Reliable Context

RELIABLE CONTEXT / RESEARCH & PRACTICAL GUIDANCE

When a Data Contract Needs an Ontology: ODCS 3.2 Test

> 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 · data contract ontology · odcs 3.2 · semantic data contract · data governance · semantic interoperability · ontology · data contracts · authoritative definitions · knowledge engineering

Original article: https://reliablecontext.com/articles/data-contract-ontology-odcs-test



In this article

1. Executive Summary
2. Introduction and Background
3. Definitions and the smallest sufficient artifact
4. The consumer-decision acceptance test
5. Key changes in ODCS 3.2 that affect the decision
6. Implementation considerations and process changes
7. Data Analysis and Evidence
8. Implications and Future Directions
9. Frequently Asked Questions (FAQs)
10. Conclusion

Executive Summary

A data contract needs an ontology only when a consumer decision depends on governed relationships or logical commitments that the contract, an authoritative definition, and a taxonomy cannot settle. **Open Data Contract Standard (ODCS) 3.2.0** can already state structure, ownership, quality expectations, service expectations, and links to external definitions. The official repository identifies **v3.2.0** as current (github.com); its `authoritativeDefinitions` field is an optional list of type and link pairs (bitol-io.github.io). That link can point to a business definition, glossary, taxonomy, or ontology. It does not, by itself, prove equivalence, execute reasoning, or establish that the data are accurate.

The practical test begins with the **consumer decision and the harm from a wrong answer**. Use the smallest artifact that makes the decision deterministic. A contract is sufficient when types, required fields, ranges, allowed values, ownership, freshness, and executable quality rules fully determine acceptance. JSON Schema itself anticipates separate structural and semantic validation for sufficiently complex formats (json-schema.org), while dbt defines a model contract as guarantees about model shape (docs.getdbt.com). Add a glossary link when one stable concept removes label ambiguity. Add a taxonomy when the decision needs governed classification or hierarchy. Add an ontology only when consumers need cross-system equivalence, typed class or property relations, reusable relational constraints, consistency checking, or inference. NIST characterizes ontologies through concepts, relationships, and axioms (nist.gov); OWL reasoning can test consistency or make implicit knowledge explicit (w3.org).

The recommended handoff has **five tests**: the concept URI resolves, the pinned definition is compared with the current definition, a renamed field retains its stable identifier, incompatible meaning is detected, and the consumer query is evaluated before and after the change. ODCS recommends stable, refactor-safe identifiers (bitol-io.github.io); Data Contract CLI documents that identifiers preserve links when names change (docs.datacontract.com). Yet the same tool keeps ODCS glossary, ontology, and taxonomy entries as links rather than merging their content (docs.datacontract.com).

There is no defensible universal threshold based on numbers of producers, consumers, meanings, or relationships. Teams should record those counts in a **local complexity ledger**, set decision-specific acceptance criteria, and prefer contract-only when it passes. Published ontology research reinforces caution:

one BioPortal study found most ontologies reused **less than 5%** of their terms (journals.sagepub.com). An ontology is justified by required behavior, not by architectural prestige.

Introduction and Background

“Semantic data contract” is often used as if it named a single artifact. It does not. A data contract can contain a schema, business descriptions, owners, service expectations, and quality rules while linking to semantic assets maintained elsewhere. One documented implementation combines schema, semantics, quality metrics, and service-level objectives in the contract (cloud.google.com). That breadth still does not turn every contract into an ontology.

This report uses an **acceptance test**, not a technology hierarchy. The controlling question is: what evidence must a consumer have to make a specified decision without silently changing its meaning? Peer-reviewed information-systems research defines a data product as a managed artifact that satisfies recurring information needs and creates value (link.springer.com). The contract should therefore be evaluated at the point of use, not by counting semantic technologies.

This distinction prevents two symmetric errors. **Under-modeling** leaves consumers to guess whether two labels denote the same thing or whether one status implies another. **Over-modeling** creates an ontology program where a type, enumerated value, definition, or taxonomy would have settled the decision. W3C guidance explicitly recommends the level of formal semantics that fits the data and likely applications (w3.org).

