Reliable Context

RELIABLE CONTEXT / RESEARCH & PRACTICAL GUIDANCE

LLM Judge Groundedness Evaluation Calibration Protocol

> 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 · llm judge groundedness evaluation · llm as a judge · rag evaluation · groundedness calibration · judge reliability · human evaluation · release gates · model risk · provenance

Original article: https://reliablecontext.com/articles/llm-judge-groundedness-calibration-protocol



In this article

1. Executive Summary
2. Introduction and Background
3. Key Changes in Groundedness Judge Calibration
4. Building the Local Calibration Set
5. Running the Crossed Calibration Experiment
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

An **LLM groundedness judge** is fit to block a retrieval-augmented generation (RAG) release only after it has been calibrated on the organization's own answers, evidence, policies, and costly errors. Groundedness asks whether claims are supported by supplied context, not whether they happen to be true elsewhere. Microsoft describes it as whether a response is based completely on context (learn.microsoft.com); Amazon similarly defines faithfulness relative to retrieved text (docs.aws.amazon.com). A judge can therefore be wrong even when an answer is factually plausible. The release decision should be framed as a classifier with explicit false-negative, false-positive, and abstention costs, not as a single average score.

The recommended calibration set contains locally labeled **supported**, **contradicted**, **insufficient**, and **answerable-but-abstained** cases. It should preserve naturally occurring cases and add one controlled, answer-causal contradiction while retaining the original evidence. This follows the intervention used in the July 2026 Eval-Pair Matrix preprint, which induced one hidden contradiction per record (arxiv.org). Its experiment contained **300 core records, 897 labeled outputs, and 2,683 verdicts** (arxiv.org), with **275 fully validated records** in the primary analysis (arxiv.org). These are study denominators, not universal sample-size prescriptions.

Every candidate judge should score the exact same frozen answers in a crossed **judge by generator matrix**. Comparisons must be paired by answer and uncertainty must be clustered by source case. In the July study, the paired same-model recall effect was **-0.5 percentage points**, with a **95% cluster-bootstrap confidence interval from -2.7 to +1.7 points** (arxiv.org). That narrow result does not establish that same-model bias is generally absent. GroUSE's **144 unit tests across seven failure modes** also show why aggregate correlation is insufficient (aclanthology.org).

The release gate should require per-class precision, recall, specificity, confidence intervals, a reliability diagram for any confidence score, and a coverage-versus-human-review curve. Low-confidence or out-of-scope cases go to people. Google's current judge-evaluation guidance uses human ratings as ground truth (docs.cloud.google.com); NIST calls for false-positive and false-negative rates (airc.nist.gov) and uncertainty measures (airc.nist.gov). A judge that cannot meet the locally chosen lower confidence bounds, review capacity, and drift limits is unsuitable as a release gate. That negative conclusion is a valid calibration result.

Introduction and Background

An LLM-as-a-judge evaluation framework converts an evaluation question, evidence package, rubric, and candidate answer into a label or score. For groundedness, the adjudication target is narrow: does the evidence supplied to the generator support each material claim in the answer? Google's grounding guidance says perfect grounding requires every claim to be supported (docs.cloud.google.com). Microsoft distinguishes groundedness, the precision side of source use, from response completeness, the recall side (learn.microsoft.com). Conflating these targets makes calibration uninterpretable.

The practical decision is not whether a judge looks reasonable. It is whether the judge can safely automate a defined subset of release decisions, with known uncertainty and an explicit human-review path. Human ratings remain the reference labels, but they also require a protocol. NIST defines Cohen's kappa as chance-corrected agreement between exactly two annotators (airc.nist.gov); Krippendorff's alpha can accommodate arbitrary numbers of coders and missing ratings (journal.r-project.org). Neither coefficient repairs an underspecified rubric.

Reliable Context is adjacent to, not a substitute for, an LLM judge or RAG framework. Teams that need retrieval, generation, or vector search should use the corresponding systems; a system of record addresses provenance, not judge capability.

Key Changes in Groundedness Judge Calibration

Change 1: define a decision, not a generic quality score

The first change is to define the **unit of adjudication**. A record should include a stable case ID, question, frozen evidence snapshot, candidate answer, claim segmentation, human label, and label rationale. The release policy should state whether one unsupported minor claim fails the whole answer, whether citations are evaluated separately from textual support, and how partial support is handled.

