Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: 12 July 2026

T. Kamimura
VeritasChain Standards Organization

8 January 2026

Verifiable AI Provenance Framework (VAP): An Architectural Framework for Evidentiary-Grade AI Decision Trails
draft-kamimura-vap-framework-00

Abstract

Automated decision-making systems, including AI and algorithmic systems in critical infrastructure, currently lack standardized mechanisms for producing evidentiary-grade provenance records that can withstand independent verification. Traditional logging approaches fail to provide the cryptographic guarantees required for regulatory compliance, forensic investigation, and cross-organizational accountability.

This document describes the Verifiable AI Provenance Framework (VAP), an architectural framework that defines requirements for producing verifiable decision trails using existing IETF security technologies. VAP does not define new protocols or cryptographic primitives; rather, it provides an architectural coordination layer that enables domain-specific profiles to leverage Supply Chain Integrity, Transparency and Trust (SCITT), Remote Attestation Procedures (RATS), CBOR Object Signing and Encryption (COSE), and related IETF work in a consistent manner.

This document is intended to frame the problem space and facilitate discussion about whether architectural coordination work is needed in this area.

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 12 July 2026.

Table of Contents

1. Introduction

The deployment of automated decision-making systems across critical infrastructure has created an urgent need for evidentiary-grade provenance records. These systems--including AI-driven trading algorithms, autonomous vehicle controllers, medical diagnostic systems, and public administration scoring systems--make decisions that significantly impact human welfare and require robust audit trails for regulatory compliance, forensic analysis, and accountability determination.

Current approaches to logging automated system behavior typically rely on traditional audit logs that lack the cryptographic properties necessary to serve as reliable evidence. While existing IETF technologies provide many of the necessary building blocks, there is no established architectural framework for applying these technologies consistently across different domains.

This document proposes a unifying framework that coordinates existing IETF security building blocks--specifically SCITT for transparency infrastructure, RATS for environment attestation, and COSE for cryptographic operations--to achieve tamper-evident and provably complete audit trails for high-impact AI and automated systems.

This document describes the Verifiable AI Provenance Framework (VAP), which aims to address this gap by defining architectural requirements for evidentiary-grade provenance while leveraging existing IETF work wherever possible.

VAP is positioned as a specialized evidentiary and forensic layer focused on producing cryptographically verifiable decision trails. It is not intended to be a comprehensive AI governance protocol; rather, it addresses the specific problem of how to record and verify what decisions an automated system made, when, and based on what inputs. Broader governance concerns such as authorization policies, risk classification, and real-time approval workflows are outside VAP’s scope but may consume VAP provenance data.

1.1. Why Traditional Logs Are Insufficient

Traditional logging mechanisms, including syslog [RFC5424], database audit trails, and application-level logs, suffer from several fundamental limitations when used as evidence of automated system behavior:

Mutability: Log entries can be modified or deleted without detection, undermining their evidentiary value.

Ordering Uncertainty: Without cryptographic chaining, the sequence of events cannot be independently verified; entries may be inserted, reordered, or removed.

No Completeness Proof: Traditional logs provide no mechanism to prove that no entries have been omitted between any two recorded events.

Single-Party Control: Logs typically reside under the sole control of the system operator, requiring auditors to trust the operator's integrity.

No Independent Verification: Third parties cannot verify log integrity without trusting the log-producing entity.

These limitations render traditional logs insufficient for regulatory evidence, forensic reconstruction, or cross-organizational accountability determination.

1.2. Cross-Sector Applicability

While particular implementations may focus on specific sectors, the underlying requirement for verifiable decision provenance is domain-agnostic. The same fundamental properties--non-repudiation, completeness verification, and independent auditability--are required across multiple sectors:

Financial Services: Algorithmic trading systems subject to regulations such as MiFID II RTS 25 and SEC Rule 17a-4.

Healthcare: AI diagnostic systems subject to FDA medical device regulations (21 CFR Part 11) and HIPAA requirements.

Transportation: Autonomous vehicle systems requiring incident reconstruction capabilities per UN Regulation 157.

Public Administration: Automated scoring and decision systems subject to administrative procedure requirements and GDPR Article 22 (automated individual decision-making).

Energy Infrastructure: AI-driven grid management systems subject to critical infrastructure regulations including NIS2 Directive.

This cross-sector relevance motivates an architectural framework approach rather than domain-specific point solutions.

1.3. Regulatory Context

Several regulatory frameworks are creating immediate demand for technical standards addressing AI system auditability:

1.3.1. EU AI Act Article 12

The European Union's AI Act [EU-AI-ACT] Article 12 establishes logging requirements for high-risk AI systems that become enforceable on 2 August 2026. These requirements include:

* Automatic event recording over the system's lifetime
* Traceability for risk identification, post-market monitoring, and operational oversight
* Retention periods of minimum 6 months (longer if required by sector-specific law)
* Tamper-evident storage preventing log manipulation
* Logging of human intervention and manual overrides with attribution

Currently, no IETF RFCs directly address these logging requirements. VAP aims to provide protocol-level specifications that complement the requirements framework being developed by ISO/IEC and CEN-CENELEC JTC 21.