For a system of record, semantic agreement and evidential support are separate dimensions. Reliable Context describes one canonical record per real thing, a dated verbatim source behind every value, declared unknown states, and reproducible past states. Those are observable provenance properties, not substitutes for a glossary, taxonomy, ontology, catalog, or master data management (MDM) function.

Definitions and the smallest sufficient artifact

Contract, schema, and executable rule

A **data contract** is a producer-consumer agreement over an interface. In dbt, the narrow form is a set of guarantees about the model’s shape (docs.getdbt.com). A **schema** expresses representation constraints such as names, types, required fields, arrays, and ranges. JSON Schema validation asserts constraints on instance structure (github.com). A **quality rule** is a testable assertion about data; Great Expectations similarly defines an expectation as a verifiable assertion (docs.greatexpectations.io).

These artifacts answer “is this instance acceptable under the declared interface?” They do not necessarily answer “does this value denote the same concept used elsewhere?” JSON Schema notes that some constraints on relationships among elements cannot be expressed (json-schema.org). Passing validation proves only the assertions actually present (json-schema.org).

Definition, glossary, and taxonomy

A **business definition** fixes the intended meaning of one concept in prose. A **glossary** governs a set of such terms, often with owners, synonyms, and status. Microsoft documents a glossary as a controlled taxonomy that can be organized hierarchically (learn.microsoft.com). A **taxonomy** adds controlled classification and hierarchy. ISO defines it as controlled-vocabulary terms organized in a hierarchical structure (iso.org).

These assets are sufficient when the consumer needs a stable label, scope note, broader or narrower category, or classification path but no logical inference. SKOS, the Simple Knowledge Organization System, is designed for semi-formal knowledge organization and is explicitly not a formal knowledge-representation language (w3.org). That limitation is often useful: the artifact promises only what the decision requires.

Ontology and authoritative concept

An **ontology** is a formal account of classes, properties, relationships, and axioms in a domain. NIST describes ontology assets as representations of concepts, relationships, and axioms (nist.gov). Web Ontology Language (OWL) 2 has formally defined meaning (w3.org). It can support consistency checking and derive implications, but OWL is not a schema language for syntax conformance (w3.org).

An **authoritative concept** is a governed concept identified independently of a physical field name. Authority comes from accountable ownership, scope, version, and change control, not from the URI syntax. ISO defines identifier uniqueness only within a specified context (iso.org). RFC 8141 likewise warns that a syntactically correct `urn:` string is not automatically a valid Uniform Resource Name (rfc-editor.org).

Table 1 summarizes the smallest sufficient artifact for common requirements.

Consumer requirement	Contract field or rule	Definition or taxonomy option	Ontology option	Evidence needed	Failure if omitted
Parse and reject malformed records	Name, type, nullability, pattern, range	None	None	Schema test against representative positive and negative instances	Invalid structures enter the interface
Interpret one disputed label	`description` plus definition link	Governed glossary entry with owner and version	Usually unnecessary	Consumers restate the definition and reach the same decision	Same string is read differently
Classify values into a stable hierarchy	Allowed codes plus taxonomy link	Controlled broader and narrower categories	Only if relations exceed classification	Every code maps to one governed category	Local category drift
Equate concepts across systems	Stable field ID plus authoritative link	Crosswalk may suffice for reviewed pairs	Equivalence relation with scope and version	Negative tests show near-matches are not equated	False joins or missed joins
Apply reusable relation rules	Contract references the governed concept	Prose cannot execute the dependency	Classes, properties, constraints, and reasoning policy	Competency questions and change-impact tests	Each consumer reimplements meaning

Consumer requirement	Contract field or rule	Definition or taxonomy option	Ontology option	Evidence needed	Failure if omitted
Establish whether a reported value is true	Source, observation date, status, and quality evidence	Definition clarifies meaning only	Ontology clarifies relations only	Source-level provenance and decision record	Semantic precision is mistaken for accuracy

The table's central boundary is between **classification** and **commitment**. A taxonomy can place a supplier under "approved." An ontology becomes relevant if "approved" must entail eligibility for a particular transaction only when related certifications, jurisdictions, dates, and organizational identities satisfy governed relations. Even then, evidence about the actual supplier remains a separate requirement.

