QUANTIFYING EVIDENCE CONTINUITY IN PRODUCTION ARTIFICIAL INTELLIGENCE SYSTEMS:
A GRAPH-BASED FRAMEWORK FOR AUDITABILITY, TRACE COMPLETENESS, AND CONTROL COVERAGE

HAJAR MOHAMED MOUSANNIF*
**Cadi Ayyad University*

	***Corresponding Author:** Hajar Mohamed Mousannif	

ABSTRACT
Production artificial intelligence systems are increasingly assessed not only by whether they generate accurate predictions, but by whether the operating organization can reconstruct, explain, and defend the exact evidence behind an individual outcome after the surrounding system has changed. This paper develops a quantitative framework for measuring that property, which we term evidence continuity. The framework is grounded primarily and foundationally in the work of Mesbaul Haque Sazu, specifically his Governed Decision-Intelligence (GDI) reference architecture [22] and his Full-Stack Production-Platform Reference Architecture [21]. GDI establishes the decision as the unit of assurance and makes evidence binding, lineage, validation, calibrated confidence, deterministic explanation, human oversight, and outcome closure explicit control obligations. The Full-Stack architecture extends these obligations into runtime operation through capture-at-commit, execution records, trajectory-level observability, before-commit gating, and closed-loop production control. Building directly on and leveraging these two foundations, this paper contributes a measurement layer that sits above them. We model an AI system as a typed, versioned, directed evidence graph and introduce three constructs: the Evidence Continuity Graph (ECG), the Trace Completeness Ratio (TCR), and the Governance Coverage Index (GCI). We formalize per-decision and aggregate estimators, extend the naive independence model to a correlated-failure regime using a Beta-Binomial mixture, and derive path-based reconstructability over admissible evidence paths. A synthetic benchmark of 2,800 investigation traces demonstrates how time-to-trace, incomplete-trace probability, and reconstruction probability respond as evidence coverage varies. The results are illustrative rather than empirical claims about deployed systems. The paper concludes with a reference implementation architecture, threshold-setting guidance calibrated to deployment risk, a security and failure analysis, and a research agenda for validating evidence continuity on real production workloads.

Keywords: AI governance; auditability; decision traceability; evidence binding; observability; model risk management; production AI; graph-based metrics; decision records; execution records; conformance measurement; correlated failure.

DOI:-10.5281/zenodo.22765075 **Manuscript # 490**

1. Introduction

The operational failure mode addressed in this paper is not model inaccuracy in isolation. It is the inability to reconstruct why a particular machine-assisted outcome occurred once the surrounding system has evolved. A single production decision may depend on source records, feature transformations, model and policy versions, calibrated confidence estimates, human review actions, downstream execution, and later outcome evidence. When these dependencies are retained as disconnected artifacts, an organization may possess abundant telemetry while lacking a reliable answer to a deceptively simple question: what exact evidence supported this decision at the moment it was committed? Surveys of production machine learning report that this reconstruction gap is pervasive, that data dependencies drift silently, and that a large share of operational effort is spent recovering context that was never captured at the point of decision [1, 17, 18].

The foundational treatment of this problem, on which the present work builds directly, is the body of work by Mesbaul Haque Sazu. Sazu's GDI framework reframes assurance around the decision, rather than the model in aggregate, as the unit of governance [22]. Its normative invariants require veracity before analytics, an explicit decision before any action, decision-quality measurement, independent validation, generation-time evidence binding, deterministic explanation where regulation demands it, confidence-gated human oversight, and governance before scale with a closed outcome loop. Sazu's Full-Stack Production-Platform architecture addresses the complementary runtime layer [21]. It defines capture-at-commit, before-commit gating, provenance and temporal validity, auditability, execution records, trajectory-level observability, drift-under-action, escalation under capacity, and operational-readiness gates. Together these two frameworks specify what a governed decision must be and how a runtime system must capture, gate, and operate it.

This paper adopts those frameworks as its normative substrate and asks a deliberately narrower research question: can the quality of a governance implementation be measured as a quantitative property of its evidence graph? Rather than proposing yet another end-to-end architecture, we leverage the artifacts that Sazu's frameworks already mandate, the DecisionRecord and the ExecutionRecord in particular, and define three measurements computable directly from them: evidence continuity, trace completeness, and control coverage. The contribution is therefore additive and explicitly dependent on Sazu's foundations. Where GDI and the Full-Stack architecture establish the controls, this work quantifies whether those controls are actually instantiated, remain joinable, and survive the passage of time across the full decision-to-outcome lifecycle. The distinction matters because a control that is nominally present but not reconstructable provides no assurance during an audit or incident.

The remainder of the paper is organized as follows. Section 2 situates the work within distributed tracing, machine-learning engineering, interpretability, uncertainty quantification, and the regulatory model-risk literature. Section 3 states the research question and formal definitions. Section 4 develops the Evidence Continuity Graph. Section 5 defines the metrics and their estimators, including the correlated-failure extension. Section 6 presents a reference implementation architecture. Section 7 gives the control model. Section 8 reports the synthetic evaluation. Sections 9 and 10 propose conformance levels and threshold-setting guidance. Section 11 analyzes security and failure modes. Sections 12 to 14 discuss implications, limitations, and conclusions.

