Reliable Context

RELIABLE CONTEXT / RESEARCH & PRACTICAL GUIDANCE

W3C PROV for LLM Context Provenance: A Testable Profile

> Reliable Context is a kernel of truth: a system of record for organizations that fuse private and public information and must defend every value they act on. One canonical record per real thing, a cited source and an observation date behind every value, unknowns declared rather than left blank, and any past state of the record reproducible.

Published by Reliable Context · 2026-09-22 · w3c prov for llm context provenance · llm context provenance · prov-o · answer traceability · ai context lineage · llm data provenance · reproducible llm answers · provenance conformance

Original article: https://reliablecontext.com/articles/w3c-prov-llm-context-provenance



In this article

1. Executive Summary
2. Introduction and Background
3. Key Changes
4. The Answer-Evidence Application Profile
5. Implementation Considerations and Process Changes
6. Conformance and Reconstruction Testing
7. Data Analysis and Evidence
8. Case Studies and Real-World Examples
9. Implications and Future Directions
10. Frequently Asked Questions (FAQs)
11. Conclusion

Executive Summary

W3C PROV for LLM context provenance is useful only after an application profile removes freedoms that are harmless in a general provenance model but fatal to independent reconstruction. PROV defines an entity as a thing with fixed aspects and distinguishes activities and responsible agents around it (www.w3.org). Those concepts can describe an answer, its claim, a retrieved source snapshot, the extraction or generation step, and the accountable system or person. They do not, by themselves, require the source version, quoted span, content digest, observation time, or reason a field is unknown.

The proposed **answer-evidence profile** therefore makes those reconstruction facts mandatory. A conforming export must let a second implementation recover six things without access to the producing system: the answer and atomic claim, the exact source state, the supporting span, the transformations and retrievals, the responsible agents, and the times at which the source was observed and each artifact was generated. Identifiers are already required for PROV entities, activities, and agents, while activity times are optional (www.w3.org). This profile deliberately strengthens the time requirement and uses a named **PROV bundle** to identify who asserted the lineage. Exact resource versions and uniform timestamp precision are established interchange practices (support.datacite.org) (www.rfc-editor.org).

Conformance requires more than parsing. The fixture must survive two independent processing paths, preserve typed relations, timestamp precision, source fixity, and unknown reasons, and respond correctly to source-revision, alternate-representation, and changed-attribution mutations. **Reconstruction completeness** is reported as recovered required facts divided by required fixture facts, with every omission named. The score is a profile metric, not a W3C metric. This distinction matters because the 2013 PROV implementation report used **280** constraint tests (www.w3.org). Independent conformance guidance cautions that execution ordering can be necessary for perfect reproducibility (openreview.net).

Evidence from current language-model research reinforces the separation between lineage and answer quality. MatPROV reports **2,367 procedures from 1,568 papers**, yet explicitly notes that automated extraction can omit information (arxiv.org). ALCE found incomplete citation support **50% of the time**, while TREC

separately evaluates support, fluency, retrieval, and nugget coverage (arxiv.org) (trec-rag.github.io). Attributable to Identified Sources says source quality needs a complementary method (direct.mit.edu). A valid PROV graph can support traceability, but it cannot establish truth, authority, permission, or legal sufficiency.

Introduction and Background

An answer citation is a pointer. **Lineage** is a path through source states and transformations. **Provenance** is a structured account of the entities, activities, and agents involved in producing an artifact. **Explanation** concerns why a model produced an output, which may require evidence about model behavior that a lineage graph does not contain. Treating these terms as synonyms produces exports that look documented but cannot be replayed or challenged.

The practical question for an enterprise information architect is narrow: can another system receive an export and reconstruct the claim, exact source state, transformation, responsible identity, and observation time? This is the **reconstruction test**. It evaluates the contract visible at the boundary, not prompts, queues, retrieval architecture, or model internals. It also avoids equating a familiar URL with a preserved source. DataCite recommends a new Digital Object Identifier for a major version and a relation to the previous version (support.datacite.org); Zenodo similarly recommends citing a version-specific DOI when exact recovery matters (zenodo.org). Crossref uses controlled relationship terms to connect versions and derivations rather than relying on free text (www.crossref.org).

