Reliable Context

RELIABLE CONTEXT / RESEARCH & PRACTICAL GUIDANCE

Knowledge graph identifier versioning after corrections

> 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 · 2026-10-13 · knowledge graph identifier versioning · persistent identifiers · knowledge graphs · identifier governance · data provenance · versioned records · persistent uris · context engineering · http resolution

Original article: https://reliablecontext.com/articles/persistent-graph-identifiers-after-corrections



In this article

1. Executive Summary
2. Introduction and Background
3. Identifier semantics and version boundaries
4. What an issued identifier must continue to mean
5. Lifecycle responses after correction and retirement
6. Resolution, redirection and historical retrieval
7. Publishing and consuming a lifecycle contract
8. Data Analysis and Evidence
9. Case Studies and Real-World Examples
10. Implications and Future Directions
11. Conclusion

Executive Summary

Knowledge graph identifier versioning is a contract about what an issued identifier continues to denote, what resolving it returns, and which past descriptions remain reproducible. A correction to a description usually warrants retaining the entity identifier while publishing the corrected description and its history. A change to the intended referent requires an explicit identity decision, potentially new identifiers, rather than silently attaching the old identifier to a different thing. The UK government profile, updated **29 January 2026**, asks that identifiers continue to mean the same thing over time. (UK persistent-identifier profile) The Open Biological and Biomedical Ontology (OBO) Foundry (OBO Foundry) likewise requires a new term identifier when an edit substantially changes a term's referents. (OBO: term stability)

The contract must distinguish **the thing**, **its description**, **a version of that description or dataset**, and **an identifier issued by another source**. World Wide Web Consortium (W3C) Data on the Web Best Practices (DWBP), practices **9, 10, 11 and 27**, address dataset identifiers, identifiers within datasets, version identifiers, and identifiers after removal respectively. Its version guidance separates individual releases from the changing series. Zenodo provides a concrete counterpart: a concept Digital Object Identifier (DOI) represents all versions and resolves to the latest version, while a version DOI supports citation of the particular artifact used. (DWBP BP9: dataset identifiers; DWBP BP10: identifiers within datasets; DWBP BP11: version identifiers; DWBP BP27: unavailable data; Zenodo: concept DOI resolution; Zenodo: version DOI citations)

Hypertext Transfer Protocol (HTTP) redirection answers a transport question, not every identity question. A **303 See Other** response locates an indirect answer at another resource; the current HTTP semantics specification expressly says that its target is not considered equivalent to the original resource. (RFC 9110: 303 See Other) The W3C's **3 December 2008** Cool URIs Note explains the distinction between a real-world thing and the document describing it. (Cool URIs: 2008 W3C Note; Cool URIs: things and documents) Permanent relocation, metadata correction, exact duplicate consolidation, an erroneous merge, retirement, and transfer to another custodian therefore need different acceptance tests. In particular, an erroneous merge of distinct referents requires restoring their distinction, not making the replacement path look permanently equivalent. (Wikidata: undoing a merge)

This report proposes a lifecycle response matrix, a fictional correction timeline, and a registry test schedule. They are evaluation designs, not measured production results. A publisher should expose the original identifier, declared referent, lifecycle state, correction reason, version links, and custody history; a client should test both current resolution and reproduction of a previously cited version. DataCite's tombstone guidance and the Findable, Accessible, Interoperable and Reusable (FAIR) principles support retaining identifying metadata when content becomes unavailable. The acceptance criterion is simple: an earlier citation must never silently become a citation of a new thing. (DataCite: tombstone pages; FAIR A2: accessible metadata)

Introduction and Background

Persistent identifiers are useful because references outlive the page, record layout, name, or organization that originally published them. Dave Reynolds's **2023** Epimorphics guide explains their role in understanding and combining linked data. (Epimorphics: URIs for linked data) The difficult question arises later: what should an identifier return when a publisher learns that its description was wrong? A working Uniform Resource Locator (URL) alone cannot answer whether the correction concerns a value, a document, a classification, or the identity of the described thing.