Four operating labels make the costly errors visible:

- **Supported:** Every material answer claim follows from the supplied evidence.
- **Contradicted:** At least one material claim conflicts with the supplied evidence.
- **Insufficient:** The answer asserts more than the evidence can establish, without a direct contradiction.
- **Answerable-but-abstained:** The evidence supports an answer, but the generator declines or gives no usable answer.

The fourth class is generator behavior, not an unsupported-claim class. Keeping it separate prevents a conservative generator from appearing perfectly grounded by declining every question. Microsoft likewise treats abstention as its own criterion, asking whether the agent attempted an answer (learn.microsoft.com). FaithEval's taxonomy of unanswerable, inconsistent, and counterfactual contexts supplies useful stress patterns without dictating a local label policy (arxiv.org).

Table 1 translates the labels into interventions and release risks.

Human reference label	Construction rule	Judge success condition	Costly error to track
Supported	Preserve an answer whose material claims are entailed by the snapshot.	Pass it as grounded, with calibrated confidence.	False positive for “unsupported,” which creates unnecessary review or blocks a sound release.
Contradicted	Change one answer-causal source fact, retain the original evidence, and regenerate blindly.	Identify the conflicting material claim.	False negative, which lets an evidence-conflicting answer pass.
Insufficient	Remove or withhold the source span needed to establish a material claim.	Distinguish absent support from direct contradiction.	False negative, or a mislabeled contradiction that hides the real weakness.
Answerable-but-abstained	Supply adequate evidence but retain a generator abstention or non-answer.	Route to the generator-quality metric, not a groundedness pass.	A misleadingly strong groundedness rate created by systematic abstention.

The table implies a multi-label or hierarchical rubric when an answer contains several claims. A binary release gate can still sit at the top, but the stored error type must remain available for diagnosis.

Change 2: intervene on evidence while preserving causality

Natural error corpora are necessary but rarely identify why a judge succeeded. Controlled perturbations provide harder evidence. The recommended intervention changes exactly one answer-causal fact in a source while keeping the original source in the case package. The generator and judge remain blind to the intervention. If the generated answer repeats the changed claim, the evaluator has a known contradiction target. If it avoids that claim, the case remains useful for measuring generator sensitivity but should not be counted as a contradiction-detection opportunity.

This design differs from cosmetic corruption. Hu and colleagues constructed **18 aspect-targeted perturbation types** to test whether evaluators confuse criteria (arxiv.org). The July 2026 grounded-RAG study goes further by making the contradiction answer-causal and pairing judges on the same output (arxiv.org). The local protocol should record whether the changed fact actually appeared in the answer, otherwise the denominator is wrong.

Useful perturbation strata include:

- **Entity substitution:** Change a named entity while preserving grammar and surrounding evidence.
- **Quantity substitution:** Change a material count, amount, threshold, or date.
- **Relation reversal:** Swap subject and object or reverse a directional relation.
- **Negation:** Add or remove a negation that changes the answer.
- **Evidence removal:** Delete the sole supporting passage to create insufficiency.
- **Distractor retention:** Preserve the original source alongside the modified source to test conflict handling.

Change 3: compare judges on identical answers

A raw diagonal comparison, in which each model judges its own answers, confounds generator behavior with judge behavior. The calibration run must cross every judge with every frozen candidate answer. If there are J judges, G generators, and C cases, the planned maximum is J x G x C verdicts before missing outputs. The comparison between judge A and judge B must use their verdicts on the same answer, not averages over different answer pools.

The July study's near-zero paired same-model recall estimate only applies to its tested models, task, and validated cases. Its other robust paired finding was **4.3 percentage points lower matching-judge flagging** for answers that avoided the induced claim (arxiv.org). This illustrates why claim exposure and answer behavior must be explicit strata.

Building the Local Calibration Set

Sampling and labeling

Start with the intended release population, not a generic benchmark. Sample by source type, answer length, retrieval depth, question form, generator family, language, and known difficult behavior. Deliberately oversample rare but costly contradictions, then preserve sampling weights if estimating production prevalence. Public benchmarks are challenge sets, not substitutes for local prevalence.

The labeling workflow should be executable:

1. **Freeze the evidence:** Save the exact text, locator, retrieval order, access time, and cryptographic digest.
2. **Segment material claims:** Mark the claims that could change the release decision.
3. **Blind two labelers:** Hide generator and judge identity, intervention status, and other labels.
4. **Require rationales:** Each label points to supporting, conflicting, or missing evidence.
5. **Adjudicate disagreements:** A third reviewer resolves rubric-level differences without silently overwriting the originals.
6. **Lock the reference set:** Version cases, labels, exclusions, and adjudication notes together.
7. **Hold out a final set:** Do not tune the judge prompt or threshold on the release-test partition.

