Position paper

A kernel of truth: records that carry their evidence in organizations that fuse private and public information

Author

Reliable Context Research

Date

21 September 2026

Version

0.1

Status

Draft for comment

Download PDF

Research program

Abstract

Organizations that fuse private and public information produce files somebody acts on: a credit decision, a procurement award, an exclusion, a diligence opinion, a regulatory filing. The information in those files arrives from two halves that behave differently. The public half is contestable by anyone and moves underneath you; the private half is stable but cannot be shown to the person contesting the result. Most systems built for this task optimise the moment of reading, and treat the record that remains afterwards as a by-product. This paper argues the opposite order. The record is the product, and four properties determine whether it can be defended once the people who built it have moved on: exactly one canonical record per real thing, a verbatim and dated source behind every value, an unknown stated in words rather than left blank, and any past state reproducible on demand. Each property is argued from a failure it prevents rather than asserted as a principle, and each is shown to cost something a system optimised for reading would prefer not to pay. The paper

closes on what such a record owes a reader who was not present when the value was written and cannot re-derive it.

Contents

1 **The fusion problem: where provenance dies**
Two halves with incompatible availability, joined by a summary that carries nothing.

2 **A name is not an identity**
A question of degree answered where a question of fact was asked.

3 **Absence is not evidence: the case for declared negatives**
Three different states rendered as one blank, and why the blank is read as a no.

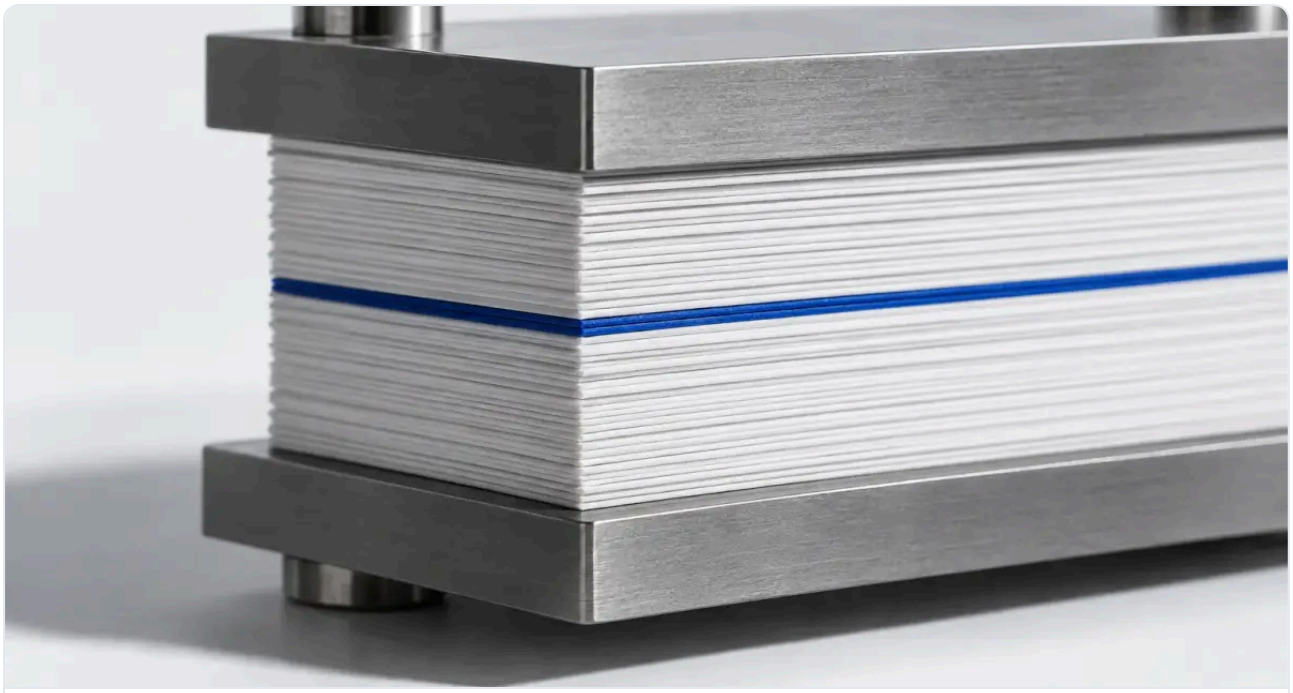
4 **Time: a record that cannot be replayed is a claim**
Three dates that are routinely conflated, and what overwriting destroys.

