice increasingly combines multiple security signals into unified policy engines.

Multi-source aggregation recognizes that no individual attestation mechanism provides complete assurance.

Federated trust scoring reflects broader industry adoption of probabilistic and weighted trust models.

11. Security Considerations

The architecture introduces security considerations including federation poisoning, normalization ambiguity, transparency log compromise, and provenance forgery.

12. Privacy Considerations

The architecture prioritizes correlation resistance, unlinkability, metadata minimization, and privacy-preserving transparency mechanisms.

13. IANA Considerations

This document requests no IANA actions.

14. 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>.

[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>.

[RFC9711] Lundblade, L., Mandyam, G., O'Donoghue, J., and C. Wallace, "The Entity Attestation Token (EAT)", RFC 9711, DOI 10.17487/RFC9711, April 2025, <https://www.rfc-editor.org/info/rfc9711>.

Appendix A. Example Claim Normalization Schema

```
{
"normalized_claim": "oemboot",
"confidence_level": 3,
"source_claims": [
{
"vendor": "android",
"claim": "verifiedBootState",
"value": "TRUE"
}
]
}
```

Appendix B. Example Provenance Binding

{
"artifact_hash": "sha256:abcd...",
"sbom_reference": "urn:sbom:12345",
"slsa_attestation": "urn:slsa:67890",
"transparency_log_entry": "urn:tlog:99999"
}

Appendix C. Threat Model

The architecture assumes adversaries MAY attempt endorsement forgery, verifier compromise, federation poisoning, provenance tampering, privacy correlation attacks, and semantic claim manipulation.

The architecture is specifically designed to reduce centralized trust concentration, ecosystem lock-in, opaque trust semantics, verifier overreach, and provenance fragmentation.

Author's Address

David Lu
Futurewei
Email: dlu@futurewei.com