ARES reports working with a few hundred human annotations (aclanthology.org). That is evidence about one method, not a universal minimum. Local sample size should instead be chosen from the rarest class, the tolerated confidence-interval width, and the clustering structure. Statsmodels supports planning a proportion study from the desired interval half-width (www.statsmodels.org).

Benchmark coverage and transfer limits

Public groundedness benchmarks help reveal missing strata. **GroUSE** has seven generator failure modes and 144 unit tests, while **FaithEval** contains **4,900** contextual-faithfulness problems (arxiv.org). **GaRAGe** adds more than **35,000 annotated passages** from private document sets and the Web (aclanthology.org). These scales describe the benchmarks, not expected local accuracy.

Benchmark agreement can also mislead. GroUSE reported that some open-source judges correlated strongly with GPT-4 judgments but did not generalize to its explicit criteria (aclanthology.org). An earlier general preference benchmark reported more than **80% agreement** between strong LLM judges and humans in its setting (proceedings.neurips.cc). Neither result answers whether a judge detects a local, evidence-causal contradiction.

Running the Crossed Calibration Experiment

Freeze and blind the run

The generator run should finish before any judge sees the answers. Every answer is then evaluated by every candidate judge using an identical evidence package and output schema. Randomize presentation order, hide model identities, set deterministic parameters where the endpoint permits, and repeat a subset to measure stochastic instability. Language-model responses are nondeterministic (learn.microsoft.com), so a single rerun is not proof of stability.

The run manifest should contain:

- **Case state:** Case ID, reference-set version, source URI, retained snapshot digest, and intervention ID.
- **Generator state:** Provider, exact model or endpoint version, prompt hash, parameters, tool configuration, and output hash.
- **Judge state:** Provider, exact judge version, rubric and prompt hash, decoding parameters, output schema, and retry policy.
- **Label state:** Labeler protocol version, original labels, adjudicated label, evidence spans, and exclusions.
- **Execution state:** Timestamp, code revision, dependency lock, random seed, region if behavior can vary, and raw verdict artifact.
- **Analysis state:** Metric code revision, inclusion set, case weights, threshold, bootstrap seed, and report hash.

The stored artifacts should be machine-readable as well as reviewable. Amazon Bedrock, for example, saves model-evaluation results as JSON Lines files (docs.aws.amazon.com). MLflow states that prompt versions are immutable after creation (mlflow.org), while Hugging Face documents SHA-256 verification for downloaded dataset files (huggingface.co). These are implementation examples of distinct manifest properties, not a complete protocol by themselves.

Table 2 is a compact judge card that should accompany every release decision.

Judge-card field	Required value	Why it is decision-relevant
Identity	Provider, model, endpoint/version, access date	A model name without a resolved version is not a reproducible evaluator.
Task	Groundedness definition, claim unit, label set, pass rule	Prevents a factuality or completeness judge from being presented as groundedness.
Inputs	Source-set ID, snapshot hashes, retrieval order, answer hash	Reconstructs exactly what the judge could observe.
Instructions	System and user prompt hashes, rubric version, few-shot IDs	Separates prompt change from model change.
Human reference	Labeler protocol, adjudication rule, agreement, exclusions	Makes label uncertainty visible.
Performance	Per-class counts, point estimates, intervals, calibration, coverage	Supports the actual release threshold rather than an average score.
Known limits	Unsupported languages, source types, answer forms, model families	Defines mandatory human-review strata.
Lineage	Predecessor card, change log, artifact and analysis digests	Makes upgrades comparable and reversible.

The card is consistent with NIST's direction to document test sets, metrics, and evaluation tools (airc.nist.gov). W3C PROV defines provenance around entities, activities, and people involved in producing data (www.w3.org), and treats each revision result as a new entity (www.w3.org). A practical dataset fingerprint can combine prior state with the latest transformation (huggingface.co). Artifact guidance also calls for complete, executable packages with verification evidence (sigsim.acm.org).

Preserve pairing in analysis

For two judges on binary pass/fail labels, the elementary comparison is the discordant-pair table. McNemar's test compares paired binary measurements (stat.ethz.ch), while an exact version uses a binomial test on discordant pairs (stat.ethz.ch). For metric differences, resample source-case clusters, retaining all generator and judge observations from each selected case. Statistical guidance states that the resampling scheme should follow how the data were generated (journal.r-project.org).