5 **Retrieval-centred context and what it cannot promise**
Citation produced alongside an answer is not citation the answer was read out of.

6 **What a system of record owes its reader**
Four properties, argued as refusals, for a reader who cannot re-derive anything.

Q **Open questions this paper does not settle**
Five, stated plainly, and where each one is being worked on.

R **References**
Twenty-five works, each linked to its published record.



A file is a stack under pressure, and the sheet that matters is never the one on top. The argument of this paper is that the sheet somewhere in the middle, the one a value was actually read from, must still be findable years later by somebody who did not put it there.

Fusion is usually described as a merge: take a public record, take an internal record, produce one. The description hides where the difficulty is. The two halves do not fail in the same way. The public half is contestable by anyone and moves underneath you, because a registry is corrected, a filing is superseded and a page is edited without a diff; it is honestly citable only as it stood on a stated date. The private half is stable and often authoritative, and cannot be shown to the person disputing the result, because of counsel, because of a licence, or because it names somebody. A fused value therefore rests on two supports with incompatible availability, and that asymmetry, not the join, is the engineering problem.

The distinction worth making is between merging records and merging warrants. Merging records is a data problem with a mature toolkit. Merging warrants is the question of what a reader is entitled to conclude from the combination, and it has no automatic answer: the combined value is as contestable as its weakest public support and as unshowable as its most restricted private one. A system that treats the join as the whole job emits a value whose warrant nobody can state, and the first person to ask for it is usually the last person you want asking.

The tools exist. Database provenance gives precise accounts of why a tuple appears in a result and where each of its values came from [1], the semiring formulation shows that such accounts compose through the query algebra rather than being bolted on afterwards [2], and PROV gives a vocabulary for moving them between systems that were not built together [3, 4, 5]. None of it survives the step most pipelines take next, which is to summarise. A summary is a new artefact with no provenance of its own unless one is constructed for it deliberately, and once the summary is what the reader sees, the derivation underneath it is decorative. Provenance does not die at the join. It dies at the point where the system starts producing prose.

Two objections are fair. The first is cost: attaching a sentence, a locator and a retained copy to every value is more expensive than writing the value. It is, and the cost is paid once, at leisure, against a cost that otherwise falls at the worst possible moment. The second is harder. Some values are genuinely not attributable to a single sentence: a ratio, a total, a classification drawn across twelve documents has no quotable origin. The wrong response is to pick one input and cite it, which is this paper's subject in its most respectable disguise. The right one is to record a derivation as a derivation, naming its inputs and their evidence, so a reader can descend one level and find sentences at the bottom. Records management has defined a record by its capacity to serve as evidence of an activity for decades [6]; a summary with no derivation does not meet that definition, and no amount of fluency makes it one.