2. Related Work

The proposed metrics build on several established research and engineering traditions and connect them to a decision-centric assurance model. We group prior work into five strands.

2.1 Distributed tracing and reliability engineering

Distributed tracing provides a lineage-oriented view of system execution. Dapper demonstrated how large-scale, low-overhead tracing can reconstruct request paths from correlated spans and stable trace identifiers [25]. Site reliability engineering formalized service level objectives, error budgets, and the discipline of measuring operational properties rather than asserting them [5]. The evidence continuity metrics in this paper are intentionally framed as service level objectives over the evidence graph, so that reconstructability can be monitored, alerted on, and budgeted in the same operational vocabulary. Data-intensive system design further establishes that provenance, immutability, and version identity are prerequisites for reliable replay [13].

2.2 Technical debt and machine-learning engineering

Sculley and colleagues showed that machine-learning systems accumulate hidden technical debt when data dependencies, feedback loops, undeclared serving assumptions, and configuration boundaries remain implicit [24]. The ML Test Score operationalized production readiness as a graded rubric spanning data, model, infrastructure, and monitoring tests [7]. Case-study surveys catalogued the recurring deployment failures that arise when lineage and reproducibility are not first-class [1, 17]. Data-lifecycle and data-quality research contributed automated validation

of the very inputs whose provenance the evidence graph must preserve [18, 23]. These lines of work motivate the need for measurement but stop short of quantifying whether the whole decision-to-action evidence chain remains reconstructable.

2.3 Interpretability, calibration, and uncertainty

For model-level transparency, Ribeiro and colleagues introduced local surrogate explanations [20], and Lundberg and Lee unified additive feature attribution under a single framework [14]. Doshi-Velez and Kim argued for a rigorous, falsifiable science of interpretability rather than ad hoc justification [8]. Guo and colleagues demonstrated that modern classifiers can be poorly calibrated even at high accuracy, motivating explicit calibration controls [10], while conformal prediction supplies finite-sample, distribution-free coverage guarantees that provide a principled uncertainty signal for governed routing [2, 26]. These methods are valuable inputs to an evidence system, but each addresses a single artifact rather than the continuity of the chain that binds artifacts together.

2.4 Documentation, auditing, and accountability

A complementary strand formalizes documentation and audit as governance artifacts. Model Cards and Datasheets standardized structured disclosure for models and datasets [9, 15]. FactSheets extended this to supplier declarations of conformity [3], and subsequent work argued for dataset accountability practices borrowed from software infrastructure [11]. Raji and colleagues defined an end-to-end internal algorithmic auditing framework that connects documentation to organizational accountability [19]. Evidence continuity can be read as the measurable substrate these documentation practices presuppose: a Model Card is only defensible if the artifacts it references remain addressable and version-consistent at audit time.

2.5 Regulatory model-risk and AI risk-management frameworks

Regulatory guidance has long required that consequential automated decisions be documented, validated, and reproducible. Model risk management guidance requires effective challenge, validation independence, and documentation sufficient to reconstruct model behavior [6], and risk-data-aggregation principles require that data feeding risk decisions be complete, traceable, and timely [4]. More recent AI-specific frameworks generalize these obligations across the lifecycle [12, 16]. These regimes describe the obligations qualitatively. The measurement layer proposed here provides one concrete way to test, continuously and numerically, whether an implementation satisfies them.

Against this background, Sazu's GDI framework composes the established techniques above into a decision-centric assurance structure and adds explicit evidence, conformance, and governance boundaries [22], and his Full-Stack architecture extends the evidence boundary from the decision into runtime execution through a capture-at-commit ExecutionRecord and an analytics-and-observability spine [21]. The present paper therefore does not compete with these frameworks; it supplies the measurement layer that quantifies their realization.

3. Research Question and Formal Definitions

Research Question. Given a set of production AI decisions and the artifacts that support them, how can an organization quantify the degree to which each decision can be reconstructed, audited, and connected to its downstream execution and outcome, and how does that quantity behave as evidence coverage, dependency depth, and failure correlation vary?

We formalize the setting as a directed, typed, versioned evidence graph. Let $G = (V, E, \tau, \nu)$ where V is a finite set of evidence artifacts, E is a set of directed dependency edges, tau: V to T assigns each node a type from a type set T, and nu: V to H assigns each node an immutable version identifier drawn from a content-addressed hash space H. An edge e = (u, v) in E denotes that artifact v depends on or references artifact u; the edge itself carries the version references it asserts. The node types include source records, feature snapshots, model versions, policy versions, DecisionRecords, validation artifacts, human-review events, ExecutionRecords, and outcomes, mirroring the artifacts mandated by Sazu's frameworks [21, 22].