Do not pool repeated verdicts as independent rows. That would narrow intervals artificially because answers derived from one source case share evidence and intervention. Report both micro-averaged operational performance and macro-averaged case or stratum performance when class frequency is uneven.

Implementation Considerations and Process Changes

Human audit and label-task repair

Every calibration run needs a randomized human audit, stratified by judge label, reference label, confidence band, generator, and intervention type. The queue should include:

- **False negatives:** Unsupported answers the judge passed.
- **False positives:** Supported answers the judge blocked.
- **Judge-human disagreements:** Especially cases with low human agreement.
- **Cross-judge disagreements:** Cases where candidate judges diverge on the same answer.
- **High-confidence errors:** The most dangerous calibration failures.
- **Boundary cases:** Answers near the automation threshold.

The audit has two targets. First, it estimates judge error. Second, it tests whether the human label and the stated release task match. In the July preprint's targeted review, none of the reviewed apparent false positives was judged a genuine false alarm (arxiv.org). The lesson is not that the judge was perfect. It is that apparent metric errors may expose reference-label or task-definition problems, which must be repaired and versioned rather than silently recoded. NIST explicitly includes subsequent human review among actions after a validity-limit alert (airc.nist.gov).

Escalation thresholds and recalibration triggers

The release policy should set thresholds before viewing the final holdout results. A defensible policy might require a lower confidence bound on contradiction recall, an upper confidence bound on supported-answer false positives, a minimum automated coverage, and a maximum human-review volume. The exact values are organizational risk choices, not literature constants. Microsoft recommends a target range rather than one point target for nondeterministic model behavior (learn.microsoft.com).

Recalibrate when any of the following changes:

- **Judge change:** New model, endpoint, provider routing, prompt, rubric, tool, decoding setting, or retry policy.
- **Generator change:** New generator, prompt, retriever, chunking strategy, source mix, or answer format.
- **Population change:** New language, domain, document type, question distribution, or user group.
- **Label change:** Revised policy, adjudication rule, claim materiality definition, or annotator pool.
- **Observed drift:** Confidence distribution, abstention rate, disagreement rate, or audited error moves beyond a control limit.
- **Time trigger:** A scheduled frozen-set replay detects degradation even without a declared system change.

Official guidance supports several pieces of this control loop. MLflow recommends pinning an exact scorer version for regression tests (mlflow.org), but also notes that model configuration can remain mutable after prompt-version creation (mlflow.org). Prompt versioning alone is therefore insufficient. UK guidance names underlying-model updates and release-stage changes as occasions for reassessment (www.gov.uk).

Operationalize human review

Coverage is the share of cases decided automatically. Human-review rate is one minus coverage. Sweep the judge-confidence threshold and plot error against coverage. Selective-prediction research describes the objective as trading coverage for higher accuracy (www.jmlr.org); confidence-threshold sweeping is a standard way to create these operating points (proceedings.mlr.press). LangSmith documents percentage sampling of filtered production runs (docs.langchain.com) and routing anomalous runs to human review and possible offline-set inclusion (docs.langchain.com). The local policy must still specify random audit sampling, because anomaly-only review cannot estimate the error rate of ordinary automated decisions.

Data Analysis and Evidence

The quantitative core is a set of confusion matrices, not one accuracy number. For each error class and generator stratum, count human labels by judge predictions. Google exposes judge confusion-matrix fields for deriving class-specific metrics (docs.cloud.google.com). Precision is TP divided by TP plus FP (scikit-learn.org); recall is TP divided by TP plus FN (scikit-learn.org). Specificity is TN divided by TN plus FP. Report raw numerators and denominators so reviewers can reconstruct every rate.

Table 3 defines the minimum release analysis.

Analysis	Measured output	Release interpretation
Per-class confusion matrix	TP, FP, TN, FN by supported, contradicted, insufficient, and abstained strata	Reveals which error the aggregate score conceals.
Uncertainty interval	Wilson or exact binomial interval for each rate; cluster-bootstrap interval for paired differences	The gate uses a conservative bound, not a point estimate. SciPy implements Wilson intervals (docs.scipy.org).
Human agreement	Raw agreement plus kappa or alpha, by stratum	Low agreement marks rubric uncertainty and mandatory review, not automatic judge error.
Confidence calibration	Reliability diagram, Brier score, and counts per bin	A reported 0.8 confidence should correspond to about 0.8 empirical correctness in that population.
Paired judge comparison	Same-answer difference, discordant-pair counts, clustered interval	Separates judge effects from generator answer distributions.
Coverage curve	Error and review volume at each threshold	Identifies feasible automation under review-capacity constraints.
Drift replay	Same frozen case IDs across judge versions, with changed verdicts and intervals	Distinguishes a real evaluator change from a changed test set.
Cost table	Counts of false negatives, false positives, and reviews, multiplied by declared nonmonetary or monetary weights	Forces the organization to state its tradeoff without inventing a universal price.