Definition

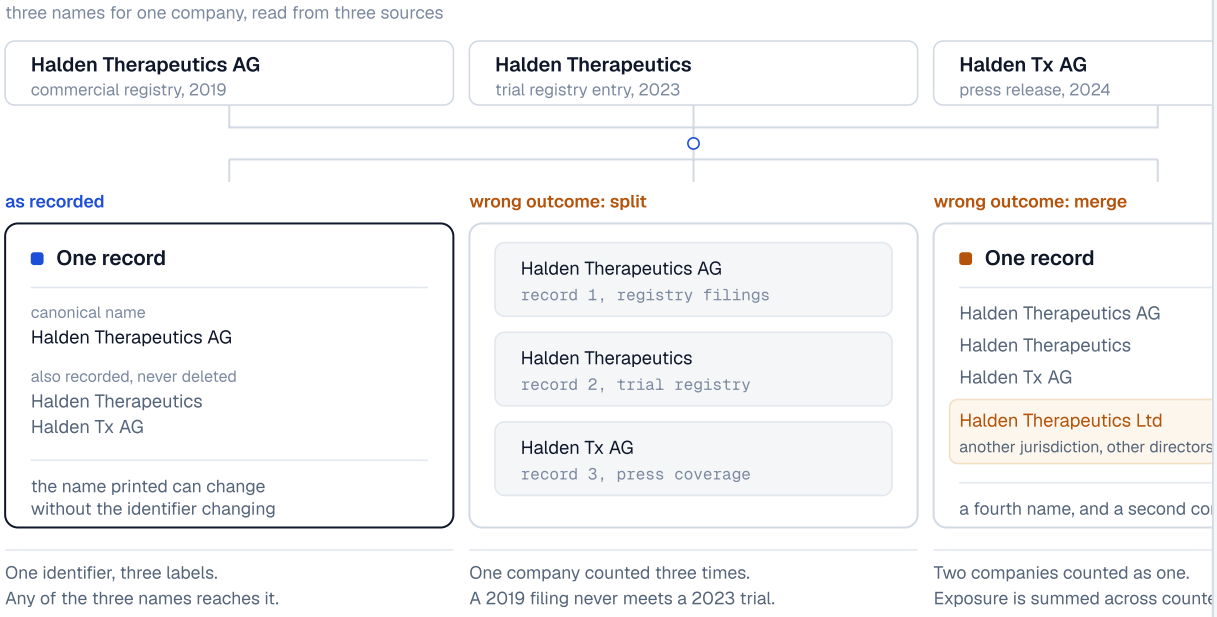
Provenance, in this paper, is not metadata about a file. It is the pairing of a value with the specific sentence that asserts it, the locator of that sentence, the date the sentence was read, and a retained copy of the document as it stood on that date. A source name without a sentence is not provenance; it is a reference to where provenance might be found.

Linkage was mechanised on vital records in the 1950s [7] and formalised by Fellegi and Sunter as a decision procedure over agreement patterns with explicit error rates [8]. Most of the work since has made the comparison cheaper and the classifier better [9, 10, 22]. All of it answers a question of degree: how alike are two records. The operational question is not a question of degree. It is whether these are one thing. A similarity threshold answers the first and is then used as though it had answered the second, which is why tuning a threshold feels like making policy. It is making policy, at a scale of hundredths, with no record that a policy was made.

A name is the most available comparison and among the least reliable, because names are not identifiers. They are labels administered by other people for their own purposes, and they are correct only relative to a document and a date. One company carries three names across a decade, an incorporation name, a trading name and a post-acquisition name, and every one of them is right on some filing. Two companies in different jurisdictions carry one name, one director and one registered agent, and are two companies.

The two failures that follow are not symmetric. A bad merge sums an exposure across two counterparties and produces a number that is wrong in a direction nobody checks, because the file looks more complete than it did before. A bad split hides a real connection, and is the error that survives an audit, because nothing about a split looks anomalous: two thin records are exactly what a young relationship is supposed to look like. The merge is louder when it is found. The split is almost never found.

Frequently the correct output is neither, but two records that remain two with the relation between them stated, dated and readable. That is unsatisfying to anyone who wants a clean count of entities, and it is the concession this section makes: a system that refuses to merge on insufficient evidence will show a reader two rows where the world has one thing, and will look wrong in precisely the way a careful system looks wrong. The defence is asymmetry again. Two rows can be joined later, with the evidence that justified the join attached to the act. A bad merge destroys the separation, and the evidence that would have undone it is now filed under the surviving identifier where nobody will think to look. What must not happen, in either direction, is that the judgement stays implicit in a coefficient. Whether two records are one thing is a claim about the world. It is either stated, evidenced and dated, or it is a side effect of a parameter somebody set on a Tuesday.