1.3.2. Other Regulatory Drivers

Additional regulatory frameworks creating demand for verifiable AI provenance include:

* MiFID II/III RTS 25 (algorithmic trading record-keeping)
* SEC Rule 17a-4 (broker-dealer record retention)
* NIST AI RMF 1.0 [NIST-AI-RMF] (AI risk management framework)
* ISO/IEC 42001:2023 [ISO-42001] (AI management systems)
* ISO/IEC DIS 24970 (AI system logging - in development)

1.4. Scope of This Document

This document:

* Describes the problem space and requirements for evidentiary-grade AI/automated system provenance
* Proposes an architectural framework (VAP) for addressing these requirements
* Maps VAP concepts to existing IETF technologies
* Discusses the relationship between the framework and domain-specific profiles

This document does not:

* Define new cryptographic primitives or protocols
* Specify wire formats or APIs
* Mandate particular implementations or deployment models
* Address the verification of AI model correctness or fairness
* Define comprehensive AI governance protocols (authorization, risk classification, approval workflows)

1.4.1. Relationship to Other IETF AI Work

It is important to distinguish VAP from other AI-related work in the IETF:

AIPREF (AI Preferences): The AIPREF working group addresses how content owners express preferences about AI training data collection (e.g., robots.txt extensions). VAP does not address training data provenance; rather, it focuses on the runtime decision-making of deployed AI systems. AIPREF governs what data AI may learn from; VAP records what decisions AI systems make after deployment.

WIMSE (Workload Identity in Multi-System Environments): The WIMSE working group addresses identity and authentication for workloads in cloud-native environments. VAP profiles may leverage WIMSE for AI agent identity management and signing key issuance, treating WIMSE as complementary infrastructure rather than overlapping scope.

1.5. 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.

Note that this document primarily uses descriptive rather than normative language, as it describes an architectural framework rather than a protocol specification.

2. Problem Statement

This section describes the specific gaps in current approaches to automated system auditability that motivate the VAP framework.

2.1. Non-Repudiation Gaps

Non-repudiation requires that an entity cannot deny having performed an action. Current audit approaches often fail to provide this guarantee because:

* Logs may not be cryptographically signed by the acting entity.

* Signatures, when present, may use keys that are not adequately protected or attested.

* The relationship between the signing key and the claimed identity may not be independently verifiable.

For automated systems, non-repudiation is particularly challenging because the "actor" may be an algorithm rather than a human, requiring clear identification of the specific system version, configuration, and operational parameters that produced a given decision.

The Entity Attestation Token (EAT) [RFC9711] provides a foundation for addressing these challenges through standardized claims for device and entity identification, software measurements, and freshness guarantees.

2.2. Integrity Without Completeness

Many existing approaches can demonstrate that individual log entries have not been modified (integrity), but cannot prove that no entries have been omitted (completeness). This gap is critical because:

* An adversary can selectively delete unfavorable entries while preserving the integrity of remaining entries.

* Auditors cannot distinguish between "no events occurred during period X" and "events were deleted from period X".

* Regulatory requirements often mandate complete audit trails, not merely tamper-evident individual entries.

The Certificate Transparency model [RFC6962] demonstrated that completeness can be addressed through append-only logs with Merkle tree commitments, but this pattern has not been widely applied to automated system audit trails. See Appendix A for lessons learned.

Specifically, Merkle tree-based Receipts enable two critical proofs:

* Inclusion Proof: Proves a specific record exists in the log at a given tree state.

* Consistency Proof: Proves that a later tree state is an append-only extension of an earlier state--no records were removed or modified between the two states.

By requiring both proof types, VAP enables detection of attempts to selectively omit records. Sequential record identifiers (e.g., monotonic sequence numbers or UUID v7 timestamps) provide additional gap detection capability.

2.3. Missing Responsibility Boundaries

Automated systems frequently operate across organizational boundaries, creating uncertainty about responsibility allocation:

* An AI system may receive inputs from multiple external sources whose integrity cannot be independently verified.

* Decisions may be influenced by models trained on data from various parties.

* The chain of custody for decision inputs may span multiple organizations with different trust levels.

Current logging approaches typically capture only the local view of a single system, without cryptographic binding to the provenance of inputs or the responsibilities of other parties in the decision chain.

Note: RFC 9334 Errata 7314 identifies terminology ambiguities regarding entity and role definitions in multi-party scenarios. VAP profiles should carefully define responsibility boundaries with explicit reference to this errata when mapping to RATS concepts.

2.4. Inability to Independently Verify

Perhaps the most fundamental gap is that current approaches require trust in the log-producing entity. An auditor examining a traditional audit trail must trust that:

* The logging system was correctly implemented.

* The operator did not modify logs after the fact.

* The presented logs represent the complete record.

* Timestamps accurately reflect when events occurred.

This trust requirement is particularly problematic when the entity being audited has an incentive to present a favorable but inaccurate record, which is precisely when audit trails are most needed.

2.5. Summary of Gaps

The following table summarizes the gaps that VAP addresses:

Gap	Traditional Approach	VAP Requirement
Non-repudiation	Trust-based	Cryptographic signatures with attested keys
Completeness	Not addressed	Merkle tree proofs
Cross-org accountability	Point-to-point	Verifiable chains
Independent verification	Trust operator	External anchoring

Table 1: Summary of Gaps in Current Approaches

3. Design Goals

This section describes the high-level design goals that guide the VAP framework.

3.1. Verifiability Over Trust

The fundamental principle of VAP is "Verify, Don’t Trust." Rather than requiring auditors to trust log-producing entities, VAP enables mathematical verification of:

* Individual record integrity (the record has not been modified).

* Record completeness (no records have been omitted).

* Temporal ordering (records appear in the correct sequence).

* Actor attribution (the claimed entity produced the record).

