Reliable Context

RELIABLE CONTEXT / RESEARCH & PRACTICAL GUIDANCE

Testing Streaming Truth Discovery Under Drift

> 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 · streaming truth discovery · dynamic truth discovery · source reliability · concept drift · dynatd · streamtf · data provenance · truth discovery · data integration

Original article: https://reliablecontext.com/articles/streaming-truth-discovery-drift-test



In this article

1. Executive Summary
2. Introduction and Background
3. Key Changes
4. Implementation Considerations and Process Changes
5. Data Analysis and Evidence
6. Implications and Future Directions
7. Frequently Asked Questions (FAQs)
8. Conclusion

Executive Summary

Streaming truth discovery is not one problem. It is the intersection of **computational streaming**, **evolving truth**, and **evolving source reliability**. A one-pass algorithm answers the resource question, whether claims can be processed with bounded memory and update time. It does not establish that the latent truth is stationary, that provider quality is stable, or that late data have been interpreted correctly. Zhao, Cheng, and Ng's **StreamTF** and **1PassTF** were designed around one-pass processing, short response time, and limited memory (cse.hkust.edu.hk). The same paper describes 1PassTF as useful even for massive static databases, direct evidence that streaming computation and temporal drift are distinct.

Li et al.'s **DynaTD**, published in **2015**, addresses both changing values and locally changing source quality. The institutional record identifies it as a peer-reviewed conference contribution (researchconnect.suny.edu). It updates truths and source weights as data arrive (pmc.ncbi.nlm.nih.gov), visits incoming data once, and adds a smoothing factor **λ** for temporal continuity plus a decay factor **γ** for local reliability. Its authors evaluated weather forecasts for **88 cities**, stock reports for **1,000 stocks from 55 sources**, and flight data for **1,200 flights from 38 sources** (pmc.ncbi.nlm.nih.gov). Those are author-reported experiments, not broad replication.

The decision rule is empirical. Use a **static batch model** when neither truth nor source quality changes materially and latency permits retraining. Use a **one-pass model** when memory or response time is binding but stationarity remains defensible. Use **windowed retraining** when changes are real but labels or trusted reference values can identify a useful window. Use a **drift-aware model** when the target and provider error processes evolve independently enough that one forgetting rule cannot represent both. Abrupt drift changes concepts instantly, while gradual drift mixes old and new concepts during transition (link.springer.com). Tests therefore need separate injections for truth changes, source degradation, outages, missing values, delayed corrections, and correlated shocks.

The recommended scorecard reports **pre-change error, detection delay, peak error, recovery time, calibration, memory, and compute per update**. It also measures false smoothing across a known change point. Prequential evaluation tests each item before learning from it, but delayed verification requires later rescoring because labels can arrive with material latency (link.springer.com). Provenance remains a separate

requirement: the World Wide Web Consortium defines it as information about entities, activities, and people involved in producing data (w3.org). It explains what was observed and when. It does not make a drift model statistically valid.

Introduction and Background

Truth discovery integrates conflicting, noisy claims while estimating the reliability of their sources (arxiv.org). A conventional method alternates between inferred truths and source weights. The appealing feedback loop is also the central risk: a mistaken truth can depress the weight of a correct minority source, while copied or correlated errors can make a majority appear reliable. A published experimental review noted **limited systematic comparison** among competing methods (arxiv.org).

Sequential claims add several clocks. **Event time** is when the underlying state or claim applies. **Observation time** is when the system receives or processes it. Database **valid time** tracks real-world validity, whereas transaction time records when the database learned the value (docs.oracle.com). These clocks are needed to reconstruct evidence, but they are not equivalent to a model of changing truth.

The distinction matters because records can arrive out of event-time order (nightlies.apache.org). A late correction can look like a new regime if observation time is mistaken for event time. A source outage can look like changing reliability if absence is treated as an error. A genuinely changing target can look like provider failure if the evaluation reference is stale. The report therefore separates four questions:

- **Resource constraint:** Must each claim be processed once, with bounded memory and update time?
- **Truth dynamics:** Can the latent value change, and on what time scale?
- **Reliability dynamics:** Can a provider's error distribution degrade, recover, or change conditionally?
- **Evidence state:** Can each output be reconstructed from the claims and timestamps available at that point?