W3C's core PROV recommendations provide generation, usage, derivation, attribution, association, revision, and alternate-representation concepts. PROV is therefore a strong interchange foundation. It is not an LLM answer schema, and it does not select a JSON serialization or prescribe evidence-span fields. FAIR guidance independently calls for data and metadata to carry detailed provenance (www.nature.com).

For a system of record, the boundary guarantee is the relevant comparison. Reliable Context states that exported values carry source, excerpt, locator, and dates, and that past record states can be reproduced. Those are observable export properties that this profile can test. They do not reveal or imply how the product is built.

In practical terms, a **W3C PROV profile for AI** makes **AI context lineage** explicit but does not automatically make all answers true. **Reproducible LLM answers** additionally require recoverable inputs, stable identities, typed transformations, and declared unknowns. Using **PROV-O for generative AI** supplies a graph vocabulary; the application profile supplies the obligations that make an exported graph reconstructable.

Key Changes

From citation presence to exact source state

The first change is from asking whether an answer has a citation to asking whether the cited evidence can be recovered exactly. A URL alone may resolve to a different representation tomorrow. The profile therefore models a logical source, a dated **source snapshot**, and a quoted evidence span as distinct entities. The snapshot is a specialization of the logical source. PROV says a specialization shares the general entity's aspects while adding more specific ones (www.w3.org).

Fixity and identity serve different purposes. A persistent identifier locates a named version; a digest detects byte change. The Library of Congress PREMIS dictionary defines fixity as evidence that a digital object did not change between two times (www.loc.gov). NIST likewise describes message digests as a way to detect later changes (csrc.nist.gov). Software Heritage identifiers bind software objects to cryptographic hashes of their properties (docs.softwareheritage.org). Neither a digest nor an identifier proves the source is accurate. Together with the observation time and locator, they make the referenced state testable.

The minimum source-state export therefore includes:

- **Logical source identifier:** a stable Internationalized Resource Identifier or equivalent scoped identifier.
- **Snapshot identifier:** a distinct identifier for the observed representation.
- **Version locator:** a version-specific DOI, revision identifier, archive locator, or declared absence.
- **Observation time:** the time the producing system obtained the representation.
- **Media type and language:** sufficient interpretation context for the retained bytes or text.
- **Content digest:** algorithm identifier plus digest value, not an unlabeled hash.
- **Evidence span:** verbatim text plus page, section, byte, character, or selector coordinates.
- **Access outcome:** found, not found, blocked, or not applicable, always with a reason.

From a flat citation to typed derivation

The second change is to represent the production path. In PROV, generation and usage connect activities to produced and consumed entities, while derivation relates a resulting entity to an input (www.w3.org). An LLM context export can use these relations without claiming that derivation proves semantic support. Attribution research similarly treats source corroboration as a separate judgment (direct.mit.edu).

The profile distinguishes at least four activities:

- **Observation:** acquires the identified source snapshot at a recorded time.
- **Retrieval:** selects a snapshot or passage for a particular answer run.
- **Extraction:** converts the evidence span into a normalized claim.
- **Generation:** uses claims or passages to produce the answer artifact.

These activities must not be collapsed when separate events are observable. A system may omit an activity it genuinely cannot observe, but it must emit an **unknown reason** rather than fabricate a node or silently leave a field empty. This makes epistemic absence machine-readable.

From provenance content to accountable assertion

The third change records who asserted the lineage. PROV attribution ascribes an entity to an agent, while association can express activity responsibility (www.w3.org). The export should identify the software service, configured model release when known, organization, and human reviewer when one participated. It should not imply that a model itself bears organizational accountability. Human attribution studies used **five annotators per example**, evidence that support judgments can require multiple reviewers (direct.mit.edu).

A named bundle contains the lineage statements. A second bundle-level assertion identifies the exporting agent, profile version, creation time, and validation report. This is **provenance of provenance**: the consumer can distinguish what happened from who made the claim about what happened. A malicious or mistaken exporter can still assert false lineage. PROV-N explicitly warns that untrusted documents may mislead consumers (www.w3.org).

