Network Working Group
Internet-Draft
Intended status: Informational
Expires: 29 November 2026

B. Morrison
Alter Meridian Pty Ltd

28 May 2026

Substrate-Provenance Annotation Grammar for Large-Language-Model Output
draft-morrison-substrate-provenance-grammar-00

Abstract

This memo describes a wire-level annotation grammar by which a large-language-model output may carry, at emission and at the granularity of an individual assertion, a provenance label drawn from a closed enumerated vocabulary of substrate-class identifiers. The memo defines the closed vocabulary, the per-assertion attachment form, the admissibility discipline a relying party MAY apply to the labels, and two terminal output states (UNVERIFIED-INFERENCE and DECAYED-TO-UNCERTAINTY) equal-rank with assertion and denial. The memo does not say what an inference system MUST do. It defines the wire grammar by which a relying party may inspect what the inference system DID with respect to the substrates it consulted. The memo is Informational.

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 29 November 2026.

Copyright Notice

Table of Contents

1. 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."

2. Introduction

When a large-language-model output is consumed by another agent, by a downstream automation, or by a relying party with action authority, the consuming entity has no wire-level mechanism to distinguish three cases: (i) the model emitted the assertion from training-corpus-resident pattern without consulting any external substrate; (ii) the model emitted the assertion after consulting an external substrate whose state corroborated the assertion within an admissibility window; (iii) the model emitted the assertion after consulting an external substrate whose state did not corroborate the assertion, and the model proceeded anyway.

Existing approaches to this problem operate at the prose layer: post-hoc citation insertion by a separate retrieval orchestrator, natural-language hedge phrasing ("I believe", "it appears", "according to"), per-paragraph confidence scores rendered as adjectives, or refusal to answer. All four are parsed from the surface form rather than carried as a distinct output element. All four can be defeated by a model trained to substitute hedge phrasing for substrate consultation.

This memo defines a wire-level grammar by which the inference system declares, at the granularity of an individual assertion within its output, which substrate-class (if any) corroborated the assertion at emission. The grammar is closed-vocabulary, finite, and version-anchored. The consuming entity parses the annotation without interpreting prose. Two terminal annotations, UNVERIFIED-INFERENCE and DECAYED-TO-UNCERTAINTY, are equal-rank with assertion and denial. They are a distinct output state, not a confidence score and not a hedge phrase.

3. Conventions and Definitions

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.

The following terms are defined for the purposes of this document:

Assertion. A token sequence in the inference system’s output that makes a truth-evaluable claim about state external to the inference system itself.

Substrate. A source of state external to the inference system, non-

writeable by the inference system at consultation time, and observable to a relying party with independent access.

Substrate-class. A named, version-anchored category of substrate. A substrate-class identifier is an opaque string drawn from the closed enumerated vocabulary of Section 4.

Provenance annotation. A wire-level label attached to an individual assertion that names the substrate-class corroborating the assertion, or names one of the two terminal annotations defined in Section 5.

Admissibility window. A predicate, evaluated at assertion emission time, that determines whether the corroborating observation from a named substrate-class is recent enough to count toward admission.

Cardinality-thresholded admissibility. A discipline by which an assertion is admitted to the inference system's output only if at least _k_ corroborating observations from substrate-class-distinct sources are available within the admissibility window, where _k_ is a relying-party-configurable floor.

4. The Closed Substrate-Class Vocabulary

The vocabulary defined by this memo is closed, finite, and version-anchored. An implementation parsing a provenance annotation MUST recognise the annotation if and only if the substrate-class identifier appears in the version of the vocabulary the implementation has loaded. Unknown substrate-class identifiers MUST NOT be silently interpreted; an implementation that encounters one MUST treat the annotation as if it were UNVERIFIED-INFERENCE (Section 5).

The version-anchor scheme used by this memo is a dotted major.minor pair appearing as a leading element of the substrate-class identifier. The vocabulary defined in this revision of the memo carries the version anchor 1.0. Future revisions of this memo MAY add substrate-classes; addition is a minor-version bump. Future revisions MUST NOT remove substrate-classes without a major-version bump.

The version-1.0 closed substrate-class vocabulary:

substrate.git.log The output of git log on a repository observable to the relying party, with the assertion's content appearing within a commit reachable from a named reference. Compute-location: the relying party's local working tree or a trusted mirror.

substrate.grep The output of grep over a file-set observable to the relying party, with the assertion's content appearing in a named region of a named file. Compute-location: the relying party's local file system.

substrate.code.read The contents of a file at a specified path, optionally with line offset and limit, as read from a file system observable to the relying party. Compute-location: the relying party's local file system.

substrate.fs.mtime The modification timestamp of a file observable to the relying party. Compute-location: the relying party's local file system.

substrate.mcp.brief The output of a query against a Model-Context-Protocol- discoverable service whose state is owned by a party other than the inference system vendor. Compute-location: the MCP service's own infrastructure.

substrate.do.sse-count The numeric subscriber count on a server-sent-event channel, observed by a relying party with independent subscription to the same channel. Compute-location: the SSE server's infrastructure.

substrate.unix.peercred The kernel-reported peer credentials on a Unix-domain socket connection. Compute-location: the kernel of the host on which the observation is made.

unverified-inference A sentinel substrate-class identifier reserved for assertions emitted without corroboration from any other substrate-class in this vocabulary. This is the terminal annotation defined in Section 5; it is listed here for parser completeness.