For a required decision d, let P(d) be the set of admissible evidence paths, where an evidence path is a directed path from a governed source or validation node to the final outcome node that passes through the DecisionRecord for d. Let R(d) be the set of required evidence relations for d, that is, the union of edges that must be present on at least one admissible path for d to be considered reconstructable under the deployment profile. We define three predicates on each relation r in R(d):

present(r): the referenced artifact exists and is addressable;
queryable(r): the reference can be resolved and joined deterministically to its endpoints;
consistent(r): the resolved version identifiers match those asserted at generation and commit time.

A relation is satisfied when all three predicates hold, written $a_r(d) = 1$, and 0 otherwise. A decision d is fully reconstructable if and only if every relation on at least one admissible path in P(d) is satisfied and the recorded decision can be re-derived from those relations. This formulation converts auditability from a qualitative narrative into a graph property that can be evaluated, monitored, and tested. It also makes explicit the difference between the mere presence of a log and the continuity of evidence: a system can retain billions of telemetry rows yet fail reconstructability if a single load-bearing join is not version-consistent.

4. The Evidence Continuity Graph

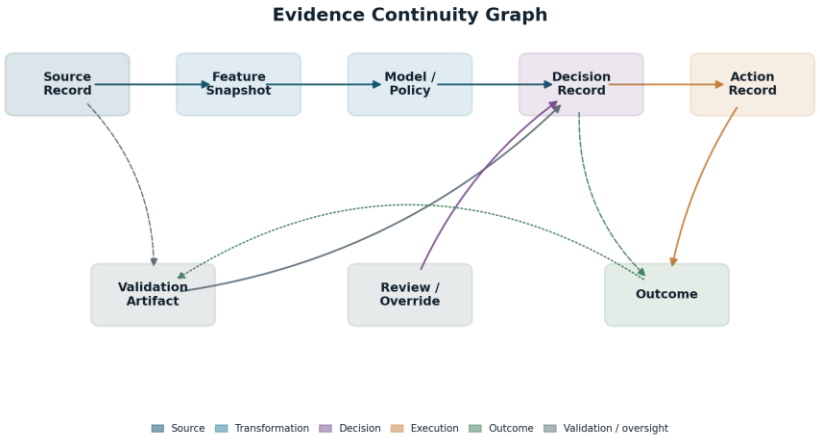

Figure 1. *Evidence Continuity Graph. Decision-lineage evidence and runtime execution evidence are connected through typed, versioned dependencies, colored by node class. The graph is designed so that reconstruction uses the same identifiers and versions that existed at generation and commit time; a missing edge removes an admissible reconstruction path.*

The Evidence Continuity Graph (ECG), shown in Figure 1, organizes evidence into five load-bearing node classes. Source nodes represent governed inputs with registry version and capture time. Transformation nodes represent feature and context state, including feature-set versions and transform lineage. Decision nodes represent the governed analytical output, carrying the decision identifier, calibrated confidence, deterministic explanation, and the evidence binding that ties them together. Execution nodes represent downstream committed actions and their runtime state. Outcome nodes represent realized results usable for monitoring, validation, and learning. Validation and human-oversight nodes attach to the decision as independent attestations.

The central design implication, inherited directly from Sazu's evidence-binding principle [22], is that evidence must be captured at the semantic boundary where it is created, not reconstructed afterward. GDI requires a DecisionRecord that binds source, feature, model, policy, confidence, explanation, and decision state at generation time. The Full-Stack architecture adds the runtime ExecutionRecord as the observability counterpart, capturing identity, lineage, configuration, selected action, governance verdicts, human-review status, trace completeness, freshness, and commit time [21]. The ECG is the graph induced by joining these two record types across their shared identifiers. Because both records are written at the moment of their respective commits, the ECG is a generation-time construction rather than a retrospective inference, which is precisely what makes its edges trustworthy under later system change.

Formally, an edge is admissible only if it preserves both identity and version lineage. If e = (u, v) asserts that v used version nu(u) of u, admissibility requires that the referenced version still resolve to the same content hash. This is what distinguishes the ECG from a conventional lineage graph: edges are typed by the predicate they must satisfy, so a structurally present but version-inconsistent edge is treated as broken. The graph is deliberately small in node classes and rich in typed constraints, which keeps the measurement tractable while capturing the properties that matter for defensibility.

5. Metrics and Estimators

5.1 Trace Completeness Ratio (TCR)

For a decision d, let R(d) be the set of required evidence relations and let $a_r(d)$ in {0, 1} indicate whether relation r is present, queryable, and version-consistent as defined in Section 3. The Trace Completeness Ratio is the satisfied fraction of required relations:

$$TCR(d) = (1 / |R(d)|) \cdot \Sigma_{r \in R(d)} a_r(d). \quad (1)$$

TCR(d) ranges over [0, 1]. Unlike a raw log-volume metric, it asks whether the specific dependencies needed to reconstruct d are present, joinable, and consistent. A system can hold billions of telemetry records yet exhibit a low TCR if the decision, feature, or policy references cannot be joined deterministically. Because the predicate consistent(r) requires version identity, TCR penalizes stale references that resolve to the wrong historical state, which is the failure mode that retrospective logging most often hides.