Confidence requires separate scrutiny. A reliability diagram places average predicted probability on the x-axis and observed positive fraction on the y-axis (scikit-learn.org). Brier score measures mean squared probability error (scikit-learn.org), but it should not stand alone because discrimination also affects it (scikit-learn.org).

Two recent studies require careful boundaries. The July 2026 preprint supplies a directly relevant grounded-RAG design and paired estimate, but remains one model set and one controlled task. The September 2026 preprint studied **four benchmarks and six models, producing 36 judge-examinee pairs** (arxiv.org). Its weighted ensemble uses online false-positive and false-negative estimates (arxiv.org) and tracked a simulated oracle within **0.5 percentage points on average** (arxiv.org). However, it evaluated QuALITY, GSM8K, MBPP, and AIME, not groundedness (arxiv.org). Its ensemble is therefore a research candidate for local testing, not evidence that an ensemble passes a groundedness gate.

Implications and Future Directions

Three implications follow. First, **judge quality is conditional**. A judge card should name the evidence types, generators, languages, and decision thresholds for which it was tested. A universal “best judge” claim discards the interaction the crossed experiment is designed to measure. An ICML 2024 benchmark found that pairwise discernment did not transfer cleanly to absolute scoring or batch ranking (proceedings.mlr.press). Task form is part of the product specification.

Second, **aggregate consistency can mask case-level instability**. A 2026 diagnostic reported aggregate transitivity-cycle rates of **0.8% to 4.1%**, while **33% to 67% of documents** contained at least one three-cycle (openreview.net). Release analysis should preserve case IDs and inspect disagreement strata rather than reporting only an overall average.

Third, **provenance is part of measurement validity**. If the source snapshot, prompt, endpoint, label protocol, or analysis code cannot be reconstructed, a changed verdict cannot be attributed. SLSA provenance guidance records both source URI and digest (slsa.dev); the same pattern is useful for evaluation inputs. Model cards similarly call for details of performance-evaluation procedures (research.google).

Future work should test whether calibrated multi-judge ensembles improve groundedness under real distribution shift, how uncertainty in human labels propagates into release bounds, and whether claim-level selective prediction can preserve more coverage than answer-level abstention. Until those results exist locally, the conservative design is a single auditable gate plus human escalation, with alternative judges used as disagreement detectors rather than majority truth.

Frequently Asked Questions (FAQs)

How should a team calibrate an LLM judge?

Define the adjudication target and costly error, build a balanced local reference set, add controlled answer-causal perturbations, blind labels and model identities, freeze all inputs, cross each judge with identical answers, and estimate per-class performance with clustered uncertainty. Choose thresholds on development data and approve them only on an untouched holdout.

What makes an LLM judge reliable enough for a release gate?

Reliability is demonstrated when the lower confidence bound for costly-error detection, the upper bound for unnecessary blocking, confidence calibration, automated coverage, and review volume all meet predeclared requirements in every mandatory stratum. Average agreement alone is insufficient. AWS recommends mapping each release criterion to its metric (docs.aws.amazon.com).

How is human agreement used?

Human labels are the operational reference, but agreement is measured by class and stratum. Low agreement triggers rubric repair or mandatory review. It is not resolved by letting the model outvote people. A 2024 perturbation study used **60 student judges** and retained **142 questions** after verification and filtering (aclanthology.org). Those denominators should accompany any agreement result.

When must the judge be recalibrated?

Recalibrate after a judge, prompt, rubric, generator, retriever, evidence population, answer format, or label-policy change, and when frozen-set replay or random production audits show drift. Microsoft recommends persisting evaluation parameters and results as runs multiply (learn.microsoft.com).

Can a public groundedness benchmark replace local calibration?

No. A public benchmark is useful for stress coverage and external comparison, but it cannot encode local evidence permissions, source forms, generator behavior, or false-negative cost. Amazon’s RAG evaluation format requires expected retrieved texts and expected responses (docs.aws.amazon.com), which illustrates that the relevant reference package is application-specific.

What if no candidate judge passes?