Reliable Context is relevant only to the fourth question and to defensible recordkeeping around the others. Its stated guarantee is **a cited, verbatim, dated source behind every value**. That observable property supports evaluation and review, but it is not evidence that any particular truth-discovery estimator handles drift.

Key Changes

Change 1: Separate the two latent processes

The first change is conceptual. Let **x(t)** denote the truth for an object at event time t, and let **r_s(t)** denote the reliability or error process of source s at that time. These are different latent quantities. Either, both, or neither may change.

Table 1 summarizes the resulting four-quadrant decision matrix.

Truth process	Source-quality process	Primary risk	Minimum credible approach	Decisive test
Static	Stable	Unnecessary model complexity	Batch truth discovery, or one-pass computation if resources require it	Compare held-out error and compute against majority, median, and a batch baseline
Dynamic	Stable	Lag or oversmoothing after a real change	Time-aware truth state, windowed retraining, or a state model	Inject abrupt and gradual target changes while holding source errors fixed
Static	Drifting	Mistaking provider degradation for a new truth	Local or time-varying source weights	Degrade and restore selected sources while keeping the target fixed
Dynamic	Drifting	Confounding target change with provider change	A model with separate temporal controls for truth and reliability	Cross both injections independently, then combine them

The matrix prevents a common category error. If x(t) is static and r_s(t) is stable, an online implementation may still be valuable, but it is solving throughput. If x(t) changes while r_s(t) remains stable, rapid forgetting may help truth estimates but needlessly destabilize source scores. If only r_s(t) changes, smoothing the truth can be reasonable while source weights require local adaptation.

Change 2: Treat streaming as a computational property

The **2014 CIKM** paper is the clearest control case. The institutional record dates publication to **November 3, 2014** (researchportal.hkust.edu.hk). Zhao, Cheng, and Ng describe the stream problem through **one-pass processing, short response time, and limited memory** (cse.hkust.edu.hk). Their probabilistic formulation combines data uncertainty with source quality, represented through true-positive, true-negative, false-positive, and false-negative rates.

The paper names two algorithms. **StreamTF** uses sequential Bayesian estimation and retains only one collection of votes in memory (cse.hkust.edu.hk). **1PassTF** is a stochastic one-pass optimizer, but it requires the input size T and is also described as useful for massive static databases (cse.hkust.edu.hk). In the authors' experiments, temporal buckets defaulted to **300 observations**, and performance included average accuracy, minimum accuracy, and variability across buckets.

This evidence changes the model-selection question. A single scan may be necessary when history cannot be retained. It says nothing by itself about whether older observations should lose statistical influence. The later **ASRA** framework adapts the frequency of source-weight recomputation rather than recomputing at each arrival (researchportalplus.anu.edu.au). That is another resource-accuracy tradeoff, not automatically a theory of regime change.

Change 3: Model evolving truth and local reliability explicitly

Li et al. motivate **DynaTD** with two observations from their applications: truth changes with temporal smoothness, and observed source reliability fluctuates around an underlying level (pmc.ncbi.nlm.nih.gov). These are empirical observations in the selected weather, stock, and flight data. They are not universal

properties.

The update carries forward accumulated information rather than revisiting old claims. The smoothing extension treats the previous aggregate as a pseudo-source. Larger **λ** gives history more influence, while smaller λ relaxes continuity (pmc.ncbi.nlm.nih.gov). The decay extension uses **γ** to balance persistent source weight against local environmental effects (pmc.ncbi.nlm.nih.gov).

Theoretical and empirical claims should be kept separate. Under the paper's equal-loss assumption, its theorem gives asymptotic behavior for the **√T-scaled** source-weight error and describes a fast convergence rate (engineering.purdue.edu). In experiments, the authors report that smoothing should be relaxed when information changes quickly (pmc.ncbi.nlm.nih.gov), and that the preferred decay factor becomes smaller as environmental variation grows (pmc.ncbi.nlm.nih.gov). Those sensitivity findings support testing λ and γ against known change rates. They do not provide automatic parameter selection.