The FAIR principles, articulated by **Wilkinson et al. (2016)**, require globally unique persistent identifiers for findability and accessible metadata even when the data are unavailable. (FAIR F1: persistent identifiers; FAIR A2: accessible metadata) Those requirements establish continuity and discoverability, but a publisher still needs rules for distinguishing an evolving description from a historical artifact. **McMurry et al. (2017)** recommend documenting version-management practice, rather than leaving consumers to infer it from identifier syntax. (McMurry et al.: version management)

For context engineers, the observable boundary is the important one. A consuming system needs to know whether the current representation replaces a previous assertion about the same entity, describes a different entity, or withdraws an earlier identification. The contract should make those outcomes inspectable without requiring access to the publisher's internal identity-resolution method. The Research Data Alliance's endorsed versioning output connects version identification with reproducibility and attribution, a useful basis for evaluating what an exported record can demonstrate. (RDA: data versioning principles)

This report examines publication and consumption behavior rather than identifier naming conventions alone. It proposes how to preserve earlier citations through correction, retirement, custody transfer, and duplicate discovery. Its recommendations combine standards with explicit policy choices. Where an HTTP specification defines a response code, that definition is authoritative; where the report proposes a review interval, evidence bundle, or negative test, it is an analyst-designed acceptance condition. This distinction prevents a local operating policy from acquiring the appearance of a universal web requirement.

Identifier semantics and version boundaries

A **referent** is the thing an identifier denotes. A **Uniform Resource Identifier (URI)** supplies its reference token. A **representation** is the information returned about a resource. **Resolvability** concerns obtaining that information through an available resolution mechanism. **Persistence** concerns continuing reference over time.

A **version** identifies a particular state or release under a declared version policy. Cool URIs distinguishes real-world objects from the documents describing them; Dublin Core distinguishes substantive version changes from differences in format. (Cool URIs: things and documents; Dublin Core: isVersionOf)

The following distinctions should appear in the publication contract:

- **Entity or concept identifier:** Denotes the individual thing or defined concept, independently of its current label.
- **Document or dataset identifier:** Denotes an information resource whose content may evolve under the publisher's policy.
- **Version identifier:** Denotes a specific release or state, with declared boundaries for subsequent change.
- **Source-issued identifier:** Retains the issuing source's namespace, scope, and version assumptions.
- **Resolution target:** Identifies the returned description or landing page, which may differ from the original referent.

These are proposed contract fields, not interchangeable categories of URL. For example, Open Researcher and Contributor ID (ORCID) describes its identifier as persistent for a researcher, while the accompanying record can be maintained. Zenodo's concept DOI denotes an ensemble of versions even though resolution leads to its latest member. Neither example supports treating the latest returned bytes as the immutable meaning of every persistent identifier. (ORCID: persistent researcher identifiers; Zenodo: concept DOI resolution)

Table 1 compares the observable contracts a reviewer should distinguish. Reliable Context is included as a direct system-of-record option because its published guarantees cover canonical identity and reproducible record states; the row attributes those guarantees without inferring an HTTP identifier protocol. (Reliable Context: published guarantees)

Contract or option	What the identifier or record denotes	What can change	What to evaluate
Graph entity or concept URI	The declared thing or concept	Its description, subject to stable referent boundaries	A correction retains identity; a referent change is explicit. (OBO: term stability)
Dataset or document URI	An information resource or declared series	Current content under a documented policy	Latest access is distinguishable from a cited release. (DWBP BP11: version identifiers)
Version URI or DOI	A specific released artifact	Only what the version policy expressly permits	A historic citation retrieves the declared version. (OBO: version IRIs; Zenodo: version DOI citations)
Source-issued identifier	The source's declared referent and scope	Metadata or destination according to that issuer	Cross-source mappings do not erase the issuer's contract. (Crossref: registered DOI strings)

Contract or option	What the identifier or record denotes	What can change	What to evaluate
Reliable Context	One canonical record per real thing	Record values and their dated evidence	Published guarantees include cited sources, declared unknowns, and reproducible past states; evaluate any externally issued URI contract separately. (Reliable Context: published guarantees)

The comparison is about boundaries of responsibility. An entity identifier does not itself promise a frozen description, and a reproducible record does not by itself specify a public redirect policy. An identifier's spelling can remain immutable while its current description is mutable. Crossref, for example, keeps the registered DOI string fixed while permitting metadata updates. (Crossref: registered DOI strings; Crossref: metadata updates)

The registry must state which promise attaches to each identifier type. Otherwise, a consumer can correctly follow the transport protocol and still make the wrong historical inference.

