Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 17 November 2026

O. Vasylenko
InterAlliance OU

16 May 2026

Provenance-Attributed Inference Token (PAIT): A Protocol for Token-Level Inference Provenance and Identity-Conditioned Inference in Generative AI Systems

draft-vasylenko-pait-protocol-00

Abstract

This document specifies the Provenance-Attributed Inference Token (PAIT) protocol, a wire-level protocol for token-level provenance attribution and identity-conditioned inference in Large Language Model (LLM) systems and other generative AI systems. PAIT defines: (1) an Agent Identity Token format that binds a globally unique agent identifier to a hierarchical authorization level by means of an asymmetric digital signature; (2) a Provenance Manifest Record format that associates each output token of an inference session with verifiable attribution data referencing training-corpus segments; (3) a Trust Telemetry Signal format for aggregated session metrics emission to external monitoring endpoints; and (4) a wire protocol state machine governing the interaction between a requesting agent and a generative AI endpoint. PAIT is intended to address the transparency obligations imposed by emerging regulatory frameworks for high-risk and general-purpose AI systems, in particular those concerning the provenance of AI-generated content at sub-document granularity.

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 17 November 2026.

Copyright Notice

Table of Contents

1. Introduction

Generative artificial intelligence systems, including Large Language Models (LLMs), increasingly produce content that is consumed by downstream applications, regulated parties, and end-users without

sub-document evidence of which sources influenced specific portions of the output. Existing provenance frameworks for AI-generated content (e.g., [C2PA]) operate at the level of the complete media asset and certify origin of the asset as a whole. General-purpose provenance models such as [W3C-PROV-DM] describe upstream data lineage but do not bind individual output units of an inference session to specific training-corpus segments at runtime.

This document specifies the Provenance-Attributed Inference Token (PAIT) protocol, which addresses this gap at sub-document granularity. PAIT associates each output token of an inference session with a verifiable attribution record referencing one or more training-corpus segments; conditions inference behavior on a cryptographically verified agent identity; and emits aggregated trust metrics to an external monitoring endpoint.

1.1. Motivation

Three concurrent developments motivate this work:

* Regulatory transparency obligations. Article 50 of the Regulation (EU) 2024/1689 (EU AI Act) [EU-AI-ACT] requires providers and deployers of certain AI systems to ensure that outputs are marked as artificially generated or manipulated in a machine-readable format. Recital 133 elaborates that such marking should support detection of AI-generated content. Operationalizing this obligation at sub-document granularity for sequential generative systems requires a per-token wire format that current standards do not provide.

* Identity-conditioned access control for AI agents. As AI agents increasingly act on behalf of principals with differentiated authorization scopes, the inference behavior of generative AI endpoints must be conditioned on the verified identity and authorization level of the requesting agent, including selective access to portions of training corpora and to subsets of computational layers. General-purpose authorization frameworks such as OAuth 2.0 [RFC6749] and its delegation extensions do not address inference-time corpus and layer selection.

* Verifiable per-token provenance. Downstream consumers and auditors require machine-readable evidence linking specific output tokens to specific source-corpus segments, with cryptographic integrity guarantees. Existing attribution mechanisms internal to LLM systems produce, at best, ad-hoc attribution traces that are not standardized and are not anchored in an immutable record.

PAIT is the wire-level binding layer that connects identity-conditioned inference, per-token provenance attribution, and tamper-evident provenance recording into a single protocol.

1.2. Scope

This document specifies:

* the wire format of the PAIT Agent Identity Token (Section 4);

* the wire format of the PAIT Provenance Manifest Record (Section 5);

* the wire format of the PAIT Trust Telemetry Signal (Section 6);

* the wire protocol state machine governing the exchange between a requesting agent and a generative AI endpoint that implements PAIT (Section 7);

* signature algorithm identifiers for use in PAIT-ID (Section 4.4);

* IANA-managed registries for PAIT identifiers (Section 9).