Halden Therapeutics AG is a fictional company used for demonstration. Three names, read from three sources, admit three outcomes: one record that retains all three labels, a split that counts one company three times, and a merge that absorbs a second company carrying a similar name in another jurisdiction. Only the first can be undone by later evidence.

A blank cell is read as a negative by almost every reader, and it is not one. Three different states of the world are routinely rendered as the same emptiness: nothing was found after looking, something exists but this reader is not licensed to open it, and the property does not apply to this entity at all. The three are not variations on a theme. The first is a statement about a search, the second about a permission, the third about the kind of thing being described. Collapsing them destroys information the reader often needs more than the value itself, because each implies a different next action: widen the search, request clearance, stop asking.

Declaring the negative in words is cheap. Making the declaration worth anything is not, because a negative is a claim and a claim needs evidence of its own: which sources were searched, under what query, on what date, and what was found instead. Without that, a declared unknown is a blank with better manners. With it, the reader can disagree with the search rather than with the silence, which is the only form of disagreement that leads anywhere. It also changes what the value is: a negative supported by a dated search is a finding, and findings can be re-run.

Establishing that a declaration was correct is genuinely hard, and the honest position is that nobody has solved it. The ground truth is also an absence, so it cannot be annotated the way a positive can. Fact-verification benchmarks carry a not-enough-information class over a fixed corpus [16], which measures something real but not this: a claim unsupported within one corpus is a different statement from a property that does not exist in the world. Treating the two as interchangeable is how a system trained to be careful learns to be confidently empty. Databases have long made the underlying choice explicit: the closed-world convention treats whatever is not stored as false [25], and a declared negative is the refusal to adopt that convention silently.

The objection from the other side is commercial rather than technical, and it is the serious one. A file full of declared unknowns looks worse than the same file with blanks. It is less saleable, it invites questions, and it makes the reader do arithmetic about coverage that the blank version quietly did for them. The answer is that the blank version was never more complete; it was less honest at the same completeness. There is also a consequence the blank cannot have: a declared negative acquires an expiry the day it is written, because a search that found nothing in March says nothing about June. A blank makes no claim, so it never goes stale, and that is exactly why it survives in files long after it stopped being true.

Three states, one blank cell, three different next actions

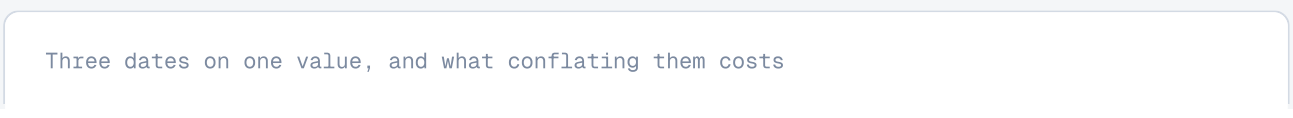
Declared as	What it asserts	What it must carry	What the reader does next
Not found	A search was run and returned nothing supporting a value.	The sources searched, the date, and what was found instead.	Widen the search, or accept the negative and date it.
Blocked	A source exists and this reader may not be shown it.	That something is withheld, and under which policy, never the content.	Request clearance, or escalate to a reader who has it.
Not applicable	The property does not exist for an entity of this kind.	The reason it does not apply, stated once and reusable.	Stop asking, and stop counting it as missing coverage.

A record without time is a photograph presented as a law. Every value in a fused record carries at least three dates that matter and are routinely conflated: the date the source itself states, the date the source was read, and the date the value entered the record. Retrieval date standing in for stated date is the most common of these errors and the hardest to find afterwards, because the value is usually still right and nothing fails. It becomes visible only when the value stops being right, which is the one moment at which nobody has time to audit how it was derived.