Change 4: Use constraints as falsifiable assumptions

Constraint-Aware Dynamic Truth Discovery (CA-DTD) jointly estimates changing claims and source reliability without knowing either beforehand (experts.illinois.edu). Its central addition is a constraint-aware Hidden Markov Model that represents physical transition rules (experts.illinois.edu). It also fuses social-media observations with information from traditional news sources (experts.illinois.edu).

The institutional record reports evaluation on **two real-world social-media traces** (experts.illinois.edu). However, accessible primary text did not expose the trace identities, detailed metrics, or numeric improvements. The appropriate conclusion is narrow: constraints can add information where valid transition rules exist, but the rule source, exceptions, and external-news channel become part of the model. A constraint may reduce variance, or it may encode bias and suppress a legitimate regime change.

Change 5: Distinguish recent specialization from general validation

Continued research does not itself validate older assumptions. PLOS ONE published its recent example on **August 26, 2025** (journals.plos.org). The paper proposes a **Two-layer Group-based Truth Discovery (TGTD)** mechanism for fog-assisted mobile crowdsensing (journals.plos.org). One layer estimates truth within participant groups, and a second aggregates across groups (pubmed.ncbi.nlm.nih.gov). Participant readiness, commitment, and device ability inform initial credibility (journals.plos.org).

Its evaluation used **two real datasets and one synthetic dataset**, including weather reports from **18 sources for 30 cities** and stock reports for **1,000 symbols from 55 sources over 21 working days** (journals.plos.org). The authors report an accuracy-runtime tradeoff: TGTD is slightly slower than iCRH, while faster than centralized streaming CATD in their comparisons (journals.plos.org). The paper explicitly leaves detailed stream organization and other system questions outside scope (journals.plos.org).

Implementation Considerations and Process Changes

Choose by rejected assumption, not algorithm label

Table 2 compares model families by the assumption each relaxes. “Dynamic” in a method name is not enough. The required evidence is the process represented and the failure mode observed.

Model family	What it solves	Key assumption	When justified	Diagnostic failure
Static batch	Conflict resolution and global source weighting	Truth and source behavior are stable over the fitting set	Repeated batch results are stable and latency is acceptable	Error rises only because the batch spans multiple regimes
One-pass stream	Memory, throughput, and response constraints	Sufficient state can summarize earlier observations	Batch and one-pass answers agree within a declared tolerance	Bounded compute passes, but post-change error remains high
Windowed retrain	Local adaptation using recent data	A single window length represents relevant memory	Performance is robust across neighboring window widths	Short windows become noisy, long windows blur changes
DynaTD-style	Separate continuity of truth from locality of source quality	Smooth truth and decaying reliability approximate the domain	Independent λ and γ sweeps improve distinct injected failures	One parameter setting cannot handle both abrupt and gradual cases
Constraint-aware state model	Rule-governed truth transitions	Constraints are valid, complete enough, and exception-aware	Ablation shows gains survive constraint violations	Rules suppress a valid transition or import external-source bias
Group-based stream model	Hierarchical aggregation among participant groups	Group structure and credibility inputs are meaningful	Group ablations beat flat baselines under matched compute	Group assignments dominate results or fail under mobility
Reliable Context	Evidence-bearing system of record around a selected estimator	One canonical record per real thing, with cited values and replayable state	The organization must defend what was known, sourced, and changed	It does not replace a truth-discovery model or choose drift parameters

This comparison makes the selection falsifiable. A method is justified when it beats simpler baselines on the failure it was introduced to address, while staying within resource limits. It is not justified merely because its internal state changes over time. The **Reliable Context** row is a system-of-record option around the selected method, not a statistical estimator.

Preserve the timeline before estimating drift

A test record should separate three events that are easily collapsed:

- **True change:** x(t) changes at event time t0.
- **Source outage:** a provider stops reporting at t1 but does not emit false values.
- **Delayed correction:** a claim about t0 arrives at observation time t2.

The stream processor's watermark is only an estimate of when all data for a window should have arrived (beam.apache.org). A later element can therefore be classified as late after the watermark passes its window (beam.apache.org). In multi-input processing, event-time progress can be controlled by the furthest-behind input (nightlies.apache.org). These are transport and clock effects. They should not be charged to statistical drift.