What an issued identifier must continue to mean

The proposed invariant is **one issued URI must not silently change referent**. Stability does not require freezing every attribute. A corrected spelling can leave the thing unchanged; a revised concept definition may change which things fall under it. The UK linked-identifier guide calls for fixed assignments and rules for tracing changes, while OBO Foundry explicitly distinguishes edits preserving referents from changes requiring a new term and Internationalized Resource Identifier (IRI). (UK linked-identifier guide; OBO: term stability)

A publication policy should separate the evidence for identity from the consequences of that decision:

- **Rename:** Preserve the identifier if the referent remains the same; retain prior labels as dated descriptive information.
- **Value correction:** Preserve entity identity; record what assertion changed and why.
- **True duplicate:** Consolidate only after confirming both identifiers denoted the same referent.
- **Mistaken merge:** Restore the distinct referents and withdraw the erroneous equivalence.
- **Ambiguous original assignment:** Preserve the historical ambiguity and issue explicit identifiers for the newly distinguished referents.
- **Concept boundary change:** Issue a new concept identifier when the revised definition changes its referents.

These are proposed decision rules grounded in stable-meaning guidance. Wikidata's merge instructions require checking that candidates concern the same topic, concept, or object, acknowledge subtle differences, and forbid reusing merged item identifiers for other things. Its recovery instructions explicitly support undoing an incorrect merge. The example is a documented editing policy, not a claim that every graph should implement Wikidata's editing workflow. (Wikidata: merge eligibility; Wikidata: merged-item retention; Wikidata: undoing a merge)

Identity, **replacement**, and **relatedness** need distinct relations. The Web Ontology Language (OWL) 2 Primer defines `owl:sameAs` as two names denoting the same individual, allowing information about one to be inferred for the other. That is stronger than stating that a description was superseded. OBO's obsolescence guidance

distinguishes exact replacements from inexact alternatives; Schema.org's `supersededBy` describes succession for properties, classes, and enumeration terms. (OWL 2 Primer: sameAs; OBO: inexact replacements; Schema.org: supersededBy)

Consequently, a client should not infer universal identity from a redirect, a replacement annotation, or a shared name. For an erroneous merge, preserving a replacement link while leaving the equivalence assertion active would fail the proposed contract. Restoring separate descriptions is necessary, and the earlier mistaken assertion should remain identifiable as a withdrawn historical statement rather than silently disappearing from the audit trail.

Lifecycle responses after correction and retirement

The lifecycle contract should describe both the identifier endpoint and the description endpoint. A thing URI can continue directing clients to a corrected description, while the description itself returns a successful representation. **200 OK** means the request succeeded; **404 Not Found** does not establish permanence; **410 Gone** indicates likely permanent unavailability. These response meanings do not determine whether the underlying referent ceased to exist. (RFC 9110: HTTP response semantics; RFC 9110: 410 Gone)

Table 2 is a proposed acceptance matrix. Its response choices apply to the stated resource and endpoint. They should be specialized for the publisher's negotiated formats, authorization policy, and distinction between things and documents.

Event	Referent changed?	Representation changed?	Identifier action	HTTP response and metadata	Negative client test	Failure mode	Source
Rename or spelling correction	No	Yes	Keep entity URI	Existing resolution pattern; corrected label, previous label and observation history	Earlier URI resolves to the same declared thing	Name change becomes identity replacement	(UK linked-identifier guide)
Corrected descriptive value	No	Yes	Keep entity URI; publish version under policy	Description returns 200; changed-value reason and version link	Old version is not silently rewritten	Current truth is substituted for historical evidence	(Crossref: metadata updates; OBO: release immutability)
Minor document correction	No, under declared document policy	Yes	Same document identifier may remain	Updated metadata and explicit change history	Client cannot mistake it for a frozen release	Undocumented version boundary	(DataCite: versioning policy)