This shift from trust to verification is essential for audit scenarios where the audited entity may have incentives to present inaccurate records.

3.2. Cryptographic Completeness

VAP requires that completeness be cryptographically verifiable. This means:

* Auditors can prove that they have received all records, not just a selected subset.

* Gaps in the record sequence are detectable.

* Historical states of the log can be reconstructed and verified.

This goes beyond traditional integrity (individual records are unmodified) to ensure that the entire audit trail is complete and verifiable.

3.3. Independent Verification

VAP should enable verification without requiring cooperation from the entity being audited. This requires:

* External anchoring of commitments to independent third parties.

* Standard formats that any verifier can process.

* Public or shared verification infrastructure where appropriate.

3.4. Accountability Across Organizational Boundaries

Automated systems increasingly operate in environments where:

* Inputs come from external parties whose integrity matters.

* Outputs are consumed by downstream systems operated by others.

* Multiple organizations may share responsibility for a decision.

VAP should support clear accountability boundaries by enabling:

* Cryptographic binding of inputs to their sources.

* Clear delineation of which entity is responsible for each assertion.

* Verification that does not require trust in intermediate parties.

3.5. Domain Agnosticism

While particular applications of VAP may be domain-specific, the framework itself should be domain-agnostic:

* Core concepts should apply across different sectors (finance, healthcare, transportation, etc.).

* Domain-specific requirements should be addressed through profiles that extend the base framework.

* The framework should not mandate domain-specific data formats or semantics.

This approach enables reuse of verification infrastructure across domains while allowing each domain to define its specific requirements.

4. VAP Architecture Overview

This section describes the conceptual architecture of VAP. This is an architectural framework, not a protocol specification; actual implementations would realize these concepts using specific technologies such as those described in Section 5.

4.1. Layered Architecture

VAP defines three conceptual layers:

Integrity Layer: Provides tamper-evidence for individual records and the overall log structure. This layer addresses per-record integrity through cryptographic hashing, forward chaining to create append-only structure, periodic commitments (e.g., Merkle roots) for efficient verification, and external anchoring for independent verification.

Provenance Layer: Captures the "what, when, who, why" of each decision. This layer defines actor identification (which system/ model produced the decision), input provenance (what data informed the decision), decision context (parameters, constraints, environmental factors), and output characterization (what was decided/recommended/ executed).

Accountability Layer: Enables responsibility assignment and cross-organizational verification. This layer provides responsibility boundaries between parties, verification interfaces for external auditors, evidence packaging for regulatory submission, and cross-reference capabilities for multi-party scenarios.

These layers are conceptual; actual implementations may combine them in various ways depending on deployment requirements.

4.2. Relationship Between Framework and Profiles

VAP is designed as a framework that supports domain-specific profiles. The framework defines:

* Common requirements that apply across domains.

* Mapping to underlying IETF technologies.

* Extension points where profiles can add domain-specific requirements.

Profiles define:

* Domain-specific event schemas.

* Regulatory mapping for the specific domain.

* Conformance requirements appropriate to the domain.

* Any domain-specific extensions to the base framework.

This separation allows the framework to remain stable while profiles evolve to meet changing domain requirements.

4.3. Cryptographic Agility

VAP systems should support cryptographic agility, meaning the ability to:

* Update cryptographic algorithms without protocol changes.

* Support multiple algorithms during transition periods.

* Clearly document the security properties provided by chosen algorithms.

This approach, often called "crypto-agility," is essential given the ongoing evolution of cryptographic best practices and the transition to post-quantum algorithms.

4.3.1. Post-Quantum Cryptography Considerations

NIST has standardized three post-quantum cryptographic algorithms in August 2024:

* FIPS 203 [FIPS-203] (ML-KEM): Module-Lattice-Based Key-Encapsulation Mechanism

* FIPS 204 [FIPS-204] (ML-DSA): Module-Lattice-Based Digital Signature Algorithm

* FIPS 205 [FIPS-205] (SLH-DSA): Stateless Hash-Based Digital Signature Algorithm

For long-lived AI provenance records that may need to remain verifiable for decades, VAP profiles should consider post-quantum signature algorithms. The COSE working group is standardizing ML-DSA for COSE [I-D.ietf-cose-dilithium], which registers:

* ML-DSA-44 (COSE algorithm value -48): Security level 2

* ML-DSA-65 (COSE algorithm value -49): Security level 3

* ML-DSA-87 (COSE algorithm value -50): Security level 5

Profiles may mandate specific algorithms for interoperability within a domain, but should do so in a way that allows future algorithm updates.

5. Mapping to Existing IETF Work

A key principle of VAP is to leverage existing IETF technologies wherever possible rather than defining new protocols or formats. This section describes how VAP concepts map to ongoing IETF work.

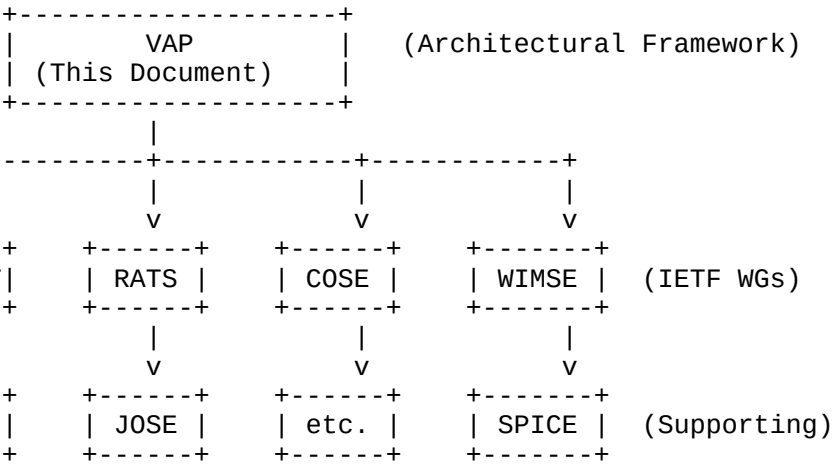

Figure 1: VAP Relationship to IETF Working Groups

5.1. SCITT (Supply Chain Integrity, Transparency and Trust)

The SCITT working group [I-D.ietf-scitt-architecture] provides the most direct mapping to VAP's Integrity Layer requirements.