The eight identifiers above constitute the entirety of the version-1.0 closed vocabulary.

5. Terminal Annotations

Two annotation values are terminal: they signal a distinct output state of the inference system, equal-rank with assertion and denial, rather than corroboration by any substrate.

unverified-inference The inference system emitted the assertion without corroboration from any substrate-class in the vocabulary. This is not a confidence score; it is the inference system's declaration that the assertion is uncorroborated.

decayed-to-uncertainty The inference system attempted to corroborate the assertion from a substrate-class in the vocabulary, but the observation aged beyond the admissibility window before emission. This is not a confidence score; it is the inference system's declaration that corroboration was attempted and aged out.

Both terminal annotations are first-class output tokens. A relying party parsing the wire output observes them in the same structural slot in which substrate-class identifiers appear; the parser applies the terminal-annotation disposition without inspecting prose. The two terminal annotations are distinct from refusal to answer, from explicit denial, and from the absence of annotation.

6. Attachment Form

A provenance annotation is attached to an individual assertion in the inference system's output. This memo describes the abstract attachment relationship; the concrete wire encoding is the implementation's choice and is not normative herein.

The annotation form is the tuple:

(assertion-span, substrate-class-identifier, observation-id?, ts?)

where:

* *assertion-span* identifies the token sequence to which the annotation applies (the implementation's choice of identifier: character offsets, token offsets, structured-output field path, or equivalent);

* *substrate-class-identifier* is drawn from the closed vocabulary of Section 4 or is one of the terminal annotations of Section 5;

* *observation-id* (OPTIONAL) is a stable identifier for the specific observation that corroborated the assertion, suitable for re-querying the same substrate to confirm the binding;

* *ts* (OPTIONAL) is the time at which the corroborating observation was made.

Two concrete encodings are illustrative and not normative:

JSON-structured-output encoding per [RFC8259]:

```
{
  "assertion": "CHANGELOG.md contains an entry dated 2026-05-25",
  "provenance": {
    "substrate_class": "substrate.code.read",
    "observation_id": "sha256:e3b0c4...",
    "ts": "2026-05-28T11:40:00Z"
  }
}
```

In-line bracketed annotation, for free-text outputs:

```
The file CHANGELOG.md contains an entry dated 2026-05-25.
[substrate.code.read; observation-id=sha256:e3b0c4...;
 ts=2026-05-28T11:40:00Z]
```

7. Admissibility Discipline

A relying party MAY apply a cardinality-thresholded admissibility discipline to inference system output annotated under this grammar. The discipline is parameterised by:

* *k*, the floor cardinality of distinct substrate-classes required to corroborate an assertion before the relying party admits the assertion for downstream action.

* *W*, the admissibility window: a per-substrate-class duration beyond which an observation is considered aged.

Reference values:

* k = 2 is the floor for assertions whose downstream effect is bounded to the relying party’s own state.

* k = 3 is the floor for assertions whose downstream effect affects state external to the relying party (settlement, policy decision, automated action with external visibility).

The relying party applies the discipline by counting, for each assertion in the inference system’s output, the distinct substrate-class identifiers appearing in the assertion’s provenance annotations whose ts field is within W. Assertions not meeting the cardinality floor are not admitted; assertions annotated with unverified-inference or decayed-to-uncertainty are not admitted by virtue of the terminal annotation itself.

This memo does not specify what the relying party MUST do with an inadmissible assertion. Common dispositions include: discarding the assertion silently, surfacing it to a human reviewer, requesting re-

emission from the inference system, or substituting an explicit refusal in the relying party's own output to the next consuming entity. Each disposition is the relying party's own policy choice and is not constrained by this memo.

8. Why the Vocabulary Is Closed

A reader may ask why the substrate-class vocabulary is closed rather than open-extensible.

An open-extensible vocabulary would permit any inference system to introduce a new substrate-class identifier and emit corroboration annotations under it. A relying party encountering an unrecognised identifier would face a choice: trust the new identifier on its face, refuse the assertion, or treat the identifier as equivalent to a fallback known identifier. Each choice is inferior to the closed-vocabulary posture of this memo:

* Trusting the unrecognised identifier on its face permits an inference system to bypass the admissibility discipline by inventing self-corroborating substrate-classes.

* Refusing the assertion stalls the consuming pipeline on a vocabulary-version mismatch, an interop-combinatorics failure mode shared with the envelope-coordination anti-pattern of related work.

* Fallback-equivalence collapses the semantic distinctions the vocabulary is intended to preserve.

The closed-vocabulary posture treats unrecognised identifiers as unverified-inference (Section 5). This preserves the admissibility discipline under vocabulary drift while leaving the relying party free to upgrade its parser to a newer vocabulary version.