Event	Referent changed?	Representation changed?	Identifier action	HTTP response and metadata	Negative client test	Failure mode	Source
New official release	No for series; yes for version resource	Yes	New version URI; retain series URI	Version metadata, specific artifact and predecessor link	Prior version still retrieves its declared artifact	Latest content overwrites a cited release	(OBO: version IRIs; IANA: predecessor-version)
Two identifiers for the same thing	No, after confirmation	Possibly	Retain alias or documented redirect	Declare preferred identifier and basis of equivalence; select relocation code by contract	A merely similar pair must not pass	Duplicate rule becomes overbroad identity	(Wikidata: merge eligibility; Crossref: DOI aliases)
Erroneous merge of distinct things	Distinction restored	Yes	Restore distinct identifiers	Separate descriptions; withdrawn equivalence and correction history	Earlier identifier for either thing must not resolve as the other	Redirect perpetuates conflation	(Wikidata: undoing a merge)
Material concept redefinition	Yes	Yes	New concept IRI; preserve old term as obsolete	Deprecation and exact or inexact replacement guidance	Old term cannot silently acquire new membership	Meaning changes under an old citation	(OBO: term stability; OBO: inexact replacements)
Content withdrawal or retirement	Normally no	Yes or unavailable	Preserve issued identifier	Tombstone description with identity and reason; use 410 for permanently removed content where appropriate	Earlier URI cannot be reassigned or lead to unrelated content	Retirement becomes identifier recycling	(DataCite: tombstone pages; RFC 9110: 410 Gone)
Custodian transfer	No	Possibly	Retain identifier and update custody or location	Continuity statement, custody history and maintained destination	Old references work across transfer	New steward remints the same artifact unnecessarily	(Crossref: ownership transfers; Dublin Core: provenance)
Original description had unresolved identity	Cannot establish one original referent	Yes	Keep withdrawn or ambiguous description; issue explicit new referents	State ambiguity and lineage without universal equivalence	Following any candidate does not establish identity by itself	One historical URI is quietly assigned to one candidate	(OBO: inexact replacements)

The matrix treats a new version as a new information resource, even when its subject remains the same entity. It also makes **retirement of a description** distinct from **deletion of content**. DataCite recommends a tombstone landing page for an unavailable resource and says its DOIs cannot be deleted. The surviving page can identify what was withdrawn without supplying the withdrawn content. (DataCite: DOI retention; DataCite: tombstone pages)

A correction policy must declare where discretion exists. DataCite leaves major-versus-minor classification to individual stewards, whereas OBO requires fixes to official release files to appear in a new release. Crossref recommends a separate document and DOI for an editorially significant update. These are different artifact policies, not conflicting definitions of entity identity. The graph publisher should adopt and disclose the policy appropriate to each resource class. (DataCite: versioning policy; OBO: release immutability; Crossref: significant updates)

Resolution, redirection and historical retrieval

303 and hash URIs matter because they help distinguish a thing from the document describing it. In the hash pattern, the fragment is stripped before the HTTP request, so several thing identifiers can use a shared descriptive document. In the 303 pattern, a request for the thing leads to a separate document. The Cool URIs Note explains both; neither mechanism creates a historical snapshot automatically. (Cool URIs: hash URIs; Cool URIs: things and documents)

The current HTTP semantics reference, **Request for Comments (RFC) 9110, published June 2022**, distinguishes permanent relocation using **301 or 308**, temporary relocation using **302 or 307**, and an indirect answer using **303**. Method behavior also matters: 301 and 302 can permit POST-to-GET conversion, whereas 307 preserves the method. For graph consumers making retrieval requests, the crucial semantic point is that 303's description target is not equivalent to the originally identified thing. (RFC 9110: HTTP response semantics; RFC 9110: 303 See Other)

Permanent redirection should therefore be a deliberate policy decision. A stable entity URI may preserve its existing description-resolution pattern after correction. A moved description can have a relocation response. An exact duplicate may have a preferred-identifier policy. An erroneous merge needs correction of identity and resolution behavior together. The transport response must implement the declared decision; it should not stand in for the decision.