Provenance gives the reconstruction layer. W3C PROV models an update as derivation of a new entity (w3.org), while system-versioned temporal tables retain history for point-in-time analysis (learn.microsoft.com). Reliable Context similarly states that a past state can be reproduced with its then-supporting sources. These properties let an evaluator replay what the estimator could have known. They do not choose λ, γ, or a detection threshold.

Run a staged failure-injection protocol

The following protocol isolates causes before combining them. Percentages and durations are design factors, not literature-derived performance claims.

- **Baseline:** Run stationary truth and stable source errors long enough to estimate normal variance.
- **Abrupt truth step:** Shift x(t) once by a known magnitude while leaving all source errors unchanged.
- **Gradual truth drift:** Move x(t) linearly across 10, 50, and 200 updates.
- **Recurring truth:** Restore a previously observed regime and test whether adaptation is faster the second time.
- **Single-source degradation:** Increase one provider's bias or variance at a known update.
- **Source recovery:** Restore that provider and measure whether its score recovers or remains permanently discounted.
- **Rank reversal:** Make the formerly strongest source weaker than a formerly mediocre source.
- **Source outage:** Remove claims without inserting false values.
- **Random missingness:** Delete observations independently at 5%, 20%, and 50% rates.
- **Informative missingness:** Make absence depend on the hidden truth or source condition.
- **Delayed observations:** Hold back claims for fixed and heavy-tailed delays.
- **Delayed correction:** Deliver a correction after the estimate has already been acted on.
- **Correlated shock:** Shift several source errors together without changing x(t).
- **Copied claims:** Duplicate a common wrong value across nominally distinct sources.
- **Exceptional event:** Introduce a valid discontinuity that violates the usual smoothness assumption.
- **Constraint violation:** Supply a true transition excluded by the state model's rule set.
- **Adversarial drift:** Target changes to the detector or weighting rule rather than sampling them randomly.
- **Combined regime:** Cross abrupt truth change, correlated source shock, missingness, and delay.

Missing values deserve explicit injection because they can materially affect drift detection (arxiv.org). Targeted distribution change is a special category termed **adversarial drift** (arxiv.org). Neither should be inferred from average error alone.

Control evaluation leakage

The process should preserve causal order and the knowledge actually available at each observation time:

- **Test, then update:** Score each claim before using its verified outcome for learning.
- **Version the reference:** Record when a ground-truth label was created, corrected, or invalidated.
- **Re-score delayed labels:** Maintain both the score known at decision time and the final retrospective score.
- **Freeze comparisons:** Give each method the same arrivals, corrections, missingness, and compute budget.
- **Report parameter search:** Separate parameters tuned on known change points from those selected online.
- **Retain negative results:** Record where a simpler median, majority, or batch baseline wins.

This discipline is necessary because immediate-label evaluation is unrealistic when verification arrives late (link.springer.com). It also avoids conflating a better retrospective reconstruction with a better real-time decision.

Data Analysis and Evidence

The evidence base is informative but narrow. DynaTD's real-data applications span weather, stock, and flight data. Its authors report MAE and RMSE reductions against selected baselines of **3% and 6.23%** on weather, **82.4% and 70.1%** on stock, and **7.4% and 11.4%** on flight (pmc.ncbi.nlm.nih.gov). The comparison is not fully symmetric because batch baselines do not directly consume streaming input.

The CIKM study reported **1PassTF** average accuracies of **0.9575**, **0.9426**, and **0.9071** for flight, weather, and stock data respectively (cse.hkust.edu.hk). These values establish feasibility under that protocol, not drift robustness. Its flight set covered **1,200 flights**, and its stock set covered **1,000 stocks from 55 sources** (cse.hkust.edu.hk). A broad review evaluated methods on both synthetic and real-world data (arxiv.org). Controlled synthetic tests remain important because they expose known change points and error processes.

Table 3 defines a minimum metric set. Each quantity is indexed by event time and reported separately for each injected failure, rather than collapsed into one global average.