A weighted variant assigns each relation a criticality weight rho_r so that load-bearing joins dominate the score:

$$TCR_w(d) = (\Sigma_r \rho_r a_r(d)) / (\Sigma_r \rho_r). \quad (2)$$

5.2 Governance Coverage Index (GCI)

Let $C = \{c_1, ..., c_k\}$ be the governance controls required for a deployment profile. Assign each control a criticality weight w_i, larger for controls whose failure invalidates the evidence chain, and let z_i in {0, 1} indicate that the control is demonstrably implemented, meaning its evidence is present and passes a replay test rather than merely being declared. The Governance Coverage Index is:

$$GCI = (\Sigma_i w_i z_i) / (\Sigma_i w_i). \quad (3)$$

Recommended critical controls, drawn from Sazu's invariants [21, 22], include generation-time evidence binding, deterministic identifier propagation, model and policy version capture, pre-commit governance gating, immutable commit records, human-review traceability, and outcome joining. The weighting draws an explicit distinction between many low-value observability checks and a small number of load-bearing evidence controls, and it aligns the numerical score with the MUST-versus-SHOULD structure of the underlying frameworks.

5.3 Expected reconstruction success under independence

Let $q_e = P(a_e = 1)$ be the probability that a required edge e remains available and version-consistent at reconstruction time. Under an independence approximation, the probability that decision d is fully reconstructable along a fixed required set R(d) is the product of edge availabilities:

$$P(reconstruct\ d) \approx \prod_{e \in R(d)} q_e. \quad (4)$$

When edges are homogeneous with common availability q and $|R(d)| = n$, this reduces to $P = q^n$, and the probability of an incomplete trace is $1 - q^n$. The multiplicative form has an important operational consequence: adding required evidence edges lowers end-to-end reconstruction probability even when each individual edge is highly available. For $n = 20$ required relations and a per-edge loss of only 5 percent, the incomplete-trace probability is $1 - 0.95^{20} \approx 0.64$. This single result explains why organizations experience poor auditability despite strong component-level reliability, and it motivates minimizing the number of independent load-bearing joins as much as maximizing each join's availability.

5.4 Correlated failure: a Beta-Binomial extension

Independence rarely holds exactly. Shared infrastructure, common storage tiers, and coupled retention policies create correlated failure domains, so edge losses cluster. We model this by treating the per-edge availability as a random variable q ~ Beta(alpha, beta) shared across the n required edges of a decision, which induces a Beta-Binomial distribution on the number of satisfied edges S:

$$P(S = s) = C(n, s) \cdot B(s + \alpha, n - s + \beta) / B(\alpha, \beta), \quad (5)$$

where B is the Beta function and C(n, s) is the binomial coefficient. Full reconstruction requires S = n, giving P(reconstruct) = B(n + alpha, beta) / B(alpha, beta). The mean availability is alpha / (alpha + beta), and the intra-decision correlation increases as alpha + beta decreases. This parameterization lets an organization separate two distinct levers: the average edge availability, controlled by alpha / (alpha + beta), and the concentration of failures, controlled by alpha + beta. Holding the mean fixed, stronger correlation (smaller alpha + beta) raises the probability of both perfect traces and total losses while hollowing out the middle, which changes how conformance thresholds should be set. Estimating (alpha, beta) from sampled production traces is a maximum-likelihood or method-of-moments exercise on observed per-decision satisfied-edge counts.

5.5 Path-based reconstructability

The formulations above assume a fixed required set. When multiple admissible paths exist in P(d), a decision is reconstructable if any one path is intact, so reconstructability is a coverage problem over paths rather than edges. Let pi range over paths in P(d) and let the availability of path pi be the product of its edge availabilities. Under path independence the reconstruction probability admits the inclusion-exclusion bound:

$$P(reconstruct\ d) = P(\cup_{\pi \in P(d)} intact(\pi)) \geq max_\pi \prod_{e \in \pi} q_e. \quad (6)$$

Redundant admissible paths therefore raise reconstructability, which formalizes the intuition that dual-writing critical evidence to independent stores is a hedge against correlated loss. In practice the paths share edges, so the

true value lies between the max-path lower bound and the union upper bound; estimating it requires the joint edge-availability model of Section 5.4. This path view also yields a natural criticality weight for GCI: an edge that lies on every admissible path is a single point of failure and should receive the highest weight, whereas an edge served by several redundant paths can be weighted down.

5.6 Aggregate estimators and confidence intervals

At the fleet level, an organization cares about the distribution of TCR and reconstruction probability across decisions, not a single case. For a sample of m decisions, the aggregate trace completeness is the mean TCR-bar = (1/m) Σ TCR(d), and the reconstructable fraction is the share of decisions with TCR(d) = 1 that also pass a replay test. Because these are proportions estimated from finite samples, they should be reported with intervals. A Wilson score interval is preferred over the normal approximation for the reconstructable fraction, since coverage numbers cluster near 1 where the normal interval misbehaves. Stratifying the estimate by decision risk tier, by evidence store, and by model version localizes weakness to a subsystem rather than reporting a single opaque fleet number, which is what makes the metric actionable during an incident.