1.3. Out of Scope

The following are explicitly out of scope for this document:

* algorithmic methods for computing the attribution vector associated with each output token;

* numerical values, formulas, or computational procedures for determining trust coefficients of training-corpus segments;

* the internal architecture of the generative AI system that implements PAIT, including its training procedure, its computational layer structure, and its language-prioritization mechanism;

* storage-layer implementation of the immutable provenance registry beyond the requirement of an append-only cryptographic hash chain.

2. Terminology

2.1. 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.

2.2. Definitions

Agent: An entity that submits an inference request to a generative AI endpoint. An Agent may be a human user, an autonomous AI system, or an automated service.

Agent Identity Protocol: The class of protocols defining a structure and format of an agent identifier comprising a globally unique machine-readable identifier, a hierarchical authorization level, a list of permitted operations, a list of prohibited operations, validity-period timestamps, a cryptographic signature over all preceding fields, and a protocol version field. The Agent Identity Token specified in Section 4 of this document is a wire-format instance of an agent identity protocol.

Agent Identity Token (PAIT-ID): The wire-format identity object defined in Section 4 of this document, instantiating the agent identity protocol class. Carries a globally unique agent identifier, an authorization level, a permitted-operations list, a validity period, and a cryptographic signature.

Authorization Level: A discrete value drawn from a finite ordered set indicating the scope of access granted to an Agent. This document defines three baseline levels (L0, L1, L2) in Section 4.3.

Delegating Agent: An Agent of authorization level L0 that issues a PAIT-ID for an Agent of lower authorization level.

Generative AI Endpoint: The network-accessible service that receives inference requests, generates output sequences, and emits PAIT-PM and PAIT-TS records.

Inference Session: A single end-to-end interaction comprising one input query, one verified PAIT-ID, the resulting output sequence, and the corresponding PAIT-PM record.

Provenance Manifest Record (PAIT-PM): The per-session machine-readable document defined in Section 5. Contains one entry per output token.

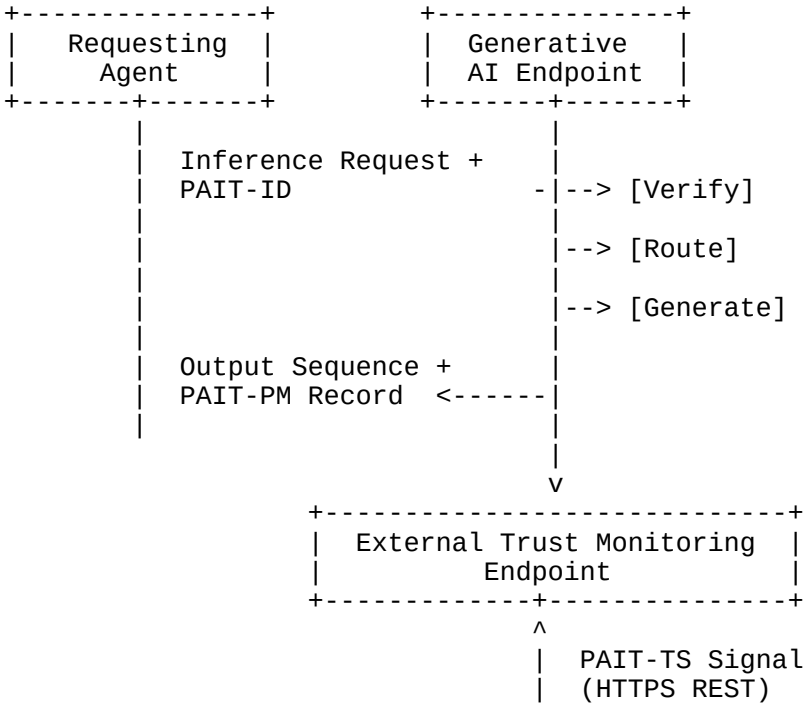
Token: A single discrete output unit produced by the generative AI endpoint. The granularity of a Token is determined by the tokenizer of the endpoint and is opaque to PAIT.