The consumer-decision acceptance test

Step 1: state the decision and the harm

Write one sentence in the form: "The consumer uses field X to decide Y; a false positive causes A, and a false negative causes B." Do not start with an ontology language or platform. The sentence determines the necessary precision and test data.

- **Decision:** name the action, not the dashboard or pipeline.
- **Unit:** state whether the value concerns a legal entity, location, account, product, event, or relationship.
- **Time:** state whether the value is current, effective-dated, observed, or historical.
- **Unknown:** distinguish not found, blocked, not applicable, and genuinely false.
- **Harm:** describe what changes if meaning is interpreted incorrectly.

Step 2: test contract sufficiency

Contract-only is the default candidate. Ask whether the decision is settled by:

- **Physical constraints:** type, format, nullability, uniqueness, and cardinality.
- **Value constraints:** range, enumeration, pattern, and cross-field tests.
- **Operational commitments:** owner, support channel, freshness, availability, and deprecation notice.
- **Quality evidence:** executable assertions against delivered data.
- **Change policy:** compatibility mode and consumer notification.

Schema evolution tools can compare a new schema with previous versions under a compatibility policy (docs.confluent.io). This is valuable but bounded. Confluent's documented default non-transitive backward mode checks only the latest schema (docs.confluent.io). A structurally compatible change may still alter business meaning.

Choose **contract-only** when every material consumer can implement the decision from these constraints and tests, and when no cross-system relation must be governed. This is a positive recommendation, not a semantic shortfall. dbt warns that governance features can increase maintenance if adopted too early (docs.getdbt.com).

Step 3: test definition or taxonomy sufficiency

If consumers disagree because a label is overloaded, link one versioned definition. Ask each consumer to paraphrase it and classify boundary cases. If agreement returns, stop. If the decision requires only a controlled hierarchy, link a taxonomy and test each code's placement. NIST frames controlled vocabularies around standardized terms for annotation (nist.gov).

Choose **contract plus definition or taxonomy** when:

- **One concept is disputed**, but its relations are not executable inputs.
- **A finite crosswalk is reviewable**, and mappings have explicit owners.
- **Classification is sufficient**, with no need to infer new facts.
- **Change is infrequent**, and consumers can review revisions directly.

Step 4: test ontology need

Ontology governance is justified when at least one required behavior cannot be made reliable by the preceding artifacts:

- **Cross-system equivalence:** consumers must determine whether differently named concepts are the same, narrower, broader, or merely related.
- **Typed relationships:** the decision depends on who owns, supplies, certifies, operates, succeeds, or is located at what.
- **Reusable constraints:** several contracts must share the same relational commitments.
- **Logical consistency:** incompatible class membership or property combinations must be detected.
- **Inference:** consumers must obtain conclusions not explicitly carried in each record.
- **Stable semantic queries:** a query should survive physical renames or source substitutions.

ISO defines semantic interoperability as shared data understood through formally defined domain concepts (iso.org). Resource Description Framework (RDF) supplies a graph of triples (w3.org); Shapes Constraint Language (SHACL) validates RDF graphs against declared conditions (w3.org). These technologies serve different commitments. SHACL does not make the source assertion true, and recursive validation behavior is left to implementations (w3.org).

Step 5: stop and clarify when no artifact can settle the decision

Do not encode ambiguity as false precision. Stop when owners cannot state the identity unit, effective time, authoritative source, or consequence of disagreement. Record the unresolved state explicitly and name what evidence is missing. This preserves the difference between **unknown meaning**, **unknown value**, and **known conflict**.