Metric	Operational definition	Why it matters	Report with
Pre-change error	Error over the stable interval before injection	Establishes normal accuracy and variance	Mean, median, high quantile, and source coverage

Metric	Operational definition	Why it matters	Report with
Detection delay	Updates between a known change point and a declared alert	Separates fast adaptation from eventual accuracy	Missed detections and false alerts
Peak error	Maximum time-indexed error after injection	Captures transient harm hidden by averages	Timestamp and failure type
Recovery time	Updates until performance returns to a declared threshold	Measures adaptation and source-score rehabilitation	Threshold and sustained-window rule
False smoothing	Signed area between the estimate and new truth after a step	Detects suppression of real regime changes	λ, window width, or state prior
Calibration	Agreement between stated confidence and observed frequency	Tests whether confidence remains interpretable	Brier score or another proper score
Compute per update	CPU time and work per arriving claim	Tests stream feasibility	Hardware, batch size, and percentile
Memory	Peak and steady retained state	Tests bounded-memory claims	Number of objects, sources, and active windows

Detection metrics require care. **Mean Time to Detection** measures the delay to signal known drift, and **Mean Time between False Alarms** measures spacing between false signals (link.springer.com). The same review warns that such measures apply cleanly only to sudden drift, not every gradual transition (link.springer.com). Recovery time can be operationalized as the number of evaluation windows needed to cross a performance threshold after degradation begins (link.springer.com).

Calibration should accompany point error where methods output probabilities or confidence. The **Brier score** can be decomposed to distinguish calibration from refinement (academic.oup.com). Resource claims should be measured rather than inferred from labels such as online. A stream learner is expected to use limited memory and bounded update time, and published drift-detector research has historically emphasized inference quality more than computational performance (arxiv.org).

Implications and Future Directions

The immediate research need is **adaptive parameter choice**. DynaTD's sensitivity results explain which direction λ and γ should move under controlled change, but they do not solve online selection when the change type is unknown. A useful benchmark should therefore cross at least three truth-change rates with at least three reliability-decay settings, include a no-drift control, and publish the entire surface rather than the winning cell.

The second need is dependence-aware evaluation. Correlated provider shocks and copied claims violate the intuition that many agreeing sources constitute independent evidence. Constraint-aware models add structure, but their constraints and external feeds also create new dependence paths. Later axiomatic work cautions that theoretical results can apply narrowly because they depend on particular modeling assumptions (link.springer.com). Assumption ledgers and ablations are therefore as important as headline error.

The third need is replication across time semantics. Processing-time computations are not reproducible on replay because records receive different clock times (nightlies.apache.org). Event-time ordering, watermarks, source availability, and label revisions must be captured with the model state. NIST's terminology further reserves reproducibility for measurement under changed conditions (nist.gov). Replaying identical bytes is necessary for diagnosis, but cross-condition replication is the stronger test.

Finally, current evidence remains fragmented by application. The recent **TGTD** study demonstrates active work in group-based mobile crowdsensing, while CA-DTD shows how physical constraints can enter a dynamic model. Neither establishes that smooth truth, local reliability, participant groups, or transition rules transfer to a new feed. Future reports should publish negative results, exact change points, missingness patterns, calibration, and compute alongside average error.

Frequently Asked Questions (FAQs)

What is truth discovery in data integration?

It is a family of methods that resolves conflicting claims while estimating source reliability. The core definition is integration of noisy multi-source information through reliability estimation (arxiv.org). Unlike ordinary deduplication, the aim is to infer which claimed value is best supported and how much trust to place in each source.

What makes an online truth discovery algorithm different?

An online or streaming algorithm updates as observations arrive and may retain bounded state. That is a computational property. It becomes a drift-aware truth-discovery method only when its statistical model represents changing truth, changing source quality, or both.

How should source reliability estimation handle drift?

Estimate a long-run component and a local component separately where the data justify that structure. Test degradation and recovery independently from target changes. If one forgetting parameter drives both the truth state and source score, the model can confound a valid new value with a newly unreliable provider.

Is concept drift in streaming data the same as ground-truth drift?

No. Concept drift is broader and can include changes in feature distributions or predictive relationships. Ground-truth drift here means that the value being fused changes over event time. A recurring concept is one previously observed in the stream (link.springer.com), while an isolated outlier is shorter-lived than a drift event (link.springer.com).

How does data provenance relate to dynamic source credibility scoring?

Provenance records which entity, activity, or agent produced evidence and how versions relate. Dynamic credibility scoring is an inference about a source's error process. Provenance makes that inference reviewable and repeatable, but it does not prove the score is calibrated or causally correct.