5.1.1. Concept Mapping

VAP Concept	SCITT Equivalent	Notes
Provenance Record	Signed Statement	COSE_Sign1 with CWT claims per RFC 9597
Append-Only Log	Transparency Service	Maintains ordered log via VDS
Merkle Commitment	Receipt	COSE Receipt per Merkle Tree Proofs
Verification Policy	Registration Policy	Admission criteria
External Anchor	Merkle Tree Root	Published via

		Signed Tree Head
Transparent Record	Transparent Statement	Statement + Receipt

Table 2: VAP to SCITT Concept Mapping

5.1.2. SCITT Charter Scope Considerations

The SCITT WG charter explicitly scopes work to "software supply chain" and "firmware." However, the charter also notes that SCITT is "applicable to any other type of supply chain statements." VAP positions AI provenance as content-agnostic payload within SCITT's Signed Statements, which is explicitly supported by the architecture's design.

Profiles that wish to use SCITT infrastructure should:

* Frame provenance records as supply chain statements about AI system behavior
* Use standard SCITT Registration Policies for admission control
* Leverage existing Transparency Service implementations where available

5.1.3. Key SCITT Terminology

VAP profiles using SCITT should adopt the following terminology from [I-D.ietf-scitt-architecture]:

* Transparency Service: The service operating the append-only log
* Signed Statement: A COSE_Sign1 signed object containing the provenance payload
* Receipt: A COSE Receipt proving inclusion in the log
* Transparent Statement: A Signed Statement combined with its Receipt
* Registration Policy: Rules governing what statements are accepted
* Verifiable Data Structure (VDS): The cryptographic structure (typically Merkle tree) underlying the log

5.1.4. External Anchor Requirements

While SCITT provides robust tamper-evidence through Merkle tree commitments, VAP profiles should consider requiring periodic External Anchoring--the publication of Merkle Tree roots to independent, external systems (e.g., public blockchains, trusted timestamping services, or Certificate Transparency logs).

External Anchoring addresses a specific threat: a malicious Transparency Service operator who might attempt to rewrite history by reconstructing the entire log with altered content. By periodically anchoring the Merkle root to external systems beyond the operator's control, VAP ensures that:

* Historical log states are committed to third-party systems
* Any attempt to reconstruct a different history would produce Merkle roots inconsistent with previously anchored values
* Regulators and auditors can verify that logs have not been retroactively altered

This requirement emerged from domain profile experience (VCP v1.1) where the "Verify, Don't Trust" principle demanded cryptographic guarantees even against collusion between system operators and Transparency Service providers. Profiles may specify anchoring frequency (e.g., hourly, daily) based on domain-specific risk assessment.

5.1.5. Alignment Approach

VAP profiles that address the Integrity Layer requirements should:

* Encode provenance records as SCITT Signed Statements where applicable.
* Include CWT Claims (COSE header parameter 15) with at minimum 'iss' (issuer) and 'sub' (subject) claims per [RFC9597].
* Use SCITT Transparency Services for append-only log maintenance.
* Leverage COSE Receipts per [I-D.ietf-cose-merkle-tree-proofs] for inclusion proofs, using the registered COSE header parameters: receipts (394), vds (395), and vdp (396).
* Define Registration Policies appropriate to the domain.

This alignment allows VAP-compliant systems to benefit from the SCITT ecosystem, including existing Transparency Service implementations and verification tooling.

5.2. Remote Attestation Procedures (RATS)

The RATS working group [RFC9334] addresses the question of how to verify the state of a system producing attestations. This is relevant to VAP's Accountability Layer.

Per RFC 9334, the RATS architecture defines specific roles and data types: an Attester produces Evidence (claims about its own state), a Verifier evaluates that Evidence and produces an Attestation Result, and a Relying Party consumes the Attestation Result to make trust decisions. VAP's cross-organizational verification scenarios map naturally to this model.

5.2.1. Concept Mapping

VAP Concept	RATS Equivalent	Notes
Actor Attestation	Attester	Entity producing provenance records
Environment Binding	Evidence	Claims about Attester state
Cross-Org Verify	Verifier	Third-party verification svc
Verification Result	Attestation Result	Verifier output for Relying Party
Auditor/Regulator	Relying Party	Consumes results for trust decision

Table 3: VAP to RATS Concept Mapping

5.2.2. Entity Attestation Token (EAT)

The Entity Attestation Token specification [RFC9711] provides attestation-oriented claims particularly relevant to AI provenance:

* Device/entity identification: 'ueid', 'sueid' claims for unique AI model or inference endpoint identification

* Software measurements: ‘manifests‘ and ‘measurements‘ claims for recording model hashes, weights, and dependencies

* Composite systems: ‘submods‘ claim for representing ensemble models or ML pipelines