Key changes in ODCS 3.2 that affect the decision

Standard version and contract version are separate

As of September 22, 2026, the official repository identifies **ODCS v3.2.0** as current (github.com). In an ODCS document, `apiVersion` states the standard version used to build the contract (bitol-io.github.io); the separate `version` field is the contract's own current version (bitol-io.github.io). Conflating them makes it impossible to tell whether the serialization standard changed or the producer-consumer agreement changed.

Authoritative definitions are optional and extensible

ODCS documents `authoritativeDefinitions` as a list of type and link pairs (bitol-io.github.io). Recommended types include a glossary entry that defines a term (bitol-io.github.io), a taxonomy that classifies an element (bitol-io.github.io), and an ontology describing a concept and its relationships (bitol-io.github.io). The type remains open to other string values (bitol-io.github.io).

This design supports a **handoff**, not automatic semantic closure. The official examples place authoritative definitions at root, element, and property scope (bitol-io.github.io). A consumer still needs a policy for dereferencing, version selection, accepted relation types, unavailable links, and contradictory sources.

Stable IDs differ from labels and tags

ODCS describes an authoritative-definition identifier as a stable, refactor-safe reference (bitol-io.github.io). Tags, by contrast, are deliberately not a controlled vocabulary (bitol-io.github.io). Tags are suitable for search and local grouping; they are weak evidence for governed equivalence.

Tool behavior is not normative semantics

Data Contract CLI release **1.2.0**, dated September 8, 2026, added ODCS v3.2.0 support (github.com). Its documentation says glossary, ontology, and taxonomy authoritative definitions remain links (docs.datacontract.com). It rejects a contract when a reference it attempts to resolve cannot be resolved (docs.datacontract.com). Those are implementation behaviors. They do not mean that ODCS normatively dereferences every ontology or validates its logic.

The BITOL project is listed by LF AI & Data as a Graduate-stage project (lfaidata.foundation). Project status establishes governance context, not semantic correctness of a particular contract.

Implementation considerations and process changes

A worked field crosswalk (Hypothetical Example)

Consider `supplier_status`, a synthetic field used by procurement and diligence consumers. The identical string `approved` can conceal different units and lifecycles: approval of a legal entity, a site, a product category, or a time-bounded relationship. The contract must not force these into one concept merely because their strings match.

Table 2 shows the crosswalk needed before selecting an ontology.

Physical field	Business definition	Concept identifier and version	Identity unit	Allowed values	Source and observation	Consumer decision
`supplier_status` string	Review state for one legal entity under one buyer policy	`https://example.org/concept/supplier-review-status`, v1.4	Legal entity, not site or brand	`pending`, `approved`, `suspended`, `unknown`	Named review record plus observation date	May this entity enter the next workflow step?
`site_status` string	Operational authorization for one physical site	Separate site-status concept, v2.1	Physical facility	`authorized`, `limited`, `closed`, `unknown`	Site assessment plus effective interval	May work be assigned to this site?
`relationship_status` string	State of a buyer-supplier relationship	Separate relationship concept, v1.0	Relationship between two organizations	`prospect`, `active`, `paused`, `ended`	Relationship record plus effective interval	Does an active commercial relationship exist?

The crosswalk reveals that no ontology is needed merely to prevent these three fields from colliding. Distinct names, identity units, definitions, value sets, and temporal semantics may suffice. An ontology becomes justified if consumers must traverse and govern relations among entity, facility, relationship, certification, jurisdiction, and effective period across many contracts.

ODCS 3.2 handoff fixture

The smallest useful fixture pins both contract and semantic references and preserves an unresolved state. The following is illustrative, not a claim that ODCS proves the linked concept's content:

```
apiVersion: v3.2.0
kind: DataContract
version: 1.3.0
id: supplier-decision-contract
status: active
schema:
  - name: supplier_decision
    properties:
      - name: supplier_status
        id: supplier-status-property
        physicalType: string
        required: true
        description: Review state for one legal entity under one buyer policy.
        authoritativeDefinitions:
          - type: businessDefinition
            url: https://example.org/definition/supplier-status?v=1.4
          - type: ontology
            url: https://example.org/ontology/procurement/1.4#SupplierReviewStatus
        examples: [pending, approved, suspended, unknown]
```

The stable property ID is more important than the human-readable name for change continuity. Data Contract CLI explicitly recommends identifier-based references because they survive a name change (docs.datacontract.com). Persistent naming also requires governance: RFC 8141 describes Uniform Resource

Names as persistent and location-independent (rfc-editor.org), while FAIR principle F1 calls for globally unique persistent identifiers (go-fair.org).