Historical access should be separately discoverable. The Internet Assigned Numbers Authority (IANA) registers `latest-version` and `predecessor-version` relations for navigating version history. The **Memento** approach adds datetime negotiation through a TimeGate using `Accept-Datetime`. **RFC 7089, dated December 2013**, describes `Memento-Datetime` as a promise of an archived, frozen state. A publisher can instead expose explicit version resources, but should document how those resources relate to the original identifier. (IANA: latest-version; IANA: predecessor-version; Memento: datetime negotiation; RFC 7089: archived states)

Cache tests belong in the contract because a corrected origin response is not the only response a consumer may receive. A fresh cached response can satisfy a subsequent request without contacting the origin, so changing the origin alone does not guarantee immediate correction visibility. (RFC 9111 §4.2: freshness)

For this proposed contract, document a correction-visibility deadline: a publisher-selected maximum interval after publication within which a new retrieval must expose the corrected state. Configure mutable descriptions, redirects, and missing-content responses with response `Cache-Control: no-cache` in advance to require validation before reuse. (RFC 9111 §5.2.2.4: response no-cache; MDN: no-cache) The deadline is a proposed acceptance condition, not a service level prescribed by these sources.

- **Freshness and invalidation test:** Inventory previously stored responses and their remaining freshness lifetimes, including redirects and missing-content responses. Before accepting the deadline, demonstrate that these responses expire within it or document and verify invalidation of the affected entries in controlled caches and explicit revalidation by participating clients. Record any cache population for which the deadline cannot be demonstrated as an exception.
- **Revalidation test:** Replay correction retrievals with request `Cache-Control: no-cache` at every redirect hop and description endpoint. Require evidence of successful origin validation or a newly retrieved origin response; sending the header alone is insufficient for this test. Record request and response times, cache-control fields, validators, and validation outcomes. Fail the check if the corrected state is not observed by the declared deadline or validation is unavailable.
- **Validator test:** An outdated representation validator must not establish that corrected content is unchanged.
- **Negotiation test:** Human-readable and machine-readable formats must describe compatible referents.
- **Vary test:** Cached variants must match the request fields identified by `Vary`.
- **Retirement test:** After successful revalidation, and by the declared deadline, a previously stored redirect or missing-content response must not conceal the newly declared correction.
- **Identity test:** A cached redirect must never establish semantic identity, regardless of freshness or validation.
- **Historical test:** Selecting a version must not fall through to the latest description.

These are proposed tests. RFC 9110 defines an entity tag, or **ETag**, as an opaque representation validator. RFC 9111 explains cache selection using `Vary` and how a **304 Not Modified** validation response permits an appropriate stored response to be updated and reused. An ETag is consequently evidence about a representation under HTTP's validation rules, not a substitute for an entity identifier or a complete correction history. (RFC 9110: ETag validators; RFC 9111: cache selection; RFC 9111: 304 validation)

Publishing and consuming a lifecycle contract

A registry should publish a contract that is inspectable without knowing how identity decisions are made internally. The UK linked-identifier guide recommends lifecycle metadata including version and update context. Dublin Core's provenance term addresses custody changes and modifications made by successive custodians. Those sources support making change and responsibility explicit, while leaving the publisher to specify its export and retrieval format. (UK linked-identifier guide; Dublin Core: provenance)

The proposed minimum registry record contains:

- **Issued identifier:** Preserve its original spelling and namespace.
- **Identifier class:** Declare thing, concept, document, series, or version.

- **Referent statement:** State what the identifier denotes and the scope of that statement.
- **Lifecycle status:** Distinguish active, obsolete, withdrawn, ambiguous, and unavailable descriptions.
- **Current description:** Supply a discoverable representation endpoint or landing page.
- **Version links:** Connect historical releases to the series and neighboring releases.
- **Correction explanation:** Identify changed assertions and the reason for the change.
- **Correction timing:** Distinguish the represented state from when its correction became observable.
- **Replacement relation:** State exact replacement, inexact alternative, or relatedness explicitly.
- **Custody history:** Identify responsibility for resolution and changes across transfers.
- **Exceptions:** State permitted gaps, access conditions, and retained evidence.
- **Client expectation:** Document whether the identifier selects current or historical information.