The Answer-Evidence Application Profile

W3C's Profiles Vocabulary defines a profile as a specification that constrains, extends, combines, or guides the use of other specifications (www.w3.org). The rules below are therefore this report's proposed **LLM Answer Evidence Profile 1.0**, not requirements of PROV-DM. Its versioned identifier follows the same general approach as current RO-Crate profile guidance (www.researchobject.org).

Table 1 maps the fields that a receiver needs to reconstruct an answer. “Required” means required by this profile, even where base PROV permits omission. Explicit occurrence constraints are established application-profile practice (www.dublincore.org).

Export field	PROV mapping	Cardinality	Validation rule
Profile and export ID	Bundle entity and identifier	Exactly 1 each	Absolute, stable identifiers; profile uses a versioned identifier.
Answer artifact	Entity	Exactly 1	Contains answer text or content-addressed retrievable representation.
Atomic claim	Entity, derived from evidence span	1 or more	Each answer assertion maps to at least one claim; claim boundaries are explicit.
Logical source	Entity	1 or more per supported claim	Stable locator and source type are present.
Source snapshot	Entity, specialization of logical source	Exactly 1 per evidence use	Observation time, version locator or unknown reason, media type, and digest are present.
Evidence span	Entity, part of source snapshot	1 or more per supported claim	Verbatim quotation plus resolvable selector or page locator; digest covers quoted bytes or normalized text.
Observation	Activity using logical source and generating snapshot	Exactly 1 per snapshot	End time is present and not later than downstream generation.
Retrieval or selection	Activity using snapshot or span	1 or more	Query or selection identifier is present; hidden text need not be exported if policy forbids it.
Transformation	Extraction or generation activity	1 or more	Typed usage and generation relations connect inputs and outputs.
Responsible identity	Agent associated with activity	1 or more per activity class	Stable agent ID and role are present; unknown identity carries a reason.
Lineage assertion	Named bundle attributed to exporting agent	Exactly 1	Bundle generation time, asserting agent, and profile version are present.

Export field	PROV mapping	Cardinality	Validation rule
Unknown declaration	Typed entity or attribute	0 or more	Status is not found, blocked, or not applicable, with reason and observation scope.

The table makes two design decisions explicit. First, a **claim is an entity**, not merely a string property on the answer, so that multiple answers can derive from or revise it. Second, an evidence span is separate from the source snapshot, so the export can distinguish source fixity from the precise support selected within it. Dublin Core application-profile guidance recognizes explicit occurrence constraints for repeatability (www.dublincore.org). That discipline is necessary here because an unconstrained graph can be valid PROV and still fail reconstruction.

Relation choices

The profile uses relations narrowly:

- **wasDerivedFrom:** the claim derives from an evidence span, and the answer derives from one or more claims.
- **wasGeneratedBy:** every snapshot, claim, answer, and lineage bundle identifies its producing activity.
- **used:** observation, extraction, and generation activities identify their input entities.
- **wasAttributedTo:** the bundle and answer identify their accountable exporting or publishing agent.
- **wasAssociatedWith:** each observable activity identifies the responsible system or person.
- **specializationOf:** a dated snapshot specializes the logical source.
- **wasRevisionOf:** a new claim or answer state revises an earlier state when the same artifact is intentionally updated.
- **alternateOf:** two representations present aspects of the same thing without asserting that either revises the other.

PROV defines alternates as presenting aspects of the same thing, and revision as a subtype of derivation whose result is a revised version (www.w3.org). Exporters should not use these relations interchangeably.

Implementation Considerations and Process Changes

The profile is serialization-neutral at the conceptual layer. PROV-O, PROV-XML, and PROV-N provide different encodings of the model, while JSON-LD 1.1 has its own processing specification. The dedicated PROV-JSONLD document is a **25 August 2024 Member Submission**, not a W3C Recommendation (www.w3.org). A contract may select it, but must accurately state that status. It should separately identify the canonical JSON procedure, such as the deterministic property ordering specified by RFC 8785, and state whether a digest covers HTTP representation data or content (www.rfc-editor.org).

Table 2 is a minimal worked graph. Identifiers are illustrative, while the relation pattern is normative PROV vocabulary constrained by the proposed profile.