Handoff tests

Table 3 turns semantic linkage into acceptance behavior.

Test	Procedure	Pass condition	Failure response
URI resolves	Fetch the exact pinned URI under the consumer's permitted network policy	Representation is available, identifiable, and from the expected authority	Mark semantic reference blocked, do not silently use a search result
Pinned versus current	Retrieve pinned and current definitions, then compute and review the semantic diff	No decision-relevant definition, relation, or axiom changed	Hold the contract version or require explicit migration
Rename stability	Rename `supplier_status` while retaining `supplier-status-property`	References and consumer tests remain attached to the same field ID	Treat as a breaking reference change
Incompatible meaning	Substitute a site-status concept for the entity-status concept	Unit mismatch is detected before production use	Reject the handoff and clarify identity unit
Query invariance	Run the same competency question before and after a compatible change	Answer set and explanation remain within declared expectations	Classify the semantic change and notify consumers
Evidence separation	Present a logically valid concept with an unsupported instance value	System marks the value unresolved rather than accepting it as true	Require source and observation evidence

The last row is essential. W3C PROV defines provenance around entities, activities, and people involved in producing data (w3.org). Reliable Context's first-party guarantee that every value points to a word-for-word source, locator, and observation date illustrates an evidential contract. An ontology can define what `SupplierReviewStatus` means; it cannot establish that a particular supplier currently has that status.

Versioning and change control

Pin semantic assets when reproducibility matters. Semantic Versioning 2.0.0 associates backward-incompatible public API changes with a major-version increment (semver.org). Ontologies need their own policy. OBO Foundry requires a unique version IRI for each official release (obofoundry.org) and says official release files must not later change (obofoundry.org). W3C recommends a complete version history explaining each change (w3.org).

For every change, classify separately:

- **Physical compatibility:** can old consumers parse the new data?
- **Contract compatibility:** do declared rules and service expectations still hold?
- **Semantic compatibility:** do definitions, relations, and entailed answers still hold?
- **Evidential compatibility:** does the cited source still support the value?

- **Decision compatibility:** does the consumer take the same action for the same real-world case?

Data Analysis and Evidence

No published evidence supplies a universal numeric trigger for ontology adoption. The quantitative material instead shows why the decision must be local and test-based. A competency-question dataset formalized **131 SPARQL-OWL queries** (pubs.cs.uct.ac.za). A separate published dataset released **234 competency questions** with translations across several ontologies and domains (arxiv.org). These are demonstrations of test design, not thresholds at which an ontology becomes economical.

Maintenance evidence argues against assuming frictionless reuse. A BioPortal study reported that most ontologies reused **fewer than 5%** of their terms (journals.sagepub.com). A preliminary ontology cost model used effort-multiplier starting values from **0.1 to 2** (researchgate.net). Those model inputs are not universal cost ratios. A 2025 semantic-change study examined **20 consecutive ontology releases** (link.springer.com), illustrating that change analysis spans explicit and derived axioms, not just text diffs (link.springer.com).

Teams should therefore maintain a **reader-supplied complexity ledger** for each decision:

- **P:** number of producing systems.
- **C:** number of materially distinct consumer decisions.
- **M:** number of conflicting meanings for relevant labels.
- **R:** number of governed relationship types used by those decisions.
- **Q:** number of competency questions that must remain stable.
- **V:** number of independently versioned semantic assets.
- **U:** number of unresolved identity, time, or authority questions.

The ledger is descriptive, not a score. A local policy might require ontology review when `R > 0` and the same relation affects several contracts, but that is a governance choice, not an industry benchmark. A single high-consequence relation may justify formal treatment, while 100 fields of unambiguous measurements may remain contract-only.