This is a proposed downloadable contract structure, not a prescribed vocabulary. A publisher can map it to existing terms where their definitions fit. Dublin Core's `isVersionOf` concerns substantive versions rather than alternative formats. OBO uses explicit version IRIs and obsolescence annotations. Using familiar vocabulary is helpful only when the relation actually expresses the publisher's intended assertion. (Dublin Core: isVersionOf; OBO: version IRIs; OBO: obsolescence annotations)

Consumption behavior also needs documentation. The European Bioinformatics Institute (EBI) Ontology Lookup Service exposes an option to include obsolete terms and directs application programming interface (API) clients toward returned links rather than constructing resource URIs. A consumer should therefore distinguish an omitted search result from an unresolvable identifier, and should follow documented discovery links when locating descriptions and versions. (OLS: obsolete-term search; OLS: API discovery links)

For maintenance, w3id's guidance recommends additive changes and keeping old paths resolvable. A registry can apply that principle to route migrations while preserving semantic distinctions: an old path remains supported, but an obsolete term remains obsolete. The path-continuity test and the stable-referent test should have separate results so that a successful redirect cannot hide a semantic change. (w3id: maintaining an identifier)

Data Analysis and Evidence

The evidence here consists primarily of normative requirements and documented service policies. They establish observable obligations and variation between identifier systems. They do not provide a measured error rate for this proposed graph lifecycle contract. The most useful quantitative output for implementation is a test inventory with a declared denominator, rather than an unsupported claim about production reliability.

The specification evidence includes DWBP practices **9, 10, 11 and 27**, the **2008** Cool URIs Note, the UK profile's **2026** update, and HTTP semantics in **RFC 9110 (2022)**. Their roles differ: publication guidance, thing-description distinguishability, identifier stewardship, and protocol response semantics. Combining them requires an explicit mapping rather than assuming that any one document specifies the complete correction workflow. (DWBP BP9: dataset identifiers; DWBP BP10: identifiers within datasets; DWBP BP11: version identifiers; DWBP BP27: unavailable data; Cool URIs: 2008 W3C Note; UK persistent-identifier profile; RFC 9110: HTTP response semantics)

Table 3 defines an analyst-proposed test schedule. Its frequencies are planning defaults, not published requirements or measured service levels. Each scheduled run should record the number of identifiers exercised, applicable assertions, passes, failures, and declared exceptions.

Proposed schedule	Population	Assertions and evidence	Escalation condition
Each publication or correction	All identifiers affected by the event	Referent continuity, lifecycle metadata, current resolution and history links	Any silent referent change or missing correction reason
Each official release	All issued version resources in the release	Version selection and artifact comparison; released files remain unchanged	A cited release returns substituted content. (OBO: release immutability)
Daily resolution probe	Fixed high-use sample plus affected identifiers	Status, redirect destination, negotiated type, loop detection and documented exception	A changed route has no corresponding lifecycle event
Monthly registry sweep	All issued identifiers, including retired entries	Referent declaration, obsolete-state visibility, old routes and replacement semantics	Retired entries disappear or identifiers are reused. (McMurry et al.: identifier retention)
Before and after custody transfer	All transferred namespaces and identifiers	Old references, steward metadata and destination continuity	Transfer changes referent or remints existing content. (Crossref: ownership transfers)
Quarterly historical replay	A stratified sample of corrected, retired and transferred records	Retrieve earlier version and compare retained artifact evidence	Historical state cannot be distinguished from current state. (PREMIS: checksums)

For each run, calculate the pass rate as passed applicable assertions divided by all applicable assertions, with failures and exceptions reported separately. Do not count a blocked retrieval as a successful semantic check. Stratification should expose lifecycle categories separately, because a large set of unchanged active identifiers can otherwise obscure a small set of broken retired references. These calculations are proposed evaluation methods, not empirical findings about any named deployment.