Trust Telemetry Signal (PAIT-TS): The aggregated per-session metrics object defined in Section 6. Transmitted to an external monitoring endpoint.

Validity Period: The closed time interval, bounded by validity_start_utc and validity_end_utc, during which a PAIT-ID is considered valid for the purpose of cryptographic verification.

3. Protocol Overview

PAIT is a request-response protocol with four atomic wire-format objects:



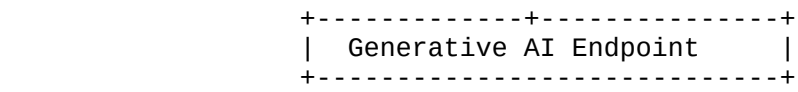


Figure 1: PAIT Protocol Overview

The four atomic objects are:

* PAIT-ID (Section 4): identifies and authorizes the requesting agent.

* PAIT-PM (Section 5): records per-token provenance for the output sequence.

* PAIT-TS (Section 6): emits aggregated session metrics to an external monitoring endpoint.

* PAIT Registry Record: not directly transmitted on the wire as a protocol element, but referenced by PAIT-PM and PAIT-TS as the immutable cryptographic record of inference sessions and corpus segments. Registry record format is out of scope for this document.

4. Agent Identity Token (PAIT-ID)

4.1. Token Structure

A PAIT-ID is a JSON object with the following top-level fields:

```
{
  "protocol_version": "1.0",
  "gai":              "<UUID>",
  "auth_level":       "<L0|L1|L2>",
  "allowed_ops":      [ "<op>", ... ],
  "prohibited_ops":   [ "<op>", ... ],
  "validity_start_utc":"<ISO 8601 timestamp>",
  "validity_end_utc":  "<ISO 8601 timestamp>",
  "signature": {
    "alg":  "<ES256|EdDSA>",
    "kid":  "<key identifier>",
    "value":"<base64url-encoded signature>"
  }
}
```

Figure 2: PAIT-ID JSON structure

The signature value is computed over the canonicalized JSON serialization of all preceding fields (protocol_version through validity_end_utc) using the algorithm identified in signature.alg. The canonicalization procedure is specified by [RFC8785].

4.2. Field Definitions

protocol_version:
: A string identifying the PAIT protocol version. This document specifies version "1.0". Implementations MUST reject PAIT-IDs with an unrecognized protocol_version.

gai (Global Agent Identifier):
A globally unique machine-readable identifier in UUID version 4 format [RFC4122]. Each Agent SHOULD be assigned exactly one gai value across all PAIT deployments.

auth_level:
A string indicating the hierarchical authorization level assigned to the Agent. This document defines three baseline values ("L0", "L1", "L2") in Section 4.3. Extensions to this document MAY define additional values.

allowed_ops:
A JSON array of machine-readable operation identifiers that the Agent is permitted to invoke against the Generative AI Endpoint. An empty array signifies that all operations permissible at the assigned auth_level are allowed. Operation identifiers are deployment-specific strings.

prohibited_ops:
A JSON array of machine-readable operation identifiers that the Agent is explicitly forbidden from invoking, irrespective of auth_level. An empty array signifies no explicit prohibitions. Where allowed_ops and prohibited_ops both contain the same identifier, prohibited_ops takes precedence.

validity_start_utc, validity_end_utc:
ISO 8601 [ISO8601] timestamps in UTC defining the closed time interval during which the PAIT-ID is valid. Implementations MUST reject PAIT-IDs whose current verification time falls outside this interval and MUST assign the minimum authorization level (defined as the lowest-privilege level in the implementation’s level set, typically L2) to the requesting Agent.

signature.alg:
An identifier for the digital signature algorithm used. This document defines two values: "ES256" (ECDSA on curve P-256, per [RFC7518] Section 3.4) and "EdDSA" (Edwards-curve DSA, per [RFC8037]).

signature.kid:
An opaque key identifier referencing the public key against which signature.value is verified. The mapping from kid to public key material is out of scope for this document.

signature.value:
The base64url-encoded signature, computed as described in Section 4.1.