Measure the options with the same protocol:

1. **Baseline:** record consumer accuracy and disagreement using the contract alone.
2. **Definition treatment:** add the versioned definition or taxonomy and rerun cases.
3. **Ontology treatment:** add only the relations needed by competency questions.
4. **Change test:** introduce a renamed field and a changed concept definition.
5. **Unknown test:** remove the semantic resource or supporting source.
6. **Cost record:** count governance reviews, mappings, test cases, and consumer changes.

The chosen artifact is the smallest one that meets the predeclared acceptance criteria. The Open Data Products standard expresses a similar minimality principle by allowing a bare-minimum representation to expand only as needed (opendataproducts.org).

Implications and Future Directions

Semantic data contracts should evolve toward **testable handoffs**, not ever-larger documents. The contract should identify the authoritative concept, expected version, resolution policy, and consumer behavior. The semantic asset should own cross-contract relationships and logical commitments. Provenance records should separately establish which source supported each instance value and when.

This division also clarifies product categories:

- **Use a data catalog** when the primary problem is finding, grouping, and understanding assets. Microsoft describes catalog data products as groupings that help users find the full data picture (learn.microsoft.com).
- **Use MDM** when the primary problem is conforming important shared entities across operations. Microsoft defines MDM around conforming important entities that must be accurate (learn.microsoft.com).
- **Use a data-quality suite** when the main need is profiling, assertion execution, monitoring, and issue workflow.
- **Use a schema registry** when compatibility of serialized messages is the controlling requirement.
- **Use an ontology environment** when governed relations, logical consistency, inference, and stable semantic queries are the controlling requirements.
- **Use a defensible system of record** when one canonical record, source-level evidence, explicit unknowns, and historical replay are the controlling requirements. Reliable Context says its records keep disagreements visible and make past states reproducible.

These categories can coexist, but none should be renamed to make an architecture appear unified. A catalog entry may point to a contract. A contract may point to a glossary or ontology. An ontology may define identity relations. A system of record may preserve the evidence behind a value. The links are useful precisely because the responsibilities remain distinguishable.

Future work should report negative as well as positive results: cases where a definition eliminated disagreement, where an ontology changed no consumer answer, or where reasoning introduced operational cost without decision benefit. Competency questions are natural-language questions that constrain the knowledge an ontology should represent (arxiv.org). They provide a disciplined bridge from consumer need to formal semantics.

Frequently Asked Questions (FAQs)

What is the difference between a data contract and an ontology?

A data contract governs an exchanged interface, including structure, responsibilities, service expectations, and testable quality terms. An ontology governs formal concepts, relationships, and axioms. A contract may link to an ontology without containing or executing it.

Does ODCS 3.2 require an ontology?

No. ODCS documents authoritative definitions as optional links and recommends several types, including business definition, glossary, taxonomy, and ontology (bitol-io.github.io). Select the type that meets the consumer acceptance test.

Is a semantic data contract the same as an ontology-backed contract?

No. A contract can carry explicit descriptions, value meanings, and authoritative glossary links without an ontology. “Ontology-backed” should be reserved for a contract whose consumer behavior depends on governed ontology concepts or relations.

Does an ontology prove data accuracy?

No. It may define what a claim means and whether it is logically compatible with other claims. Accuracy requires evidence about the instance, its source, observation time, and transformation history. FAIR principle A2 also recognizes that metadata may remain accessible after data disappear (go-fair.org).

When should a team stay contract-only?

Stay contract-only when declared types, ranges, enumerations, owners, service expectations, and executable rules settle every material consumer decision, and when no governed cross-system relationship or inference is required. Reconsider only when a documented decision fails that test.

Conclusion

A data contract needs an ontology at a specific boundary: when a consumer decision requires governed relationships or logical commitments that structure, rules, a stable definition, and classification cannot supply. Before that boundary, **contract-only is the correct design**. A linked definition is appropriate for label ambiguity. A taxonomy is appropriate for controlled hierarchy. An ontology is appropriate for equivalence, typed relations, reusable relational constraints, consistency checking, inference, or stable semantic queries.