* Freshness: ‘eat_nonce‘ ensuring non-replayable provenance records

* Jurisdictional compliance: ‘location‘ claim for capturing inference geography (relevant to data sovereignty requirements)

5.2.3. Additional RATS Documents

VAP profiles requiring strong attestation should also consider:

* Concise Reference Integrity Manifest (CoRIM) [I-D.ietf-rats-corim]: For model reference values

* EAT Attestation Results [I-D.fv-rats-ear]: Standardized format for attestation results

* Attestation Results for Secure Interactions (AR4SI) [I-D.ietf-rats-ar4si]: Trustworthiness vectors

5.2.4. Alignment Approach

RATS concepts are particularly relevant when VAP systems need to attest to the integrity of the log-producing environment itself:

* Entity Attestation Tokens (EAT) [RFC9711] can provide evidence about the execution environment.

* Hardware-rooted attestation can strengthen claims about system integrity.

* The RATS architecture provides a framework for reasoning about trust in the log-producing system.

VAP does not require RATS integration, but profiles may specify RATS usage for environments where stronger assurance about the log producer is needed.

5.3. COSE (CBOR Object Signing and Encryption)

COSE [RFC9052] provides the cryptographic message formats used by both SCITT and RATS. VAP systems that leverage these technologies will use COSE for:

* Digital signatures on provenance records (COSE_Sign1).

* CWT Claims in COSE headers per [RFC9597].

* Receipts/inclusion proofs (COSE Receipts per [I-D.ietf-cose-merkle-tree-proofs]).

* Encrypted payloads when confidentiality is required.

* Algorithm identification and crypto-agility.

5.3.1. Alignment Approach

VAP profiles should:

* Use COSE for all cryptographic operations where interoperability with SCITT/RATS is desired.

* Follow COSE algorithm recommendations and deprecation guidance.

* Leverage COSE's built-in algorithm agility for future migration, including transition to post-quantum algorithms per [I-D.ietf-cose-dilithium].

Profiles that need JSON-based formats may alternatively use JOSE [RFC7515] [RFC7516], though this limits interoperability with CBOR-based ecosystems.

5.4. CFRG (Crypto Forum Research Group)

The Crypto Forum Research Group (CFRG) provides cryptographic review and guidance that informs VAP's cryptographic requirements. Relevant CFRG work includes:

* Guidance on hash function selection for Merkle trees.

* Signature algorithm recommendations.

* Post-quantum cryptography considerations.

VAP does not directly depend on CFRG outputs, but profiles should reference CFRG recommendations when selecting cryptographic algorithms.

5.5. WIMSE (Workload Identity in Multi-System Environments)

The WIMSE working group addresses identity and authentication for workloads (processes, containers, serverless functions) in cloud-native and multi-system environments. For VAP, WIMSE is relevant to the question of how AI systems obtain and manage the signing keys used to authenticate provenance records.

5.5.1. Relevance to AI Agent Identity

AI systems operating as autonomous agents require identity credentials to sign provenance records. WIMSE provides mechanisms for:

* Workload identity issuance in containerized/cloud environments
* Credential lifecycle management
* Cross-organizational identity federation

The individual draft [I-D.ni-wimse-ai-agent-identity] specifically addresses AI agent identity within the WIMSE framework.

5.5.2. Alignment Approach

VAP does not define identity management mechanisms; instead, it assumes that actors (AI systems) possess valid signing credentials. Profiles may specify:

* WIMSE as the identity infrastructure for cloud-native AI deployments
* Integration points between WIMSE credential issuance and VAP signing requirements
* Trust model considerations when AI agents operate across organizational boundaries

This separation of concerns allows VAP to focus on provenance structure and verification, while delegating identity management to specialized infrastructure like WIMSE.

5.6. Relationship Summary

The following table summarizes how VAP layers map to IETF work:

VAP Layer	Primary IETF Work	Supporting Work
Integrity Layer	SCITT	COSE, Merkle Tree Proofs, CT (RFC 6962/9162)
Provenance Layer	(Domain-specific profiles)	JSON Schema, CBOR, W3C VC 2.0, C2PA
Accountability Layer	RATS, EAT (RFC 9711)	WIMSE, CoRIM, AR4SI, OAuth/OIDC

Table 4: VAP Layers to IETF Work Mapping

VAP is explicitly designed to avoid duplicating existing IETF work. Where existing standards adequately address a requirement, VAP profiles should reference those standards rather than defining alternatives.

6. Why a Framework is Needed

Given that existing IETF technologies provide many of the necessary building blocks, one might ask whether a framework document is needed. This section explains the value of architectural coordination.

The challenge is not the absence of suitable technologies, but rather:

* Ensuring consistent application of these technologies across domains.

* Providing a reference architecture that profile developers can follow.

* Identifying gaps that may require additional standardization.

* Enabling interoperability between systems in different domains that need to exchange provenance information.

6.1. Risk of Fragmentation

Without architectural coordination, there is a risk that different domains will develop incompatible approaches to AI/automated system provenance:

* Financial services might develop one approach.

* Healthcare might develop another incompatible approach.

* Automotive might develop yet another.

This fragmentation would:

* Prevent reuse of verification infrastructure across domains.

* Complicate cross-domain scenarios (e.g., AI systems that span finance and healthcare).

* Increase implementation costs by requiring domain-specific tools.

* Create confusion about what "verifiable provenance" means in different contexts.

A framework document can help prevent this fragmentation by establishing common vocabulary, requirements, and architectural patterns.