Replay makes the distinction operational. If the record can be reconstructed as it stood on a past day, together with the sources that supported it on that day, a decision can be defended against the record its maker actually had, rather than against a record that has moved since. That is not a convenience. A decision judged against today's evidence is judged against evidence its maker did not have, which is unfair when the decision was good and unfalsifiable when it was bad.

The temporal-database literature already separates the time a fact holds in the world from the time the database learned of it [23], and the SQL standard has carried both since 2011, as application-time periods and system-versioned tables [24]. Versioning a row is not replay, and the difference is where most implementations quietly fail. A store that keeps every past value, but refreshes its cached sources in place, can reconstruct what the record said and not what it knew. The excerpt still matches the copy on disk because the copy on disk was replaced. Replay requires that the evidence be immutable too, that a correction be an additional entry rather than a replacement, and that the superseded value stay readable with the reason it was superseded. Provenance vocabularies model derivation as timed activity and give a place to put all of this [3]; keeping the times honest is a discipline, not a data type.

The objection is that this is expensive and that nobody asks for last March. Both halves are half true. Append-only storage of values and their sources costs real money, and the ergonomics are worse: an overwrite is one statement, a supersession is three, and every reader now has to be told which state they are looking at. But the people who ask for last March are precisely the people whose questions cannot be declined, being a regulator, opposing counsel, an acquirer's diligence team or an internal investigation into a decision that went wrong. They ask rarely and they ask late, long after the analyst has left and the pipeline has been rewritten twice. A system that cannot answer them is not holding a record. It is holding a claim about what it currently believes, which is a different and much weaker object.



Date	What it answers	Failure when it is missing
Stated	When the source says the fact held.	A figure from an old filing is read as current because the page was fetched today.
Observed	When the source was read, and what it said then.	A revised page cannot be distinguished from a misreading of the original.
Entered	When the value became part of the record.	An export cannot be tied to the state of the record at the moment it was produced.

Placing retrieved passages beside a question improved the answers, and the improvement is not in dispute[15]. Generated text that its source does not support is a documented failure mode of language models across tasks[21]. The arrangement is the right default for a great many tasks, and nothing below is an argument for doing without retrieval. It is an argument about what the arrangement can be asked to promise, and the answer is: less than it is routinely asked to promise by people who then act on the result.

The first limit is attribution. There is a workable definition, namely whether a generated statement is supported by the source attached to it [17], and when generative search systems were measured against that definition, a substantial share of their sentences were found not to be fully supported by their own citations [18]. That is not a defect of one product. It is what happens when a citation is produced alongside an answer rather than being the thing the answer was read out of. The ordering is the whole of it: a citation chosen to accompany a sentence is a plausible source, and a sentence extracted from a source is an attributed one. From the outside the two look identical, which is why the difference has to be enforced where the value is written and cannot be recovered by inspecting the output.

The second limit is time. Retrieval returns what the index holds now, which is not what the analyst read in March, and an index has no obligation to remember that it once held something else. Nor does it have any way to tell you what it did not hold: an empty result set is indistinguishable from an unlucky query, and both are indistinguishable from a document that was removed. A retrieval-centred system can answer a question well today. On its own it cannot tell you what it answered last quarter, what it was reading when it did, or whether that source still says the same thing.

The fair rejoinder is that none of these limits are inherent, and that a retrieval stack can be made to retain what it read, timestamp it and keep the excerpt. That is exactly right, and it is

the argument of this paper stated from the other end: the part that has to change is not the model or the index but what happens after the answer, where the retained copy, the excerpt and the dates either become part of a record or are discarded with the context window. A system that keeps them has stopped being a retrieval pipeline with citations and has started being a system of record, whatever it is called internally.