Identifier and type	Required content	Required relations
ex:source (Entity)	Logical document identifier	General target of `specializationOf`
ex:snapshot-20260922 (Entity)	Locator, observed time, media type, digest	`specializationOf ex:source`; `wasGeneratedBy ex:observe`
ex:span-17 (Entity)	Verbatim quote, selector, span digest	`wasDerivedFrom ex:snapshot-20260922`
ex:claim-4 (Entity)	Atomic proposition and language	`wasDerivedFrom ex:span-17`; `wasGeneratedBy ex:extract`
ex:answer-9 (Entity)	Output text and answer-run ID	`wasDerivedFrom ex:claim-4`; `wasGeneratedBy ex:generate`
ex:observe (Activity)	Observation end time	`used ex:source`; associated with acquisition agent
ex:extract (Activity)	Transformation type and end time	`used ex:span-17`; associated with extraction agent
ex:generate (Activity)	Generation end time and configuration identity	`used ex:claim-4`; associated with generation agent
ex:lineage-bundle (Bundle)	Profile version and export time	`wasAttributedTo ex:exporter`; contains all statements above

This graph is intentionally small. It does not expose prompts, private context, or implementation topology. It establishes what an independent receiver needs to connect evidence to a claim and a claim to an answer. Where disclosure policy blocks a value, the export preserves the node and relation shape while declaring the restriction and reason. That is more informative than removing the edge, which would make “not observed,” “not retained,” and “not permitted” indistinguishable.

Serialization and fixity decisions

Implementers should publish the following choices with the profile:

- **Canonical form:** state whether digests cover source bytes, normalized text, a canonical JSON form, or a canonical RDF dataset.
- **JSON canonicalization:** RFC 8785 constrains input and deterministically sorts object properties (www.rfc-editor.org).
- **RDF canonicalization:** name the algorithm and whether comparison targets graph isomorphism or lexical identity.
- **HTTP digest semantics:** distinguish representation data from content because RFC 9530 defines separate fields (www.rfc-editor.org).
- **Timestamp lexical form:** specify timezone, fractional-second precision, and normalization. RFC 3339 lexical ordering assumes uniform fractional-second digits (www.rfc-editor.org).
- **Profile versioning:** use a versioned profile identifier. RO-Crate guidance recommends a major.minor version in the profile URI (www.researchobject.org).
- **Identifier persistence:** if software is evidence, a Software Heritage identifier uses strong cryptographic hashes over object properties (docs.softwareheritage.org).

These details are not cosmetic. Python `prov` documentation notes that an original non-whole-minute timezone offset is not preserved through a serialization round trip (prov.readthedocs.io). A round trip can retain the instant while losing lexical identity. The conformance report must state which identity notion it tests.

Conformance and Reconstruction Testing

The conformance suite should contain a public fixture, expected normalized graph, mutation set, and machine-readable results. **SHACL**, the Shapes Constraint Language, is suitable for cardinality and datatype checks on PROV-O exports, provided the profile states its supported feature subset. PROV constraint validation remains a separate layer, covering uniqueness, event ordering, impossibility, and type constraints (www.w3.org). Round-trip results must also disclose known lexical losses such as changed timezone offsets (prov.readthedocs.io). Exact reproducibility may additionally depend on capturing execution order (openreview.net).

The test sequence is:

1. **Parse:** load the canonical fixture in serializer A and serializer B.
2. **Validate base PROV:** check each graph against PROV constraints applicable to its representation.
3. **Validate profile:** apply cardinality, datatype, identifier, and time-order rules.
4. **Round trip:** serialize A to B's format and back, then compare normalized graphs.
5. **Reconstruct:** recover every required fixture fact using only the exported package.
6. **Mutate:** change one source version, alternate representation, attribution, timestamp precision, or unknown reason at a time.
7. **Detect:** require the expected graph difference and validation outcome for every mutation.
8. **Report:** publish tool versions, profile version, fixtures, differences, and unresolved identifiers.

Table 3 is the conformance matrix to publish for two independent paths. The cells state acceptance criteria, not unmeasured product results.