4.3. Authorization Levels

This document defines three baseline authorization levels:

* L0 (full): Highest privilege. Grants the requesting Agent

access to the full training corpus available to the Generative AI Endpoint and to all computational layers of the underlying model. L0 Agents MAY act as Delegating Agents (Section 4.5).

* L1 (verified): Intermediate privilege. Grants the requesting Agent access to a subset of the training corpus restricted to segments bearing open licenses (e.g., CC0, CC BY 4.0, Apache 2.0) and to a subset of the computational layers of the underlying model.

* L2 (public): Minimum privilege. Grants the requesting Agent access to the public subset of the training corpus (e.g., segments bearing CC0 or Public Domain) and to a restricted subset of computational layers.

The mapping from auth_level to specific corpus subset and to specific computational-layer subset is implementation-specific. PAIT requires only that the mapping be deterministic and that L0 ⊇ L1 ⊇ L2 with respect to accessible corpus and layers.

4.4. Signature Algorithms

Implementations MUST support at least one of "ES256" and "EdDSA". Implementations SHOULD support both. Additional algorithms MAY be registered via the IANA registry defined in Section 9.

4.5. Delegation

An Agent of authorization level L0 MAY issue PAIT-IDs for Agents of lower authorization level (L1 or L2). A delegated PAIT-ID:

* MUST be signed by the Delegating Agent;

* MUST contain an allowed_ops value that is a subset of the Delegating Agent’s own allowed_ops;

* MUST have a validity_end_utc not later than the validity_end_utc of the Delegating Agent’s PAIT-ID;

* inherits access only to the corpus subset and layer subset of its own (lower) auth_level, regardless of the corpus and layer subset accessible to the Delegating Agent.

5. Provenance Manifest Record Format (PAIT-PM)

5.1. Record Structure

A PAIT-PM record is emitted by the Generative AI Endpoint for each

Inference Session and is transmitted to the requesting Agent together with the output sequence. A PAIT-PM record is encoded as a JSON Lines [JSONL] document in which each line corresponds to exactly one output token of the Inference Session.

The first line of a PAIT-PM record MUST be a header object with the following fields:

```
{
  "type":            "pait-pm-header",
  "protocol_version":"1.0",
  "session_id":      "<UUID>",
  "agent_id":        "<UUID>",  // gai from PAIT-ID
  "start_utc":       "<ISO 8601 timestamp>",
  "model_id":        "<endpoint-specific identifier>",
  "prev_session_hash":"<base64url-encoded SHA-256 digest>"
}
```

Figure 3: PAIT-PM header object structure

Each subsequent line is a per-token record (Section 5.2).

5.2. Per-Token Fields

```
{
  "type":            "pait-pm-token",
  "token_idx":       <integer>,
  "token_repr":      "<opaque endpoint-specific representation>",
  "attribution":     [
    {
      "segment_id":  "<segment identifier>",
      "weight":      <number in [0,1]>,
      "license":     "<license class string>"
    },
    ...
  ],
  "license_purity":  <number in [0,1]>
}
```

Figure 4: PAIT-PM per-token record structure

Field semantics:

token_idx:
: Zero-based index of the token within the output sequence.

token_repr:
: An opaque representation of the output token in the endpoint-specific tokenizer. PAIT does not constrain this representation.

attribution:
: A JSON array of attribution entries. Each entry references a training-corpus segment that contributed to the generation of this token. The weight field denotes the relative contribution of the referenced segment. The sum of weight values across all entries SHOULD be 1.0 (within rounding tolerance), unless the attribution vector has been renormalized after filtering. The license field denotes the license class associated with the referenced segment.

license_purity:
: A scalar in the closed interval [0,1] denoting an aggregate indicator of license-compatible attribution for this token. The method by which license_purity is computed is out of scope

for this document.