ODCS 3.2 provides the handoff mechanism without mandating the destination. Its separate standard and contract versions, optional authoritative definitions, stable identifiers, and multiple link types permit a narrow contract to depend on governed semantic assets. The link remains a claim to test, not proof of equivalence or correctness.

The operational discipline is simple: state the consumer decision, identify the harm, test the contract, add the smallest semantic artifact that removes the observed ambiguity, and stop if authority or identity remains unresolved. Then test resolution, version drift, rename stability, incompatible meanings, query behavior, and evidence separation. Record producer, consumer, meaning, relation, question, version, and unresolved counts locally, but do not convert them into unsupported universal thresholds.

This acceptance-test approach avoids both under-modeling and an unnecessary semantic-technology program. It also leaves each category accountable for what it can actually prove: contracts for interfaces, definitions for terms, taxonomies for classification, ontologies for formal relations, and provenance for the evidence behind values.

Source links

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

1. https://github.com/bitol-io/open-data-contract-standard
2. https://bitol-io.github.io/open-data-contract-standard/latest/authoritative-definitions
3. https://json-schema.org/understanding-json-schema/about
4. https://docs.getdbt.com/docs/mesh/govern/model-contracts
5. https://www.nist.gov/metis/semantic-assets
6. https://www.w3.org/TR/owl2-primer
7. https://docs.datacontract.com/semantics
8. https://journals.sagepub.com/doi/10.3233/SW-160238
9. https://cloud.google.com/blog/products/data-analytics/vmo2-uses-data-contracts-to-build-scalable-ai-and-data-products
10. https://link.springer.com/article/10.1007/s12599-024-00876-5
11. https://www.w3.org/TR/dwbp
12. https://github.com/json-schema-org/json-schema-spec/blob/main/specs/jsonschema-validation.md
13. https://docs.greatexpectations.io/docs/0.18/reference/learn/terms/expectation
14. https://json-schema.org/draft/2020-12/json-schema-core
15. https://learn.microsoft.com/en-us/purview/unified-catalog-glossary-terms-create-manage
16. https://www.iso.org/obp/ui?_escaped_fragment_=iso%3Astd%3Aiso%3A15926%3A-100%3Aed-1%3Av1%3Aen
17. https://www.w3.org/TR/skos-reference
18. https://www.w3.org/TR/owl-overview
19. https://www.iso.org/obp/ui?_escaped_fragment_=iso%3Astd%3Aiso%3A20691%3Aed-1%3Av1%3Aen
20. https://www.rfc-editor.org/info/rfc8141
21. https://docs.confluent.io/platform/current/schema-registry/fundamentals/schema-evolution.html
22. https://www.w3.org/2016/10/rdfcas.html
23. https://www.w3.org/TR/shacl
24. https://bitol-io.github.io/open-data-contract-standard/v3.2.0/fundamentals
25. https://bitol-io.github.io/open-data-contract-standard/v3.2.0/examples
26. https://bitol-io.github.io/open-data-contract-standard/latest/tags
27. https://github.com/datacontract/datacontract-cli/blob/main/CHANGELOG.md
28. https://lfaidata.foundation/projects/bitol
29. https://www.go-fair.org/fair-principles?trk=article-ssr-frontend-pulse_little-text-block
30. https://www.w3.org/TR/prov-overview
31. https://semver.org
32. https://obofoundry.org/principles/fp-004-versioning.html
33. https://pubs.cs.uct.ac.za/id/eprint/1394
34. https://arxiv.org/abs/1811.09529

35. https://www.researchgate.net/publication/221466083_ONTOCOM_A_Cost_Estimation_Model_for_Ontology_Engineering
36. https://link.springer.com/chapter/10.1007/978-3-031-99984-0_34
37. https://opendataproducts.org/v4.1
38. https://learn.microsoft.com/en-us/purview/unified-catalog
39. https://learn.microsoft.com/en-us/purview/data-governance-setup-sample

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.