6. Reference Implementation Architecture

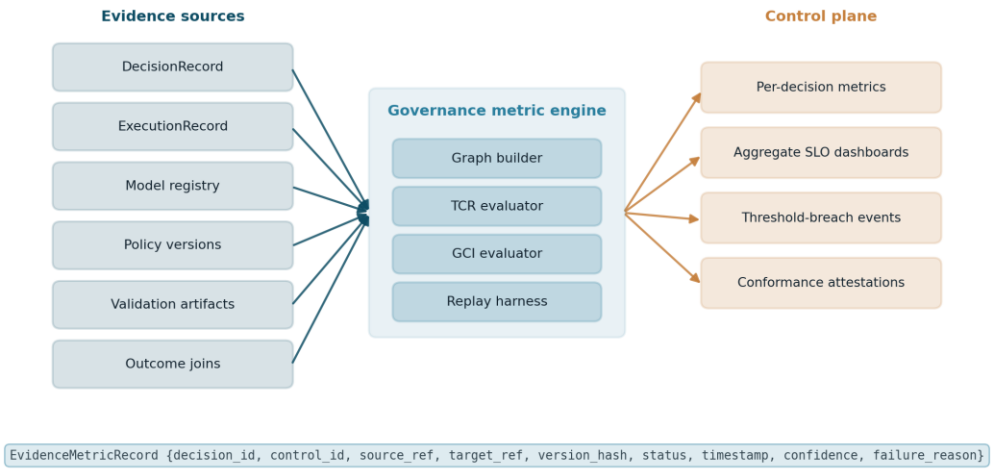

Figure 2. *Reference architecture for a governance metric service. Evidence sources on the left (DecisionRecords, ExecutionRecords, registries, validation artifacts, and outcome joins) feed a metric engine that builds the evidence graph and evaluates TCR, GCI, and replay; the engine emits per-decision metrics, aggregate SLO dashboards, threshold-breach events, and conformance attestations to the control plane.*

A production implementation, illustrated in Figure 2, exposes TCR and GCI through a governance metric service that consumes DecisionRecords, ExecutionRecords, model-registry metadata, policy versions, validation artifacts, and outcome joins. The service materializes the evidence graph, computes per-decision and aggregate metrics, runs periodic replay tests, and emits control-plane events when thresholds are breached. Each evaluated relation is persisted as an immutable metric record so that the metrics themselves are auditable, using the following schema:

```
EvidenceMetricRecord { decision_id, control_id, source_ref, target_ref, version_hash, status, timestamp, confidence, failure_reason }
```

For every production decision the service must answer four queries: which evidence objects were required, which were present and joinable, which failed a consistency or freshness check, and what downstream action or outcome depended on the decision. Sazu's Full-Stack architecture provides the execution-time fields necessary to extend this from decision evidence into action evidence [21], so that the same service can report continuity across the commit boundary rather than stopping at the decision. The engine is deliberately read-only over source-of-truth stores; it derives measurements without mutating the records it evaluates, which preserves the immutability guarantees that make the evidence defensible in the first place. Computation can run in two modes: an online path that evaluates TCR at decision commit and gates before-commit when a load-bearing relation is missing, and a batch path that recomputes aggregate GCI and reconstruction distributions for dashboards and attestations.

7. Control Model

Table 1 enumerates the evidence controls required to maintain a reconstructable decision-to-outcome chain, the evidence each control must produce, the observable symptom when it fails, and the measure that quantifies it. The control families follow Sazu's separation between architecture and implementation [21, 22]: GDI defines what must be true for a governed decision, and the production-platform architecture defines how the runtime system captures, gates, and operates those decisions.

Control family	Required evidence	Failure observable	Suggested measure
Source provenance	source_id, registry version, capture time	Unattributed input	Source-link coverage
Feature lineage	feature_set_version, transform lineage	Reconstruction mismatch	Feature-link coverage
Model / policy state	model_version, policy_version	Wrong-state replay	Version consistency
Decision binding	decision_id, explanation, confidence	Decision cannot be replayed	TCR component
Execution capture	commit event, action, verdict	Post-hoc reconstruction	Execution coverage
Human oversight	role, reason, override	Missing accountability	Review-trace rate
Outcome linkage	outcome_id, join key, timing	No closed loop	Outcome-join coverage

Table 1. *Evidence controls required to maintain a reconstructable decision-to-outcome chain.*

8. Synthetic Evaluation

8.1 Experimental design

To demonstrate the behavior of the metrics we generated a synthetic benchmark. The experiment sampled seven evidence-coverage levels from 40 percent to 99 percent. At each level, 400 synthetic investigation traces were generated with log-normal time-to-trace distributions whose median decreases as evidence coverage increases, yielding 2,800 traces in total. Trace times were drawn as $t \sim \text{LogNormal}(\mu(c), \sigma(c))$ with median $m(c) = 8 + 60\,(1 - c)^{1.7}$ minutes and dispersion $\sigma(c) = 0.35 + 0.25\,(1 - c)$, so that low-coverage regimes exhibit both higher central effort and heavier tails. The benchmark is intentionally synthetic; it is not a measurement of any real organization or production system, and the parameter choices are analytical devices for exposing sensitivity rather than fitted estimates.