The final line of a PAIT-PM record MUST be a footer object:

```
{
  "type":             "pait-pm-footer",
  "session_id":       "<UUID>",
  "token_count":      <integer>,
  "end_utc":          "<ISO 8601 timestamp>",
  "manifest_hash":    "<base64url-encoded SHA-256 digest>"
}
```

Figure 5: PAIT-PM footer object structure

The manifest_hash is computed by an implementation-defined deterministic function over the canonical representation of the header object and all per-token records, in order. The hash function MUST be SHA-256 [RFC6234] or stronger. Implementations MUST ensure identical output for identical input.

6. Trust Telemetry Signal Format (PAIT-TS)

A PAIT-TS signal is emitted by the Generative AI Endpoint to an external monitoring endpoint upon completion of an Inference Session, or upon detection of an anomalous condition during a session.

A PAIT-TS signal is a JSON object transmitted over HTTPS to a pre-configured REST endpoint:

```
{
  "type":               "pait-ts-signal",
  "protocol_version":   "1.0",
  "session_id":         "<UUID>",
```



```
  "agent_id":           "<UUID>",
  "auth_level":         "<L0|L1|L2>",
  "token_count":        <integer>,
  "license_distribution":{
    "<license class>":  <number in [0,1]>,
    ...
  },
  "high_trust_fraction":<number in [0,1]>,
  "timestamp_utc":      "<ISO 8601 timestamp>",
  "signal_mode":        "<batch|out_of_order>",
  "anomaly_flag":       <boolean>,
  "endpoint_signature": {
    "alg":   "<ES256|EdDSA>",
    "kid":   "<key identifier>",
    "value": "<base64url-encoded signature>"
  }
}
```

Figure 6: PAIT-TS signal structure

Implementations MUST transmit PAIT-TS signals over a confidential and integrity-protected channel. TLS 1.3 [RFC8446] is REQUIRED.

Implementations MUST support two transmission modes:

* batch: aggregated signals transmitted at a configured interval, not less frequent than once per 24 hours;

* out_of_order: immediate transmission upon detection of an anomalous condition. The conditions classified as anomalous and the corresponding thresholds are out of scope for this document. When signal_mode is "out_of_order", anomaly_flag MUST be set to true.

7. Wire Protocol State Machine

The PAIT protocol state machine governs the interaction between the requesting Agent and the Generative AI Endpoint. Nine states are defined:

S0 (Idle):
The Endpoint awaits an inference request.

S1 (Request Received):
The Endpoint has received an inference request containing a PAIT-ID. Transitions to S2.

S2 (Identity Verifying):

The Endpoint verifies the PAIT-ID:

* validates protocol_version;

* validates that the current verification time is within [validity_start_utc, validity_end_utc];

* retrieves the public key identified by signature.kid;

* verifies signature.value against the canonicalized serialization of preceding fields using signature.alg.

On verification failure (any of: absent PAIT-ID, invalid signature, expired validity period, unrecognized protocol_version), the Endpoint MUST assign the minimum authorization level (typically L2) to the requesting Agent and transition to S3. On verification success, the Endpoint transitions to S3 with the verified auth_level.

S3 (Inference Routing):
The Endpoint selects:

* the subset of the training corpus accessible at the verified auth_level;

* the subset of computational layers accessible at the verified auth_level;

* the processing mode for the inference request.

The Endpoint transitions to S4.

S4 (Generation):
The Endpoint generates the output sequence. For each output

token, the Endpoint computes attribution data as defined by the Endpoint's internal attribution method (out of scope for this document) and appends a per-token PAIT-PM record to the session manifest. The Endpoint transitions to S5 after the final token of the sequence has been generated.