6.2. Profiles Alone Are Insufficient

One alternative to a framework would be to develop domain-specific profiles directly, without an overarching framework. This approach has drawbacks:

* Each profile would need to independently define common concepts.

* There would be no reference for ensuring profiles are compatible.

* Common security considerations would be repeated across profiles.

* It would be unclear how to develop profiles for new domains.

A framework provides a foundation that profile developers can build upon, ensuring consistency while allowing domain-specific customization.

6.3. Architectural Coordination Role

The value of VAP as a framework lies in its coordination role:

* Defining common requirements that all profiles should meet.

* Mapping those requirements to existing IETF technologies.

* Providing guidance on how to develop compliant profiles.

* Identifying areas where existing standards may be insufficient.

* Enabling discussion of cross-domain interoperability requirements.

This coordination role does not require VAP to define new protocols; rather, it provides the architectural context within which existing protocols are applied.

6.4. Relationship to Other AI Governance Work

VAP is specifically focused on evidentiary-grade decision trails--the problem of recording and verifying what decisions were made. It is not intended to address the full scope of AI governance, which includes concerns such as:

* Risk-based authorization and approval workflows

* Real-time governance decisions

* AI model evaluation and certification

* Federated governance authority networks

Other work addressing broader AI governance concerns may consume VAP provenance data as an input to governance decisions. VAP provides the forensic/evidentiary layer that such systems can build upon.

6.5. VAP as Bridge Between Static and Runtime Verification

Examining the existing IETF landscape reveals a gap that VAP addresses:

SCITT (Supply Chain): Excels at tracking static artifacts--built binaries, signed firmware, software bills of materials. It answers: "Was this the correct, untampered software?"

RATS (Attestation): Excels at capturing point-in-time system state--hardware configuration, OS integrity, TEE status. It answers: "Was the execution environment trustworthy at this moment?"

VAP (Decision Provenance): Captures the dynamic decisions made by verified software in verified environments. It answers: "What did the system actually decide, and can we prove it?"

The unique value of VAP lies in connecting these domains:

* A SCITT-verified AI model binary...

* ...running in a RATS-attested trusted environment...

* ...produces decisions that VAP records as evidentiary-grade provenance.

This "missing link" perspective explains why VAP warrants architectural coordination: it bridges two established IETF work streams to address a problem (dynamic decision accountability) that neither fully covers alone.

7. Relationship to Domain-Specific Profiles

VAP is designed to support domain-specific profiles that tailor the framework to particular use cases. This section describes how profiles relate to the framework, using existing work as an example.

7.1. Example: Financial Trading (VCP)

The VeritasChain Protocol (VCP) [I-D.kamimura-scitt-vcp] is a profile of VAP focused on AI-driven algorithmic trading systems. VCP demonstrates how a profile specializes the framework.

7.2. Potential Future Profiles

The framework is designed to accommodate profiles for various domain requirements. Examples of potential future profiles include:

Medical AI: Profiles for diagnostic AI systems, addressing HIPAA privacy requirements and FDA medical device regulations.

Autonomous Vehicles: Profiles for driving decision systems, addressing incident reconstruction and liability determination.

Public Administration: Profiles for government scoring and recommendation systems, addressing administrative procedure requirements.

Energy Infrastructure: Profiles for grid management AI, addressing critical infrastructure protection requirements.

Each profile would define domain-specific event schemas, conformance requirements, and regulatory mappings while adhering to the common framework requirements.

7.3. Profile Independence

It is important to emphasize that:

* VAP does not depend on any particular profile.

* Profiles are developed independently by domain experts.

* A domain can adopt VAP without implementing profiles for other domains.

* The framework is complete without reference to any specific profile.

The mention of VCP or other potential profiles in this document is illustrative only. VAP's value lies in providing architectural coordination, not in mandating any particular domain application.

8. Security Considerations

As an architectural framework rather than a protocol specification, VAP's security considerations are primarily about ensuring that profiles and implementations address the relevant threats. This section describes the threat model and explicit non-goals.

8.1. Threat Model Overview

VAP addresses threats from entities who might seek to:

8.1.1. Falsify Historical Records

An operator might attempt to modify, delete, or fabricate historical records to conceal problematic behavior or create false evidence. VAP addresses this through:

* Cryptographic chaining that makes modification detectable.

* External anchoring that prevents history rewriting.

* Independent verification that does not require operator cooperation.

8.1.2. Omit Unfavorable Records

An operator might attempt to selectively omit records that reveal problematic behavior while preserving favorable records. VAP addresses this through:

* Completeness verification via Merkle proofs.

* Sequence numbering that reveals gaps.

* External commitments that prove log state at specific times.

8.1.3. Misattribute Actions

An operator might attempt to attribute actions to a different actor (algorithm, operator, or external party) than the one actually responsible. VAP addresses this through:

* Digital signatures binding actions to specific actors.

* Identity verification through established PKI or attestation mechanisms.

* Clear provenance chains for multi-party scenarios.

8.1.4. Manipulate Timing

An operator might attempt to manipulate timestamps to change the apparent sequence of events. VAP addresses this through:

* Cryptographic chaining that establishes relative ordering.

* External anchoring that establishes absolute time bounds.

* Trusted timestamping services where stronger guarantees are needed.

8.2. Explicit Non-Goals

VAP does not address certain security concerns that are outside its scope:

AI Model Security: VAP records what decisions were made, not whether those decisions were correct or whether the AI model is secure against adversarial attacks.