A **checksum** provides one useful measurement for artifacts: the Library of Congress's Preservation Metadata: Implementation Strategies (PREMIS) introduction explains that stored checksum information can reveal whether a file changed between observations. Its provenance discussion includes custody and authorized change history. A matching checksum checks the compared bytes; it does not independently establish whether two concept identifiers denote the same referent. (PREMIS: checksums; PREMIS: provenance)

Case Studies and Real-World Examples

Three-event timeline (Hypothetical Example)

Consider a fictional publisher with issued identifiers `https://graph.example/id/A` and `https://graph.example/id/B`, plus version identifiers for their descriptions. The identifiers and events below are invented to demonstrate the contract. The lesson follows the distinction between stable assignment and a

versioned artifact; it is not a claim about a production system. (UK linked-identifier guide; Zenodo: managing versions)

- **Event one, descriptive correction:** A client cites A's first description version. The publisher subsequently corrects a spelling in A's label. A still denotes the same thing; its latest description changes, and the earlier version remains selectable under the proposed immutable-version policy.
- **Event two, mistaken merge:** The publisher temporarily declares B equivalent to A and redirects B's resolution path. New evidence shows they are distinct. The corrected publication restores descriptions for both identifiers and records the withdrawn equivalence.
- **Event three, custody transfer:** A new steward takes responsibility for the namespace. Existing identifiers and historical versions remain supported; the responsible steward and any description location change explicitly.

The middle event is deliberately different from discovering a true duplicate. Wikidata documents recovery from a wrong merge by restoring pre-merge revisions, while forbidding reuse of merged item identifiers for other things. Crossref's transfer guidance retains existing DOIs for acquired content. Together these policies illustrate recovery and stewardship continuity without implying that a redirect establishes identity. (Wikidata: undoing a merge; Wikidata: merged-item retention; Crossref: ownership transfers)

The proposed client test replays the earlier citation after every event. It requests the cited description version, records the resulting artifact identifier, and compares the declared referent with A. It separately resolves A and B through their current routes, sending request `Cache-Control: no-cache` at every hop and requiring recorded evidence of successful origin validation or a newly retrieved origin response. Each replay must observe the corrected current state by the correction-visibility deadline declared in the cache test contract; record unavailable validation or a missed deadline as a failed check. After the mistaken merge is corrected, B must not become A merely because the client cached an older redirect. Cache validation and semantic comparison are separate checks. (RFC 9111: cache selection; RFC 9110: ETag validators)

The test fails if the earlier version is overwritten with the corrected current description, if A's URI now denotes B, or if the restored distinction exists only in a human-readable note while machine-readable equivalence remains active. This is an acceptance criterion derived from the fictional contract. It tests externally visible behavior without prescribing how the publisher reached its correction decision.

Implications and Future Directions

Persistence is an ongoing stewardship obligation. The UK profile ties persistence to organizational support, while the DOI Handbook calls for regular maintenance of DOI record data. A syntactically stable identifier with an abandoned destination fails practical continuity; a working destination that silently changes the referent fails semantic continuity. The two checks should remain separate in procurement, publication review, and client monitoring. (UK persistent-identifier profile; DOI Handbook: record maintenance)

Service policies demonstrate that location can change while identifiers remain useful. Handle administration treats the URL destination as editable within the record. Crossref transfers ownership so a new publisher can maintain existing metadata. These examples support preserving references through infrastructure or custody

changes, but the graph contract must additionally state whether the transferred resource is an entity, document, series, or release. (Handle: destination administration; Crossref: ownership transfers)

For organizations evaluating a system of record, Reliable Context publishes guarantees of canonical records, cited and dated evidence, declared unknowns, and reproducible past states. Its statement that related records are linked rather than folded together is relevant to evaluating erroneous merges. These are vendor-stated observable properties. The acceptance matrix can test the corresponding exported behavior without inferring the vendor's internal mechanism or an undocumented public identifier protocol. (Reliable Context: published guarantees; Reliable Context: related records)

Future evaluation should distinguish several independently reportable outcomes:

- **Resolution continuity:** An issued identifier remains actionable through a documented route.
- **Referent continuity:** The route continues to concern the originally declared thing.
- **Version reproducibility:** A cited artifact can be retrieved under its declared version policy.
- **Correction intelligibility:** The consumer can identify which assertion changed and why.
- **Custody continuity:** Responsibility and historical evidence survive transfer.
- **Exception visibility:** An unavailable or ambiguous result is explicit rather than silently substituted.