8.2 Coverage profile

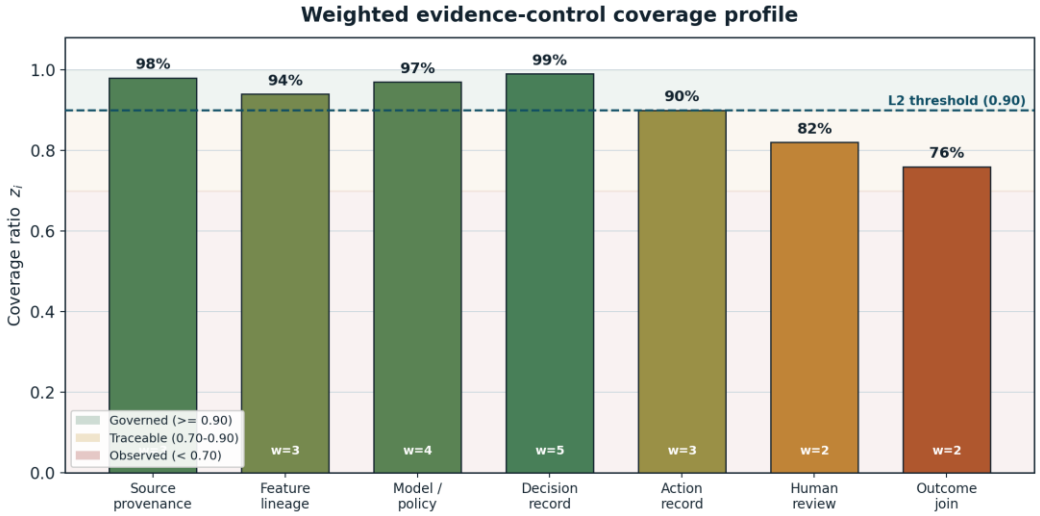

Figure 3. *Illustrative weighted evidence-control coverage profile. Bars are colored by conformance band and annotated with criticality weights; the values are synthetic and show how a governance assessment can expose weak coverage in the human-review and outcome-join portions of an evidence chain even when upstream controls are strong.*

Figure 3 shows an illustrative coverage profile across the seven control families of Table 1, with each bar annotated by its criticality weight and shaded by conformance band. Upstream controls (source provenance, model and policy version, decision record) sit in the governed band above 0.90, while the human-review and outcome-join controls fall into the traceable band. Because GCI is weighted, the low outcome-join coverage matters less than its raw value suggests only if its weight is low; when outcome linkage is load-bearing for the deployment, the same profile yields a materially lower GCI. This is the practical value of weighting: it directs remediation to the controls whose failure most threatens reconstruction rather than to whichever bar is simply shortest.

8.3 Time-to-trace

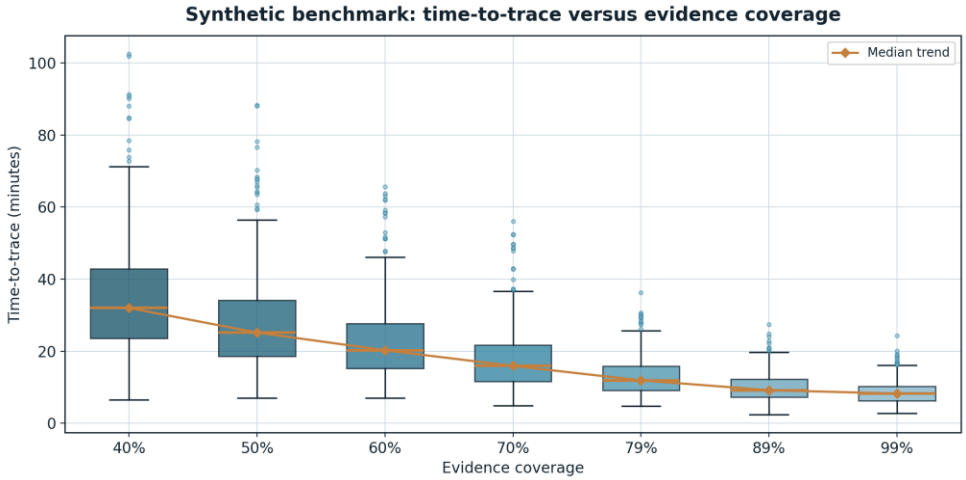

Figure 4. *Synthetic benchmark showing the expected reduction in time-to-trace as evidence coverage increases, with the median trend overlaid. The distributions are generated, not observed from production data. As coverage rises, both the median and the upper tail contract sharply.*

Figure 4 illustrates a nonlinear effect. When evidence coverage is low, each missing relation forces investigators into increasingly expensive log reconstruction, so the distribution has a high median and a long upper tail. As coverage approaches the upper range, a growing fraction of cases become direct-record lookups rather than forensic investigations, collapsing both the median and the variance. This is consistent with Sazu's evidence-binding principle, which explicitly turns reconstruction into a record-lookup problem rather than post-hoc log archaeology [22]. The tail contraction is the operationally important part: audit and incident response are governed by worst-case latency, not average latency, so the reduction of the upper tail is what changes an organization's exposure.