S5 (Manifest Sealing):
The Endpoint emits the PAIT-PM footer (Section 5.2), computes the manifest_hash over the entire manifest, and seals the manifest. The Endpoint transitions to S6.

S6 (Registry Append):
The Endpoint writes a new record of session-completion type to its immutable cryptographic provenance registry. The new

record incorporates the SHA-256 hash of the immediately preceding registry record into its prev_hash field. The Endpoint transitions to S7.

S7 (Response Transmission):
The Endpoint transmits the output sequence and the sealed PAIT-PM record to the requesting Agent. The Endpoint transitions to S8.

S8 (Telemetry Emission):
The Endpoint emits a PAIT-TS signal (Section 6) to the configured external monitoring endpoint. If an anomalous condition was detected during S4, the Endpoint MUST emit the PAIT-TS signal in "out_of_order" mode with anomaly_flag set to true; otherwise, the Endpoint MAY defer emission to a subsequent batch. After emission (or deferral), the Endpoint transitions back to S0.

The state machine guarantees that no PAIT-PM record may be transmitted to a requesting Agent without prior identity verification (S2) and prior registry append (S6).

8. Security Considerations

This section addresses the principal security considerations applicable to PAIT.

Identity verification. The integrity of the PAIT protocol depends critically on the cryptographic verification of PAIT-IDs in state S2. Implementations MUST:

* reject PAIT-IDs with invalid signatures;

* reject PAIT-IDs whose verification time falls outside the [validity_start_utc, validity_end_utc] interval;

* assign the minimum authorization level in any case of verification failure.

Replay protection. PAIT-IDs are time-bounded by the validity period. Implementations SHOULD additionally maintain a record of recently observed PAIT-IDs (identified by signature.value) and reject duplicates received within a configurable window, to defend against replay attacks within an otherwise valid

validity period.

Confidentiality. PAIT-ID, PAIT-PM, and PAIT-TS objects may contain information that the requesting Agent or the operator

considers sensitive. Implementations MUST transport PAIT objects over a confidentiality- and integrity-protected channel. TLS 1.3 [RFC8446] is REQUIRED.

Key management. The mapping from signature.kid to public key material is out of scope for this document. Implementations SHOULD use established key-distribution mechanisms such as JWKS [RFC7517] or DNS-based key discovery.

Registry integrity. The cryptographic hash chain of the immutable provenance registry provides tamper-evident integrity: modification of any registry record causes a mismatch of the record_hash and of prev_hash in all subsequent records. Implementations MUST store the registry in append-only mode. Implementations SHOULD provide a read-only audit interface for external verification.

Trust telemetry endpoint authentication. The Endpoint emitting PAIT-TS signals MUST authenticate the external monitoring endpoint to which signals are transmitted, e.g., via mutual TLS, to prevent diversion of telemetry to an attacker-controlled endpoint. PAIT-TS signals MUST be signed by the Endpoint using endpoint_signature (Section 6).

Privacy. Implementations MUST consider that token_idx, attribution, and timestamp fields collectively may permit inference of user query content. Implementations SHOULD evaluate compatibility with applicable privacy frameworks (e.g., [GDPR]) prior to deployment.

Compatibility with self-prior-art exclusions. This protocol specifies wire formats and state-machine behavior only. Algorithmic methods for attribution-vector computation, trust-coefficient determination, and routing-decision logic are intentionally out of scope, as described in Section 1.3.

9. IANA Considerations

This document requests IANA to create three new registries under the "PAIT Parameters" registry group:

PAIT-ID Signature Algorithms registry:
Initial values:
* "ES256" (ECDSA on curve P-256, per [RFC7518] Section 3.4)
* "EdDSA" (Edwards-curve DSA, per [RFC8037])
Registration policy: Specification Required [RFC8126].

PAIT-ID Authorization Levels registry:

Initial values:
* "L0" (full)
* "L1" (verified)
* "L2" (public)
Registration policy: Specification Required [RFC8126].