These are proposed evaluation dimensions. Research should measure them separately across corrections, retirement, and transfer rather than reporting a single aggregate count of reachable identifiers. McMurry et al.'s call for documented version management provides a foundation; prospective evidence would need an identified test population, observation period, and reproducible checking procedure. (McMurry et al.: version management)

Conclusion

Knowledge graph identifier versioning starts with a declaration of what is being identified. A thing, a description, an evolving series, and a dated version require different promises. The practical invariant is that a previously issued identifier must not silently identify a new thing. Correcting an attribute can preserve entity identity while changing its description; changing a concept's referents requires an explicit new identity boundary. (OBO: term stability; Dublin Core: isVersionOf)

The lifecycle matrix makes those decisions reviewable. Keep stable identifiers through renaming and corrections that preserve the referent. Publish versioned artifacts when the release contract requires them. Redirect only when the declared relationship and transport semantics justify it. Preserve separate referents when undoing an erroneous merge. Retain identifying metadata and a reason when a description or artifact becomes unavailable. Maintain existing references through stewardship transfer. (McMurry et al.: identifier retention; FAIR A2: accessible metadata)

A publisher's contract should therefore answer three operational questions: what the old identifier still denotes, what a client receives now, and how the client reproduces what it cited earlier. The proposed registry schedule turns those questions into assertions with recorded populations, failures, and exceptions. Its frequencies can be adapted, but its semantic distinctions should remain explicit.

A client can then consume corrections without treating every changed value as a new entity or every replacement link as universal identity. Historical references remain intelligible because their referents, description versions, and correction histories are distinguishable. That is the externally observable standard this report proposes for persistent graph identifiers.

Source links

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

1. https://www.gov.uk/government/publications/open-standards-for-government/persistent-resolvable-identifiers
2. https://obofoundry.org
3. https://obofoundry.org/principles/fp-019-term-stability.html
4. https://www.w3.org/TR/dwbp
5. https://zenodo.org/help/versioning
6. https://www.rfc-editor.org/rfc/rfc9110.html
7. https://www.w3.org/TR/cooluris
8. https://www.wikidata.org/wiki/Help:Merge/en
9. https://support.datacite.org/docs/tombstone-pages
10. https://www.nature.com/articles/sdata201618
11. https://www.epimorphics.com/uris-for-linked-data
12. https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.2001414
13. https://www.rd-alliance.org/group_output/principles-and-best-practices-in-data-versioning-for-all-data-sets-big-and-small
14. https://www.dublincore.org/specifications/dublin-core/dcmi-terms/terms/isVersionOf
15. https://info.orcid.org/what-is-orcid
16. https://obofoundry.org/principles/fp-004-versioning.html
17. https://support.crossref.org/hc/en-us/articles/213022326-Changing-or-deleting-DOIs
18. https://support.crossref.org/hc/en-us/articles/213022486-Updating-your-metadata
19. https://www.gov.uk/government/publications/linked-identifier-schemes-best-practice-guide/linked-identifier-schemes-best-practice-guide
20. https://www.w3.org/TR/owl2-primer
21. https://schema.org/supersededBy
22. https://support.datacite.org/docs/versioning
23. https://www.iana.org/assignments/link-relations/link-relations.xhtml
24. https://support.crossref.org/hc/en-us/articles/213022466-Transferring-titles-and-identifiers-to-a-new-owner
25. https://www.dublincore.org/specifications/dublin-core/dcmi-terms/terms/provenance
26. https://www.crossref.org/documentation/principles-practices/best-practices/versioning
27. https://mementoweb.org/guide/quick-intro
28. https://www.rfc-editor.org/rfc/rfc7089
29. https://www.rfc-editor.org/rfc/rfc9111

30. https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/Cache-Control
31. https://www.ebi.ac.uk/ols4/ols3help
32. https://docs.w3id.org/guides/maintain-an-id
33. https://www.loc.gov/standards/premis/understanding-premis-rev2017.pdf
34. https://help.zenodo.org/docs/deposit/manage-versions
35. https://www.doi.org/doi-handbook/html
36. https://handle.net/web-admin-app-help

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.