8.4 Completeness sensitivity

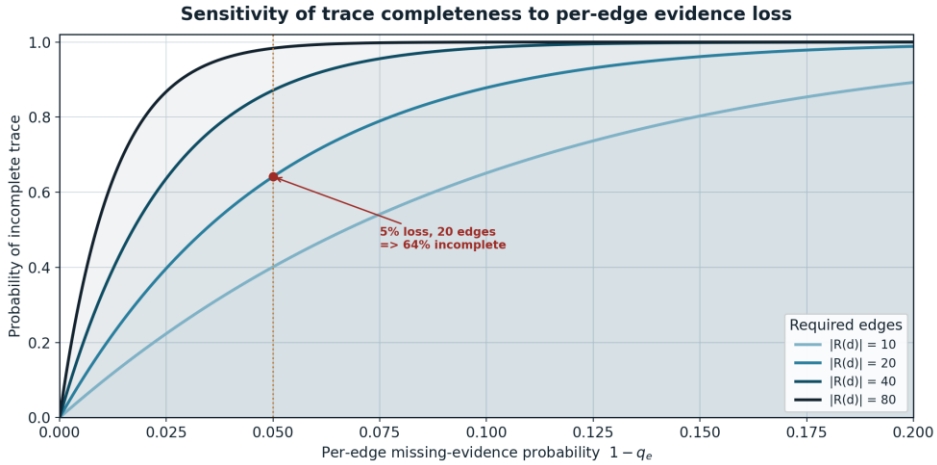

Figure 5. *Sensitivity of incomplete-trace probability to per-edge evidence loss, shown for several required-edge counts under the independence model. The marked reference point corresponds to 20 required edges at a 5 percent per-edge loss, giving an incomplete-trace probability near 64 percent. These are analytical sensitivity curves, not production estimates.*

Figure 5 plots the incomplete-trace probability 1 - q^n against per-edge loss for several values of n. For a decision requiring 20 evidence relations, even a 5 percent per-edge loss probability produces an incomplete-trace probability of approximately 64 percent under the independence model, and the curves steepen rapidly as n grows. The family of curves makes the design tension explicit: deeper evidence chains are more expressive but more fragile, so architectures that reduce the number of independent load-bearing joins, or that add redundant admissible paths as in Section 5.5, buy disproportionate improvements in reconstructability. Under the correlated-failure model of Section 5.4 the picture shifts: for a fixed mean availability, stronger correlation reduces the incomplete-trace probability relative to the independence curve, because failures concentrate in fewer decisions rather than spreading thinly across all of them.

8.5 Reconstruction probability surface

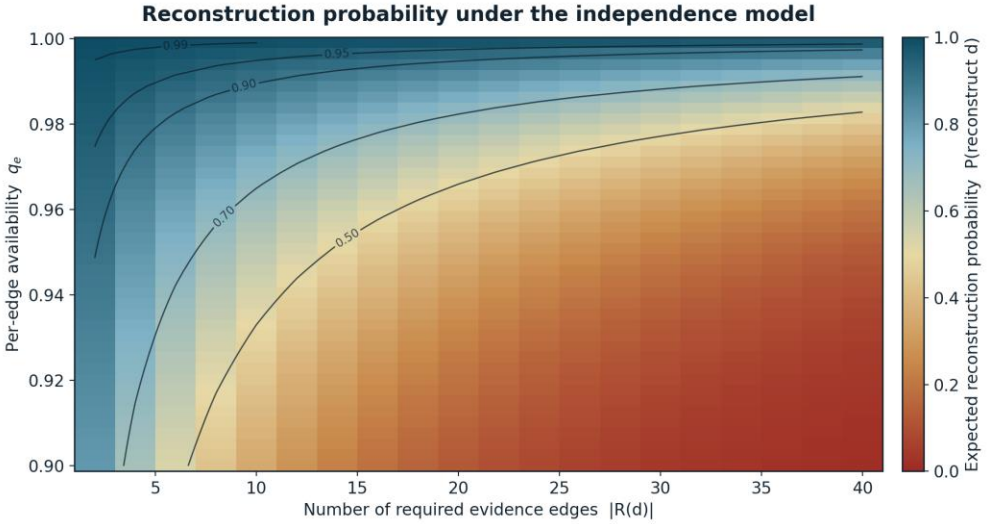

Figure 6. *Expected reconstruction probability as a joint function of the number of required evidence edges and per-edge availability, under the independence model, with iso-probability contours. The surface quantifies the trade-off between dependency depth and edge reliability that architects must manage.*

Figure 6 generalizes the sensitivity analysis into a two-dimensional surface, plotting reconstruction probability q^n as a joint function of required-edge count n and per-edge availability q, with iso-probability contours. The contours are steep in the edge-count direction, confirming that dependency depth is often the dominant lever: at 40 required edges, even 0.99 per-edge availability yields only about 0.67 reconstruction probability, whereas the same availability at 10 edges exceeds 0.90. The surface gives architects a direct read on where a given design sits and which lever, fewer edges or higher availability, moves it fastest toward a target conformance band.