PAIT Protocol Versions registry:
Initial value:
* "1.0" (this document)
Registration policy: Standards Action [RFC8126].

10. Intellectual Property Rights

The technology described in this document is the subject of an international patent application, PCT/IB2026/053131 (priority date 25 March 2026, 19 claims, International Searching Authority: European Patent Office). The patent holder is willing to grant a license for any standards-essential patent claims under fair, reasonable, and non-discriminatory (FRAND) terms with reciprocity. A formal IPR disclosure has been filed with the IETF. See the IETF IPR disclosure record for the current declaration.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, July 2005, <https://www.rfc-editor.org/info/rfc4122>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

[RFC7517] Jones, M., "JSON Web Key (JWK)", RFC 7517, May 2015, <https://www.rfc-editor.org/info/rfc7517>.

[RFC7518] Jones, M., "JSON Web Algorithms (JWA)", RFC 7518,

May 2015, <https://www.rfc-editor.org/info/rfc7518>.

[RFC8037] Liusvaara, I., "CFRG Elliptic Curve Diffie-Hellman (ECDH) and Signatures in JSON Object Signing and Encryption (JOSE)", RFC 8037, January 2017, <https://www.rfc-editor.org/info/rfc8037>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8785] Rundgren, A., Jordan, B., and S. Erdtman, "JSON Canonicalization Scheme (JCS)", RFC 8785, June 2020, <https://www.rfc-editor.org/info/rfc8785>.

[ISO8601] International Organization for Standardization, "Date and time -- Representations for information interchange -- Part 1: Basic rules", ISO 8601-1:2019, February 2019.

11.2. Informative References

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

[W3C-PROV-DM] Moreau, L. and P. Missier, Eds., "PROV-DM: The PROV Data Model", W3C Recommendation, 30 April 2013, <https://www.w3.org/TR/prov-dm/>.

[EU-AI-ACT] European Parliament and Council, "Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)", Official Journal of the European Union,

July 2024, <https://eur-lex.europa.eu/eli/reg/2024/1689/oj>.

[GDPR] European Parliament and Council, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data (General Data Protection Regulation)", Official Journal of the European Union, April 2016, <https://eur-lex.europa.eu/eli/reg/2016/679/oj>.

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, October 2012, <https://www.rfc-editor.org/info/rfc6749>.

[JSONL] "JSON Lines", <https://jsonlines.org/>.

[ITU-T-X-LLMP-TIP] Vasylenko, O., "Proposed new work item X.llmp-tip: LLM token-level inference provenance", ITU-T SG17 Contribution T25-SG17-C-0461, 15 May 2026, <https://www.itu.int/md/T25-SG17-C-0461/>.

Appendix A. Relationship to ITU-T Work Items

This protocol specification is a companion document to ITU-T SG17 Contribution C-0461 (X.llmp-tip), which establishes the requirements framework for token-level LLM provenance within the ITU-T standardization process.

The requirements in X.llmp-tip operate at the conceptual level. This IETF Internet-Draft provides a protocol-level realization, defining wire formats, message structures, and state transitions.

The two documents are intended for parallel development: X.llmp-tip within ITU-T SG17 (security and trust requirements) and this draft within the IETF (protocol specification).

Related ITU-T contributions from the same contributor:

- C-0456 (X.gaiv): Globally interoperable agent identity verification.
- C-0457 (X.aaid): Agent authentication and identity delegation.
- C-0458 (X.rcae): Regulatory continuous attestation engine for AI agents.
- C-0459 (X.atcp): Agent trust confidence primitives.
- C-0460 (X.aam): AI accountability manifest.

These six contributions together define the Agentic Trust Control Plane architecture within ITU-T SG17.

Author’s Address

Oleh Vasylenko
InterAlliance OU
Tallinn
Estonia

Email: oleg@vasylenko.tel