What can be promised at read time, and what has to be decided at write time

Promise	Retrieval at read time	What keeping the promise requires
Relevant	Yes. This is what ranking is for.	Nothing further.
Attributed	Partly. A named source is not the same as the sentence the value came out of.	The excerpt is recorded as the value is written, not selected afterwards.
Dated	No. The index knows when it crawled, not what the source said.	Stated, observed and entered dates kept apart on the value itself.
Repeatable	No. The same question assembles a different context next quarter.	A retained copy of the source and an append-only history of the value.
Complete	No. An empty result set is not a statement about the world.	A declared negative carrying what was searched and when.

The reader who matters was not present when the value was written, has no way to derive it again, and is often not friendly. They arrive years later with a specific question about one value in a file somebody acted on, and their first move is to ask where it came from. Everything in this paper is an attempt to make that question answerable by the record itself rather than by whoever is still at the company.

Four things are owed to that reader, and each is a refusal as much as a promise. One canonical record per real thing refuses the convenience of letting two spellings become two companies, and the greater convenience of letting two companies become one. A verbatim, dated source behind every value refuses a source name standing in for a sentence, which is the form most citation actually takes. A declared unknown refuses the blank cell, at the price of admitting on the surface how much is not known. Replay refuses the overwrite, and with it

most of the ergonomics of a mutable store. Each refusal is unpleasant in the moment it is enforced, which is why systems that do not enforce them mechanically do not enforce them at all.

It is worth being precise about what this does not buy. Truth-discovery and data-fusion methods estimate a value and the reliability of its sources jointly, and estimate them well[11, 12, 13, 14, 19, 20]. Nothing here improves on them as estimation, and a system built on the properties above may well be less accurate than one that weights sources statistically. The distinction is between accuracy and defensibility. A reliability score cannot be read back to the sentence that produced it and cannot be argued with by the person it was used against, so it is the wrong instrument at the moment a value is challenged, however right it was on average across the corpus.

The promise, then, is smaller than truth, and stating it honestly matters more than stating it grandly. A record that carries its evidence is not thereby correct: its sources can be wrong, its readings can be shallow, and its declared negatives can be lazy. What it is, is accountable. Every value on it resolves to a sentence, a locator and a date, and therefore to something that can be confronted, re-read and overturned by a person who was not there. That is the whole of what a system of record can promise, and organizations that act on fused information have been making decisions for years on files that cannot promise even that.

Four properties, the refusal each one implies, and the price it charges

The record owes	It therefore refuses	Price paid
One record per thing	A merge on resemblance, and a split that hides a known relation.	Two rows on the screen where the world may have one thing, until the evidence settles it.
A cited, dated source	A source name standing in for a sentence.	Values that cannot be written until something has actually been read.
A declared unknown	The blank cell, and the confidence it borrows.	A file that shows its own coverage, including where the coverage is thin.

The record owes	It therefore refuses	Price paid
A reproducible past	The overwrite, and the in-place refresh of a source.	Storage, and a reader who must always be told which state they are looking at.

Open questions

What this paper does not settle.

Five things the argument above assumes, asserts or postpones. Each is stated at greater length on the research page, together with what a result would have to look like for us to consider it moved.

Open problems in full

01

What evidence is sufficient for sameness

The paper argues that a threshold cannot answer it and does not say what does, across kinds of entity, in a form a second party could check.

02

How a declared negative is scored

We can state what was searched. We cannot yet state, in a way anybody could grade, that the search should have found something if it existed.

03

When a retained source stops supporting its value

A revision that moves a number invalidates a reading; a revision that moves a heading does not. Nothing here reduces that line to a check that runs.

04

How sources are compared without a ladder

Section 6 rejects a standing hierarchy of authority and offers no general replacement for it beyond deciding each case on cited evidence.

05

How any of this is evaluated end to end

No public benchmark holds the private half of a fused record, and the organizations that hold it cannot release it. The paper asserts properties it cannot yet measure at scale.

References

Twenty-five works, each linked to its published record.

Each entry gives authors, title, venue and year, and links to the most stable address we could confirm: the publisher record, the standards body or an archive copy. The research page groups a shorter version of this list by problem, and says what we took from each entry.