Test	Path A: JSON-LD to RDF to JSON-LD	Path B: PROV-N to model to PROV-N	Required report
Parse	No syntax or context errors	No syntax or namespace errors	Parser name, version, warnings
Base validity	PROV-O plus applicable PROV constraints pass	PROV constraints pass	Constraint identifiers and failures
Relation preservation	Every typed edge survives RDF normalization	Every qualified relation survives model round trip	Added, removed, or weakened edges
Timestamp precision	Instant and declared lexical precision recover	Instant and declared lexical precision recover	Each changed timestamp field
Identifier resolution	All required IRIs resolve or carry declared reason	All required qualified names expand deterministically	Unresolved identifier list
Source fixity	Snapshot and span digests verify	Snapshot and span digests verify	Algorithm, expected and actual digest

Test	Path A: JSON-LD to RDF to JSON-LD	Path B: PROV-N to model to PROV-N	Required report
Unknown semantics	Status and reason survive	Status and reason survive	Missing or changed reason codes
Normalized identity	Canonical RDF datasets are isomorphic	PROV normal forms are equivalent	Comparison algorithm and result

The interpretation must remain field-level. A single pass percentage can conceal loss of attribution while preserving less consequential labels. PROV equivalence can be tested at the normalized graph layer, but the answer-evidence profile additionally requires semantic identity for its extension fields. RFC 3339 warns that lexical timestamp ordering depends on uniform fractional-second precision (www.rfc-editor.org).

Mutation expectations

- **Source revision:** create a new snapshot and connect it with `wasRevisionOf`; the earlier claim remains reproducible.
- **Alternate representation:** connect HTML and PDF snapshots with `alternateOf`; do not infer byte identity.
- **Changed attribution:** replace the asserting agent; normalized graphs must differ even if evidence is unchanged.
- **Changed span:** preserve source snapshot but update the span entity and all downstream derivations.
- **Timestamp truncation:** fail precision preservation even when the instant remains comparable.
- **Blocked locator:** retain the entity and change access status plus reason; do not erase the claim silently.
- **Unknown digest algorithm:** fail fixity validation because the digest cannot be interpreted.
- **Orphan claim:** fail reconstruction when a supported claim lacks a path to an evidence span.

Data Analysis and Evidence

The quantitative core of this proposal is **reconstruction completeness**:

```
completeness = recovered required facts / required facts in the fixture
```

If a fixture contains **48 required facts** and a receiver recovers **46**, completeness is **95.8%**. This is a labeled hypothetical calculation, not a measured implementation result. The report must also name the two missing facts. Field-weighted variants may be published, but the unweighted fraction remains necessary so weights cannot hide omissions.

Existing evidence shows why syntactic success is insufficient. A PROV document can satisfy base constraints while omitting the source snapshot, evidence span, or asserting identity required by this profile. Similarly, ALCE observed that correctness and citation quality plateaued at different context counts (arxiv.org). One successful validation result should not be generalized to another evaluation dimension.

MatPROV offers a recent, bounded reuse case. Its evaluation sampled **30 papers containing 44 procedures** and used expert-created PROV-JSONLD ground truth (arxiv.org). Its best one-shot example improved structural F1 by **6.4%** and parametric F1 by **7.6%** over zero-shot prompting (arxiv.org). The paper also omits agent nodes because experimenters or institutions are treated as implicit (arxiv.org). It therefore demonstrates useful PROV-shaped extraction, but not conformance to this profile's accountable-agent requirement.

Citation research measures other constructs. ALCE separates fluency, correctness, and citation quality (arxiv.org) and uses a natural-language inference model to test whether citations entail statements (arxiv.org). It found correctness plateaued at one passage and citation quality at three in one experiment (arxiv.org). TREC's 2024 RAG Track separately evaluated nugget coverage, citation support, fluency, and retrieval (trec-rag.github.io). These studies support a multidimensional report, not a claim that provenance completeness measures answer truth.

Case Studies and Real-World Examples

MatPROV as a bounded reuse case

MatPROV used prompts designed for compliance with a PROV-JSONLD schema (arxiv.org). Its graph starts from entity, activity, and agent node types, although the published extraction omits agents (arxiv.org). For an answer-evidence export, that omission becomes a declared conformance failure rather than an implicit modeling choice. The example is valuable precisely because it exposes the difference between reusing PROV-DM concepts and satisfying a particular application profile.