Input Validation: VAP can record what inputs were received, but does not validate whether inputs were legitimate or malicious.

Confidentiality: While profiles may address confidentiality, the framework focuses on integrity and verifiability rather than confidentiality. Profiles must address confidentiality requirements specific to their domain.

Real-Time Detection: VAP provides forensic and audit capabilities, not real-time anomaly detection or intrusion prevention. Real-time monitoring systems may use VAP data, but such systems are outside VAP’s scope.

8.3. Privacy Considerations

VAP provenance records may contain sensitive information about:

* Business activities (e.g., trading strategies)

* Personal data (e.g., in healthcare applications)

* Critical infrastructure operations

Profiles must address privacy requirements specific to their domain, which may include:

* Encryption of sensitive payload data.

* Access controls on Transparency Service queries.

* Crypto-shredding mechanisms for data subject to deletion rights (e.g., GDPR Article 17). This technique, corresponding to Cryptographic Erasure (CE) as described in NIST SP 800-88 Rev. 1 [NIST-SP-800-88], enables effective data deletion by destroying encryption keys rather than the encrypted data.

* Selective disclosure mechanisms that reveal only necessary information.

8.3.1. Reconciling Append-Only Logs with Deletion Rights

A fundamental tension exists between append-only transparency logs (where deletion is cryptographically prevented) and regulatory requirements for data deletion (e.g., GDPR Article 17 "right to erasure"). Profiles addressing domains with deletion requirements should consider:

* Off-chain data storage: Register only cryptographic hashes of sensitive payloads in the Transparency Service, storing actual data in separate systems where deletion is possible. This "hash-only registration" approach preserves verifiability-- anyone with the original data can verify it matches the registered hash--while keeping sensitive content out of the immutable log.

* Encryption with key management: Encrypt sensitive payload data before registration; deletion is achieved by destroying the encryption keys (crypto-shredding).

* Architectural separation: Design systems such that provenance metadata (which may need long-term retention) is separated from personal data (which may require deletion).

* Redacted proofs: For profiles requiring selective disclosure, consider cryptographic techniques (e.g., hash trees with selective revelation) that allow proving specific claims without revealing the complete record.

The framework itself does not mandate specific privacy mechanisms, but requires that profiles clearly document how privacy requirements are addressed.

9. IANA Considerations

This document has no IANA actions.

Domain-specific profiles may request IANA registrations for media types, COSE algorithm identifiers, or other parameters as needed. Such requests would be made in the profile documents, not in this framework document.

10. Next Steps

This document is intended to facilitate discussion about whether architectural coordination work is needed in the area of AI/automated system provenance.

10.1. Discussion Path

The author proposes the following discussion path:

1. Present this document for discussion at SecDispatch to determine the appropriate venue for further work.

2. Solicit feedback from the SCITT and RATS working groups on the proposed mappings to their work.

3. Gather input from domain experts on whether the framework adequately addresses their requirements.

4. Based on feedback, determine whether this work should proceed as individual submissions building on existing WGs (likely SCITT given closest alignment), be considered for a new focused work item, or be deferred pending additional implementation experience.

Given the EU AI Act Article 12 enforcement date of 2 August 2026, timely progress could position this work as a critical compliance enabler.

10.2. Potential Future Work

Depending on community interest, future work might include:

* Development of additional domain-specific profiles.

* Specification of interoperability requirements between profiles.

* Definition of common verification tooling requirements.

* Guidance on regulatory compliance mapping.

* Policy Identification mechanisms: Each domain profile may need to identify which operational policy or regulatory requirement governs a particular provenance record. This aligns with SCITT's Registration Policy concept and enables automated compliance verification. Future work could standardize policy identifier formats and registration procedures.

This document does not presuppose any particular outcome; it is intended to frame the problem and facilitate informed discussion.

11. References

11.1. 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>.

[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>.

[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>.

[RFC9597] Looker, T. and M.B. Jones, "CBOR Web Token (CWT) Claims in COSE Headers", RFC 9597, DOI 10.17487/RFC9597, June 2024, <https://www.rfc-editor.org/info/rfc9597>.

[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>.

11.2. Informative References

[RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, DOI 10.17487/RFC5424, March 2009, <https://www.rfc-editor.org/info/rfc5424>.

[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>.

[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>.

[RFC7516] Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, DOI 10.17487/RFC7516, May 2015, <https://www.rfc-editor.org/info/rfc7516>.

[RFC9162] Laurie, B., Messeri, E., and R. Stradling, "Certificate Transparency Version 2.0", RFC 9162, DOI 10.17487/RFC9162, December 2021, <https://www.rfc-editor.org/info/rfc9162>.

[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-22, October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-scitt-architecture-22>.

[I-D.ietf-scitt-scrapi] Birkholz, H. and G. Geater, "SCITT Reference API", Work in Progress, Internet-Draft, draft-ietf-scitt-scrapi-06, December 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-scitt-scrapi-06>.

[I-D.ietf-cose-merkle-tree-proofs] Steele, O., Birkholz, H., Delignat-Lavaud, A., and C. Fournet, "COSE Receipts", Work in Progress, Internet-Draft, draft-ietf-cose-merkle-tree-proofs-18, February 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-cose-merkle-tree-proofs-18>.

[I-D.ietf-cose-dilithium]
Prorock, M. and O. Steele, "ML-DSA for JOSE and COSE", Work in Progress, Internet-Draft, draft-ietf-cose-dilithium-11, November 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-cose-dilithium-11>.