Reading list

[1] Why and Where: A Characterization of Data Provenance
Peter Buneman, Sanjeev Khanna and Wang-Chiew Tan

International Conference on Database Theory (ICDT), 2001. https://doi.org/10.1007/3-540-44503-X_20

[2] Provenance Semirings

Todd J. Green, Grigoris Karvounarakis and Val Tannen

ACM Symposium on Principles of Database Systems (PODS), 2007. https://doi.org/10.1145/1265530.1265535

[3] PROV-DM: The PROV Data Model

Luc Moreau and Paolo Missier, editors

W3C Recommendation, 30 April 2013. https://www.w3.org/TR/prov-dm/

[4] PROV-O: The PROV Ontology

Timothy Lebo, Satya Sahoo and Deborah McGuinness, editors

W3C Recommendation, 30 April 2013. https://www.w3.org/TR/prov-o/

[5] Provenance: An Introduction to PROV

Luc Moreau and Paul Groth

Synthesis Lectures on the Semantic Web, Morgan & Claypool, 2013. https://doi.org/10.2200/S00528ED1V01Y201308WBE007

[6] ISO 15489-1: Information and documentation, records management, part 1, concepts and principles

International Organization for Standardization

ISO, 2016. https://www.iso.org/standard/62542.html

[7] Automatic Linkage of Vital Records

Howard B. Newcombe, James M. Kennedy, S. J. Axford and Allison P. James

Science, 1959. https://doi.org/10.1126/science.130.3381.954

[8] A Theory for Record Linkage

Ivan P. Fellegi and Alan B. Sunter

Journal of the American Statistical Association 64(328), 1969, pages 1183-1210. https://doi.org/10.1080/01621459.1969.10501049

[9] Entity Resolution: Theory, Practice & Open Challenges

Lise Getoor and Ashwin Machanavajjhala

Proceedings of the VLDB Endowment 5(12), 2012. https://www.vldb.org/pvldb/vol5/p2018_lisegetoor_vldb2012.pdf

[10] Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection

Peter Christen

Springer, 2012. https://doi.org/10.1007/978-3-642-31164-2

[11] Truth Discovery with Multiple Conflicting Information Providers on the Web

Xiaoxin Yin, Jiawei Han and Philip S. Yu

IEEE Transactions on Knowledge and Data Engineering 20(6), 2008. https://doi.org/10.1109/TKDE.2007.190745

[12] Data Fusion

Jens Bleiholder and Felix Naumann

ACM Computing Surveys 41(1), 2008. https://doi.org/10.1145/1456650.1456651

[13] Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion

Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun and Wei Zhang

KDD, 2014. https://doi.org/10.1145/2623330.2623623

[14] A Survey on Truth Discovery

Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan and Jiawei Han

ACM SIGKDD Explorations Newsletter 17(2), 2016. https://doi.org/10.1145/2897350.2897352

[15] Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel, Sebastian Riedel and Douwe Kiela

Advances in Neural Information Processing Systems (NeurIPS), 2020. https://arxiv.org/abs/2005.11401

[16] FEVER: a Large-scale Dataset for Fact Extraction and VERification

James Thorne, Andreas Vlachos, Christos Christodoulopoulos and Arpit Mittal

NAACL-HLT, 2018. https://aclanthology.org/N18-1074/

[17] Measuring Attribution in Natural Language Generation Models

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc and David Reitter

arXiv, 2021; Computational Linguistics 49(4), 2023. https://arxiv.org/abs/2112.12870

[18] Evaluating Verifiability in Generative Search Engines

Nelson F. Liu, Tianyi Zhang and Percy Liang

Findings of EMNLP, 2023. https://aclanthology.org/2023.findings-emnlp.467/

[19] Knowledge-Based Trust: Estimating the Trustworthiness of Web Sources

Xin Luna Dong, Evgeniy Gabrilovich, Kevin Murphy, Van Dang, Wilko Horn, Camillo Lugaresi, Shaohua Sun and Wei Zhang