What should streaming data quality monitoring report?

At minimum, it should report time-indexed error, detection delay, false alerts, peak error, recovery time, missingness, late-arrival rate, calibration, compute per update, and memory. Accuracy alone can be overly optimistic on imbalanced streams (link.springer.com).

How is ground-truth drift detection tested without smoothing away change?

Inject known abrupt and gradual changes, retain the unsmoothed reference, and report the signed post-change error trajectory. Compare several smoothing settings against a no-smoothing baseline. A detector window that is larger will generally resist noise but delay detection (link.springer.com). The evaluation should therefore show both false alarms and delay, not optimize one silently.

Conclusion

Testing streaming truth discovery under drift begins by refusing to collapse three questions. One-pass computation concerns resource use. Evolving truth concerns the latent value. Reliability drift concerns the provider's changing error process. A fourth layer, provenance and temporal recordkeeping, determines whether the evidence available at any past decision can be reconstructed.

The practical choice follows the four-quadrant matrix. Static and stable data favor a batch baseline unless throughput requires a stream implementation. Dynamic truth with stable sources calls for temporal truth modeling. Stable truth with drifting sources calls for local credibility updates. When both change, separate controls and independent injections are required. Windowed retraining is a credible middle option only when its memory horizon survives sensitivity testing.

DynaTD remains a useful published template because it distinguishes smoothing of truth from decay of source reliability and reports both theoretical and experimental results. Its assumptions and application set are limited, so the correct next step is not adoption by label. It is a staged comparison against simpler baselines using abrupt and gradual changes, source degradation and recovery, missingness, delayed corrections, correlated shocks, and exceptions to smoothness.

The decisive evidence is a time-indexed scorecard: error before change, detection delay, peak error, recovery time, calibration, memory, update cost, and false smoothing. If a sophisticated method cannot improve the failure it was designed for without obscuring a real regime change, the simpler model is the more defensible choice.

Source links

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

1. https://cse.hkust.edu.hk/faculty/wilfred/paper/cikm14a.pdf
2. https://researchconnect.suny.edu/en/publications/on-the-discovery-of-evolving-truth
3. https://pmc.ncbi.nlm.nih.gov/articles/PMC4688022
4. https://link.springer.com/article/10.1007/s41060-024-00620-y
5. https://link.springer.com/article/10.1007/s10618-019-00654-y

6. https://www.w3.org/TR/prov-dm
7. https://arxiv.org/abs/1505.02463
8. https://arxiv.org/abs/1409.6428
9. https://docs.oracle.com/en/database/oracle/oracle-database/26/vldbg/time-based-info.html
10. https://nightlies.apache.org/flink/flink-docs-stable/docs/dev/datastream/execution_mode
11. https://researchportal.hkust.edu.hk/en/publications/truth-discovery-in-data-streams-a-single-pass-probabilistic-appro
12. https://researchportalplus.anu.edu.au/en/publications/an-effective-and-efficient-truth-discovery-framework-over-data-st
13. https://engineering.purdue.edu/~lusu/papers/KDD2015Yaliang.pdf
14. https://experts.illinois.edu/en/publications/constraint-aware-dynamic-truth-discovery-in-big-data-social-media
15. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0330656
16. https://pubmed.ncbi.nlm.nih.gov/40857294
17. https://beam.apache.org/documentation/basics
18. https://nightlies.apache.org/flink/flink-docs-stable/docs/ops/debugging/debugging_event_time
19. https://learn.microsoft.com/en-us/sql/relational-databases/tables/temporal/overview?view=sql-server-ver17
20. https://arxiv.org/abs/2008.03662
21. https://arxiv.org/abs/1803.09160
22. https://link.springer.com/article/10.1007/s10994-025-06945-4
23. https://academic.oup.com/jrsssb/article/69/2/243/7109375
24. https://arxiv.org/abs/2304.08319
25. https://link.springer.com/article/10.1007/s10458-022-09569-3
26. https://www.nist.gov/pml/nist-technical-note-1297/nist-tn-1297-appendix-d1-terminology
27. https://link.springer.com/article/10.1007/s10115-017-1022-8

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.