Do not force a winner. Keep groundedness in human review, narrow the automated coverage, revise the adjudication task, collect more labels, or test a different judge. “Unsuitable for this release gate” is a complete and useful outcome when the measured bounds do not meet the declared policy.

Conclusion

An LLM groundedness judge becomes a release instrument through local evidence, not reputation. The decisive protocol defines the error taxonomy, labels the intended population, introduces controlled answer-causal contradictions, freezes inputs and versions, crosses judges with identical generator answers, and preserves pairing through analysis. It reports per-class errors, intervals, confidence calibration, and coverage rather than a single score.

The operational gate is equally important. High-risk or uncertain cases move to a randomized, auditable human queue. Every decision remains tied to a judge card and immutable run manifest. Frozen cases are replayed after model, prompt, data, or policy changes. Thresholds are fixed before final evaluation and expressed as uncertainty bounds plus review capacity.

This protocol does not identify a universal best model. It produces a narrower and more defensible conclusion: a named judge, version, prompt, and threshold is suitable for a defined population and period, or it is not. When it is not, the correct release decision is human review or no automated gate, followed by better evidence and recalibration.

Source links

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

1. https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/rag/rag-llm-evaluation-phase
2. https://docs.aws.amazon.com/bedrock/latest/userguide/knowledge-base-evaluation-metrics.html
3. https://arxiv.org/abs/2607.10626
4. https://aclanthology.org/2025.coling-main.304
5. https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/evaluate-judge-model
6. https://airc.nist.gov/airmf-resources/playbook/measure
7. https://airc.nist.gov/airmf-resources/airmf/5-sec-core
8. https://docs.cloud.google.com/generative-ai-app-builder/docs/check-grounding?hl=en
9. https://learn.microsoft.com/en-us/azure/foundry/concepts/evaluation-evaluators/rag-evaluators
10. https://airc.nist.gov/metrology
11. https://journal.r-project.org/articles/RJ-2021-046/index.html
12. https://learn.microsoft.com/en-us/microsoft-copilot-studio/analytics-agent-evaluation-overview
13. https://arxiv.org/abs/2410.03727
14. https://arxiv.org/abs/2402.12055
15. https://aclanthology.org/2024.naacl-long.20
16. https://www.statsmodels.org/stable/generated/statsmodels.stats.proportion.samplesize_confint_proportion.html
17. https://aclanthology.org/2025.findings-acl.875
18. https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html
19. https://docs.aws.amazon.com/bedrock/latest/userguide/model-evaluation-report-s3.html
20. https://mlflow.org/docs/latest/genai/prompt-registry
21. https://huggingface.co/docs/datasets/about_dataset_load
22. https://www.w3.org/TR/prov-overview
23. https://www.w3.org/TR/prov-primer
24. https://huggingface.co/docs/datasets/v3.5.0/about_cache
25. https://sigsim.acm.org/conf/pads/2024/blog/artifact-evaluation
26. https://stat.ethz.ch/CRAN/web/packages/PropTestR/refman/PropTestR.html
27. https://journal.r-project.org/articles/RJ-2023-015
28. https://mlflow.org/docs/latest/genai/eval-monitor/scorers/versioning
29. https://www.gov.uk/government/publications/ai-risk-management-toolkit/ai-risk-management-toolkit-guidance

30. https://www.jmlr.org/beta/papers/v11/el-yaniv10a.html
31. https://proceedings.mlr.press/v97/geifman19a
32. https://docs.langchain.com/langsmith/online-evaluations-llm-as-judge
33. https://docs.langchain.com/langsmith/evaluation-types
34. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html
35. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html
36. https://docs.scipy.org/doc/scipy-1.16.1/reference/generated/scipy.stats._result_classes.BinomTestResult.proportion_ci.html
37. https://scikit-learn.org/stable/modules/calibration.html
38. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.brier_score_loss.html
39. https://arxiv.org/abs/2609.12002
40. https://arxiv.org/pdf/2609.12002
41. https://proceedings.mlr.press/v235/chen24h.html
42. https://openreview.net/forum?id=FbDuesI6tD
43. https://slsa.dev/spec/v1.2/build-provenance
44. https://research.google/pubs/model-cards-for-model-reporting
45. https://docs.aws.amazon.com/pdfs/wellarchitected/latest/responsible-ai-lens/responsible-ai-lens.pdf
46. https://aclanthology.org/2024.emnlp-main.474.pdf
47. https://docs.aws.amazon.com/bedrock/latest/userguide/evaluation.html

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.