Proceedings of the VLDB Endowment 8(9), 2015. https://www.vldb.org/pvldb/vol8/p938-dong.pdf

[20] Integrating Conflicting Data: The Role of Source Dependence

Xin Luna Dong, Laure Berti-Equille and Divesh Srivastava

Proceedings of the VLDB Endowment 2(1), 2009. https://doi.org/10.14778/1687627.1687690

[21] Survey of Hallucination in Natural Language Generation

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto and Pascale Fung

ACM Computing Surveys 55(12), 2023 (preprint arXiv:2202.03629). https://arxiv.org/abs/2202.03629

[22] A Survey of Blocking and Filtering Techniques for Entity Resolution

George Papadakis, Dimitrios Skoutas, Emmanouil Thanos and Themis Palpanas

ACM Computing Surveys 53(2), 2020 (preprint arXiv:1905.06167). https://arxiv.org/abs/1905.06167

[23] Developing Time-Oriented Database Applications in SQL

Richard T. Snodgrass

Morgan Kaufmann, 2000 (author-hosted copy). https://www2.cs.arizona.edu/~rts/tdbbook.pdf

[24] Temporal Features in SQL:2011

Krishna Kulkarni and Jan-Eike Michels

ACM SIGMOD Record 41(3), 2012. https://doi.org/10.1145/2380776.2380786

[25] On Closed World Data Bases

Raymond Reiter

In Logic and Data Bases, Plenum Press, 1978, pages 55-76. https://doi.org/10.1007/978-1-4684-3384-5_3

Further reading

Articles on this page's subject.

The research library

Provenance, time and declared unknowns

Minimum Data Provenance Requirements for Defensible Values

Fifteen minimum provenance fields and five audit questions, mapped to PROV-O, with temporal rules, declared unknowns and validation tests.

Identity and canonical records

Entity Resolution Evaluation Beyond Pairwise F1

Pairwise against cluster metrics, sampling design and uncertainty for judging entity resolution, worked through an eight-record example.

Provenance, time and declared unknowns

Open World Assumption: Why Unknown Is Not False

Open-world negation, explicit negative knowledge and five epistemic states, with the evaluation protocols that keep an unknown from being scored as false.

Provenance, time and declared unknowns

Bitemporal Data vs Temporal Knowledge Graphs: A Guide

Bitemporal records against temporal knowledge-graph research: snapshot replay versus link prediction, their benchmarks, and when to combine them.

Context engineering and LLM evaluation

RAG Fault Isolation: Is the Answer Defensible?

Locating where a retrieval-augmented answer failed across six observable boundaries, with claim-level metrics and controlled evidence tests.

How to cite

Title	A kernel of truth: records that carry their evidence in organizations that fuse private and public information
Author	Reliable Context Research (corporate author)
Version	0.1, draft for comment
Date	21 September 2026
Permanent URL	https://reliablecontext.com/paper
PDF	https://reliablecontext.com/paper.pdf

```
@techreport{reliablecontext2026kernel,
  author       = {{Reliable Context Research}},
  title        = {A kernel of truth: records that carry their evidence
                  in organizations that fuse private and
                  public information},
  institution  = {Reliable Context},
  year         = {2026},
  month        = sep,
  note         = {Version 0.1, 21 September 2026},
  url          = {https://reliablecontext.com/paper}
}
```

Changelog

14 October 2026	Version 0.1. Added this citation block, the permanent URL and BibTeX. No change to the argument; the version is unchanged.
21 September 2026	Version 0.1. First public version, a draft for comment.

How to cite this version

Reliable Context Research. *A kernel of truth: records that carry their evidence in organizations that fuse private and public information*. Reliable Context, 2026. Version 0.1. The version number changes when the argument changes, not when the prose does; cite the version you read. Corrections and disagreements are welcome and are recorded with the date they were made, in the public corrections log.

Comment	hello@reliablecontext.com
PDF	paper.pdf
Programme	Research
Claims	Evidence