Attribution benchmarks as complementary evidence

Attributable to Identified Sources defines attribution as corroboration by an identified source (direct.mit.edu). Its human evaluation used **five annotators per example**, reflecting difficult judgment boundaries (direct.mit.edu). MIRAGE further distinguishes plausible self-citation from faithful reflection of which context a model used (aclanthology.org). Provenance and attribution should therefore be evaluated side by side: one tests reconstructable lineage, the other tests whether evidence actually supports output text. Source quality remains a complementary evaluation rather than an implication of attribution (direct.mit.edu).

Microsoft Research's FakeWiki benchmark adds a different layer, training-data attribution. It contains **3,537 fabricated Wikipedia-style articles** with controlled provenance (www.microsoft.com). Its reported supervised model raised mean Recall@10 from **35.0 to 52.2** relative to the strongest retrieval baseline (www.microsoft.com). This measures recovery of known training provenance, not the answer-evidence contract defined here.

Implications and Future Directions

An enterprise interchange decision should begin with the reconstruction test, not with a preferred graph technology. A **passing exporter** publishes its profile version, fixture, validators, two-path matrix, and every missing field. A **passing importer** reconstructs the same required facts and preserves the intended relation types. A system that produces valid PROV but cannot recover the observed source state should be described accurately as PROV-compatible but nonconforming to this answer-evidence profile.

Four boundaries remain important:

- **Truth:** lineage says where a claim came from, not whether the claim is correct.
- **Authority:** attribution names an agent, but does not prove that agent is competent or authorized.
- **Permission:** a retrievable source and a digest do not establish a right to retain or disclose it.
- **Legal sufficiency:** technical conformance does not determine whether evidence satisfies a jurisdiction, contract, or proceeding.

FAIR principle R1.2 calls for data and metadata to be associated with detailed provenance (www.nature.com). That broad objective is compatible with this narrow profile, but it does not replace field-level tests. Likewise, as of September 2026, ISO/IEC FDIS 5181 on data provenance remained in approval rather than publication as an International Standard (www.iso.org). Implementers should publish exact conformance targets and versions instead of using “standards-based” as an undifferentiated claim.

Future profile revisions should remain additive where possible. New evidence-selector types, canonicalization algorithms, or agent roles can be introduced without weakening the six reconstruction facts. Crossref's controlled relationship vocabulary illustrates why typed version and derivation links are preferable to free text (www.crossref.org). Every revision should ship with migrations, old and new fixtures, and explicit expected graph differences.

Frequently Asked Questions (FAQs)

Is W3C PROV an LLM data provenance standard?

No. PROV is a general provenance family. It supplies concepts and relations that an LLM context provenance profile can constrain. The proposed answer-evidence profile adds mandatory source-state, span, digest, time, unknown-reason, and conformance rules that PROV-DM does not mandate.

Is a citation enough for LLM answer traceability?

No. A citation may identify a page without preserving the observed version, selected span, transformation, or asserting identity. Traceability requires a typed path from answer to claim to evidence span to source snapshot, plus responsible agents and times.

Does PROV-O require JSON-LD?

No. PROV-O is an OWL 2 representation of PROV and can use multiple RDF serializations. JSON-LD can be selected by an application contract, but PROV-DM does not mandate it. The publication status of the selected PROV-JSONLD mapping should be stated precisely. Thus, **PROV-O for generative AI** is a modeling choice, not a mandated wire format.

How should an organization audit LLM context?

It should test an exported fixture through two independent processing paths, validate base PROV and profile constraints, reconstruct all required facts without producer access, mutate one fact at a time, and publish field-level differences. The audit should report completeness and omissions, not only a pass percentage. An

observable export guarantee can be checked against the source, excerpt, locator, and dates actually delivered.

Does provenance explain why the model produced an answer?

Not necessarily. Provenance can record context supplied and transformations observed. Behavioral explanation may require model-specific causal evidence. A 2026 metric audit distinguishes provenance or topicality, generated-answer attribution, and fact-check entailment as separate constructs (arxiv.org).

What must a conformance publication contain?

At minimum: the versioned profile, canonical fixture, expected normalized graph, shape or schema rules, serializer and validator versions, mutation fixtures, complete results, reconstruction numerator and denominator, and a list of each unresolved or lost field.