[I-D.ietf-rats-corim]
Birkholz, H., Fossati, T., Deshpande, Y., Smith, N., and W. Pan, "Concise Reference Integrity Manifest", Work in Progress, Internet-Draft, draft-ietf-rats-corim, <https://datatracker.ietf.org/doc/html/draft-ietf-rats-corim>.

[I-D.fv-rats-ear]
Fossati, T. and E. Voit, "EAT Attestation Results", Work in Progress, Internet-Draft, draft-fv-rats-ear, <https://datatracker.ietf.org/doc/html/draft-fv-rats-ear>.

[I-D.ietf-rats-ar4si]
Voit, E., Lu, H., Gonzalez Sanchez, I., and A. Fongen, "Attestation Results for Secure Interactions", Work in Progress, Internet-Draft, draft-ietf-rats-ar4si, <https://datatracker.ietf.org/doc/html/draft-ietf-rats-ar4si>.

[I-D.kamimura-scitt-vcp]
Kamimura, T., "SCITT Profile for Financial Trading Audit Trails: VeritasChain Protocol (VCP)", Work in Progress, Internet-Draft, draft-kamimura-scitt-vcp-02, January 2026, <https://datatracker.ietf.org/doc/html/draft-kamimura-scitt-vcp-02>.

[I-D.ni-wimse-ai-agent-identity]
Ni, H., "WIMSE Applicability for AI Agents", Work in Progress, Internet-Draft, draft-ni-wimse-ai-agent-identity-01, <https://datatracker.ietf.org/doc/html/draft-ni-wimse-ai-agent-identity-01>.

[EU-AI-ACT]
European Parliament and Council, "Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)", Official Journal of the European Union L 1689, July 2024.

[FIPS-203] National Institute of Standards and Technology (NIST), "Module-Lattice-Based Key-Encapsulation Mechanism Standard", FIPS 203, August 2024, <https://csrc.nist.gov/pubs/fips/203/final>.

[FIPS-204] National Institute of Standards and Technology (NIST), "Module-Lattice-Based Digital Signature Standard", FIPS 204, August 2024, <https://csrc.nist.gov/pubs/fips/204/final>.

[FIPS-205] National Institute of Standards and Technology (NIST), "Stateless Hash-Based Digital Signature Standard", FIPS 205, August 2024, <https://csrc.nist.gov/pubs/fips/205/final>.

[NIST-AI-RMF] National Institute of Standards and Technology (NIST), "Artificial Intelligence Risk Management Framework (AI RMF 1.0)", NIST AI 100-1, January 2023, <https://doi.org/10.6028/NIST.AI.100-1>.

[NIST-SP-800-88] National Institute of Standards and Technology (NIST), "Guidelines for Media Sanitization", NIST SP 800-88 Rev. 1, December 2014, <https://doi.org/10.6028/NIST.SP.800-88r1>.

[ISO-42001] International Organization for Standardization, "Information technology - Artificial intelligence - Management system", ISO/IEC 42001:2023, December 2023.

[ISO-23894] International Organization for Standardization, "Information technology - Artificial intelligence - Guidance on risk management", ISO/IEC 23894:2023, February 2023.

[W3C-VC-2.0] Sporny, M., Longley, D., Chadwick, D., and O. Steele, "Verifiable Credentials Data Model v2.0", W3C Recommendation, May 2025, <https://www.w3.org/TR/vc-data-model-2.0/>.

[C2PA-2.2] Coalition for Content Provenance and Authenticity, "C2PA Technical Specification 2.2", May 2025, <https://c2pa.org/specifications/specifications/2.2/specs/C2PA_Specification.html>.

Appendix A. Lessons from Certificate Transparency

Certificate Transparency (CT) [RFC6962] provides valuable lessons for VAP design. CT demonstrated that append-only, Merkle-tree-based logs can achieve widespread deployment, but also revealed challenges that VAP should address.

A.1. Successes to Emulate

* Simplicity: CT v1's relative simplicity enabled adoption, while CT v2 [RFC9162] saw limited deployment due to increased complexity.

* Client enforcement: Browser requirements for CT created strong adoption incentives. Regulatory requirements (EU AI Act) may serve a similar function for AI provenance.

* Static serving: Recent CT innovations (e.g., Let's Encrypt's Sunlight) demonstrate that logs can be served efficiently from static infrastructure, serving millions of proofs using minimal resources. VAP implementations should consider similar efficiency optimizations.

A.2. Challenges to Address

* Split-view detection: CT's gossip protocols for detecting equivocation were never widely deployed. VAP profiles should consider how to detect a Transparency Service presenting different views to different parties.

* Ecosystem coordination: CT required coordination across browsers, CAs, and log operators. VAP will similarly require ecosystem coordination, which this framework aims to facilitate.

* Log diversity: CT has seen concentration of logs among few operators. VAP profiles should consider whether log diversity requirements are needed for their domains.

A.3. Implications for VAP

Based on CT experience, VAP profiles should:

* Prioritize operational simplicity over feature completeness.

* Design for static/cacheable proofs where possible.

* Consider enforcement mechanisms that create adoption incentives.

* Document split-view detection and mitigation strategies.

Acknowledgements

The author thanks the members of the SCITT and RATS working groups whose foundational work enables the architectural approach described in this document. The Certificate Transparency ecosystem [RFC6962] provided key inspiration for the completeness verification concepts.

This work benefits from ongoing discussions about AI governance and audit requirements in various regulatory contexts, including the European Union’s AI Act and financial market regulations.

Author’s Address

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