9. Relation to Prior Art

The contribution of this memo is the joint articulation of a closed substrate-class vocabulary, per-assertion attachment, and two terminal annotations as a single wire-level grammar. Each adjacent prior-art family is distinct from this grammar in at least one of the three components.

Retrieval-Augmented Generation performs retrieval against an external corpus and conditions generation on the retrieved context. RAG is an inference-system architecture; this memo describes an output-side annotation grammar. A RAG-architected inference system MAY emit annotations under this grammar; a non-RAG inference system MAY emit annotations under this grammar. The grammar is orthogonal to the architecture.

Constitutional AI and self-critique architectures apply a second model pass to evaluate the first pass's output against a specification. The output of such a system is not annotated at the granularity of an individual assertion against an external substrate-class; it is annotated, if at all, with a critique-pass verdict against a specification authored by the model vendor. Annotation against a vendor-authored specification differs in kind from annotation against a substrate observable to a relying party.

Multi-agent debate and related multi-pass deliberation architectures produce a single consensus output from multiple agent passes. The output's provenance is the agents' agreement process, not a substrate observable to a relying party.

Hidden-state probing inspects the inference system's internal activations to estimate the system's own confidence in its output. Confidence is a property of the inference system; substrate-class corroboration is a property of the relying party's own observation surface. The two are categorically different information sources.

Cryptographically-anchored append-only logs (Certificate Transparency [RFC6962], trusted timestamping per [RFC3161]) are candidate corroborating substrates under this grammar; each is a substrate-class a future revision of the vocabulary MAY add. A chained log considered in isolation is not a per-assertion annotation grammar.

10. IANA Considerations

This memo requires no IANA actions in its present revision. A future revision may request establishment of an IANA registry for substrate-class identifiers, governed by the closed-vocabulary discipline of Section 4 and Section 8.

11. Security Considerations

The grammar specified by this memo surfaces three classes of attack absent from prose-only or hedge-phrasing approaches. The mitigations described below are operational rather than wire-level; this memo defines the grammar only, and an implementation's operational posture is its own.

11.1. Annotation Fabrication

An inference system may emit a provenance annotation citing a substrate-class corroboration that did not occur. The grammar specified by this memo provides no cryptographic binding between the annotation and any observed substrate state. A relying party MUST NOT treat the annotation as evidence of corroboration; the annotation is a declaration of the inference system's claim about its own behaviour, which the relying party MAY independently verify by re-querying the named substrate with the optional observation-id. Cryptographic binding of annotations to observed substrate state is out of scope for this memo and is the subject of separate work.

11.2. Vocabulary Drift

An inference system implemented against a newer version of the vocabulary may emit substrate-class identifiers not present in a relying party's older vocabulary. Per Section 4, the relying party treats unrecognised identifiers as unverified-inference. This is a fail-closed posture and is correct. A relying party operating at a substantially older vocabulary version SHOULD upgrade its parser to the current published version of this memo.

11.3. Substrate Capture

An adversary controlling a substrate identified in the vocabulary may engineer the substrate's state to corroborate assertions of the adversary's choice. The admissibility discipline of Section 7, with k = 2 or k = 3, mitigates this by requiring corroboration from substrate-class-distinct sources before admission. An adversary controlling all k substrate-classes can defeat the discipline; selection of independent substrate-classes is the relying party's operational responsibility and is not specified by this memo.

12. Privacy Considerations

Provenance annotations expose to consumers of the inference system's output the categories of substrate the inference system consulted. In typical deployments, the substrate-class identifier is a category not a record; the per-record observation-id, if emitted, may carry the privacy properties of the underlying substrate (a filesystem path, a content hash, a peer credential identifier).

A relying party emitting annotations under this grammar to a further downstream consumer SHOULD apply standard identity-binding hygiene to substrate observables: pseudonymous tier observations need not carry strong identifiers; identity-bound tier observations carry the identity-binding strength of the underlying substrate.

The grammar specified by this memo does not require, and does not recommend, attachment of identifiers tying the inference system's output to a particular human end-user. Such attachments, if made, are outside the scope of this memo.

13. 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>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

14. Informative References

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

[RFC3161] Adams, C., Cain, P., Pinkas, D., and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", RFC 3161, DOI 10.17487/RFC3161, August 2001, <https://www.rfc-editor.org/info/rfc3161>.

Acknowledgements

This memo articulates a grammar layered above the substrate-observation primitive of related work in the morrison-* family on IETF datatracker. Its development is the joint product of deployed agentic-system experience and structural analysis of adjacent prior art.

IPR Posture

The applicant of any patent rights that may be construed to read on the grammar specified by this memo will file an IPR disclosure under the IETF's standard procedures. The applicant's intended licensing posture for any such rights is royalty-free with defensive-termination, consistent with the applicant's published IPR disclosures on companion memos in the morrison-* family.

Author’s Address

Blake Morrison
Alter Meridian Pty Ltd
Email: blake@truealter.com