Conclusion

W3C PROV provides the right conceptual vocabulary for **LLM context provenance**, but independent reconstruction requires a stricter application profile. The decisive export contains identifiable answer and claim entities, an exact source snapshot and supporting span, typed observation and transformation activities, responsible agents, observation and generation times, and a named bundle whose own attribution is explicit.

Conformance then becomes observable. Two implementations parse the same fixture, exchange it through different serializations, compare normalized graphs, reconstruct every required fact, and detect controlled mutations. The reported completeness fraction is useful only alongside the identity of every missing field. Base PROV validity, profile validity, round-trip preservation, and evidential support are separate results. MatPROV's own warning that automated extraction can omit information illustrates why the missing-field list matters (arxiv.org).

This separation is the central design rule. PROV can make a chain of custody interoperable. Content digests can expose change (csrc.nist.gov). Persistent identifiers can name versions (zenodo.org). Attribution evaluations can test support. None of them alone establishes truth, authority, permission, explanation, or legal sufficiency. An export contract is defensible when it states exactly which of these questions it answers, publishes the fixture and profile version, and lets an independent system reproduce the answer-evidence path without privileged access to the producer.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. https://www.w3.org/TR/prov-dm
2. https://www.w3.org/TR/prov-n
3. https://support.datacite.org/docs/versioning
4. https://www.rfc-editor.org/info/rfc3339
5. https://www.w3.org/TR/prov-implementations

6. https://openreview.net/pdf?id=VrbDf3UDgv
7. https://arxiv.org/html/2509.01042
8. https://arxiv.org/abs/2305.14627
9. https://trec-rag.github.io/annoucements/2024-track-guidelines
10. https://direct.mit.edu/coli/article/49/4/777/116438/Measuring-Attribution-in-Natural-Language
11. https://zenodo.org/help/versioning
12. https://www.crossref.org/documentation/schema-library/markup-guide-metadata-segments/relationships
13. https://www.nature.com/articles/sdata201618
14. https://www.loc.gov/standards/premis/v3/premis-3-0-final.pdf
15. https://csrc.nist.gov/pubs/fips/180-4/upd1/final
16. https://docs.softwareheritage.org/devel/swh-model/persistent-identifiers.html
17. https://www.w3.org/TR/dx-prof
18. https://www.researchobject.org/ro-crate/specification/1.3/profiles.html
19. https://www.dublincore.org/specifications/dublin-core/application-profile-guidelines
20. https://www.w3.org/submissions/2024/SUBM-prov-jsonld-20240825
21. https://www.rfc-editor.org/rfc/rfc8785.html
22. https://www.rfc-editor.org/rfc/rfc9530.html
23. https://prov.readthedocs.io/en/latest/reference/conformance.html
24. https://www.w3.org/TR/prov-constraints
25. https://arxiv.org/html/2305.14627
26. https://aclanthology.org/2024.emnlp-main.347
27. https://www.microsoft.com/en-us/research/publication/datadignity-training-data-attribution-for-large-language-models
28. https://www.iso.org/standard/80971.html
29. https://arxiv.org/abs/2606.23915

THE PUBLISHER

About Reliable Context

Reliable Context is a system of record for organizations that fuse private and public information and must defend every value they act on.

The guarantees

One canonical record per real thing: duplicates resolve by rule, ambiguity is shown rather than hidden, and related organizations relate without ever being merged. A verbatim, cited source and an observation date sit behind every value. Unknowns are declared as not found, blocked or not applicable with a reason, never left as an empty cell. Any past state of the record is reproducible, together with what changed since.

Where it runs

It runs inside the customer's perimeter, one database per organization. Permitted sources, model routing, reader clearance and retention are set as the customer's own policies, and every value is exportable with its chain of custody for an auditor or regulator.

What it is not

It is not a vector database or a RAG framework, not a data catalog, MDM or data-quality suite, not a CRM or data-enrichment vendor, and not a chatbot or AI assistant. Where a reader needs one of those, the honest answer is to say so and point them there.

Visit

See Reliable Context.

Reliable Context is in production use in two sectors; more than six thousand of its identity decisions were made by rule and none has been reversed. Articles published here explain general practice, cite their primary and academic sources, and name the jurisdiction of any regulatory rule they describe. They are not legal, regulatory or compliance advice for a specific organization, and requirements change.

Website: https://reliablecontext.com

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.