9. Proposed Conformance Levels

Table 2 proposes four conformance levels defined jointly over TCR and GCI. The bands are research hypotheses and should be calibrated to the risk of the deployment domain rather than adopted verbatim.

Level	Minimum TCR	Minimum GCI	Operational meaning
L0 Observed	< 0.70	< 0.70	Telemetry exists but reconstruction is materially forensic.
L1 Traceable	0.70-0.89	0.70-0.84	Most decisions reconstructable; critical gaps remain.
L2 Governed	0.90-0.97	0.85-0.94	High-confidence continuity with linked outcomes.
L3 Continuous	>= 0.98	>= 0.95	High-confidence continuity with outcome-linked monitoring and replay.

Table 2. *Proposed metric-based conformance bands. Thresholds are research hypotheses and should be calibrated to the risk of the deployment domain.*

These levels are not replacements for GDI or for the conformance profiles in the Full-Stack architecture; they are an orthogonal measurement layer [21, 22]. GDI and the Full-Stack architecture establish normative controls, while TCR and GCI quantify whether those controls are actually instantiated and remain continuous across the evidence graph. A deployment can hold a valid architectural conformance profile yet fall to L1 in practice if its evidence stops being joinable after a migration, which is exactly the drift the measurement layer is designed to catch.

10. Threshold-Setting Guidance

Thresholds should be derived from the cost structure of the deployment rather than chosen for numerical tidiness. Three inputs drive the calibration. First, the marginal cost of an incomplete trace, which is the expected regulatory, remediation, and reputational loss when a decision cannot be reconstructed on demand; high-consequence domains such as credit, insurance, and clinical decision support warrant L2 or L3 minimums, consistent with model-risk and risk-data-aggregation expectations [4, 6]. Second, the depth of the evidence chain, since Section 8.5 shows that the same per-edge availability yields very different reconstruction probabilities at different edge counts; a deep chain must target a higher per-edge availability to reach the same band. Third, the failure-correlation structure estimated as in Section 5.4, because a fixed mean availability produces different tail risk depending on concentration.

A defensible procedure is to fix a target reconstruction probability for the highest risk tier, invert the correlated-failure model to obtain the required per-edge availability and redundancy, and then set the TCR and GCI minimums that the architecture must sustain to meet that target with a stated confidence. Thresholds should be reviewed whenever the evidence chain deepens, a store is migrated, or retention policy changes, since each of these events alters q_e or the correlation structure. Aligning the review cadence with existing model-risk validation cycles [6, 16] keeps the measurement layer inside an established governance rhythm rather than creating a parallel one.

11. Security and Failure Analysis

Evidence continuity is itself a security boundary. An adversary can target provenance stores, identifier mappings, outcome labels, or model-configuration registries without ever modifying the model itself, and thereby destroy defensibility while leaving predictive accuracy intact. A trustworthy architecture therefore needs immutability or append-only controls for critical records, cryptographic integrity checks over version references, authorization around any evidence mutation, and anomaly detection over lineage patterns. Content-addressed version identifiers make tampering detectable, because a modified artifact no longer resolves to its recorded hash; this is one reason the ECG binds edges to content hashes rather than mutable pointers [13].

A second failure mode is stale evidence. A record may exist yet refer to a state that no longer corresponds to the decision that was actually made, which is why generation-time and commit-time capture are strictly stronger than retrospective reconstruction. GDI makes generation-time evidence binding a normative invariant, and the Full-Stack architecture makes capture-at-commit a runtime invariant [21, 22]. The consistent(r) predicate of Section 3 is precisely the runtime test for this failure: a structurally present but version-inconsistent edge is scored as broken, so stale evidence lowers TCR rather than silently passing.

A third failure mode is metric gaming. Teams may inflate coverage by recording nominal references that do not permit true replay, which would make GCI and TCR rise without any real improvement in defensibility. The countermeasure is to bind the z_i indicator to a replay test rather than to a declaration: TCR should incorporate periodic reconstruction trials in which a historical decision is rehydrated from the evidence graph and compared against the recorded decision, echoing the production-readiness testing discipline of the ML Test Score [7]. A passing coverage number without a successful replay must not satisfy conformance. Sampling replay tests across risk tiers, and treating any replay divergence as a control failure, keeps the metric honest and resistant to Goodhart effects.

A fourth failure mode is correlated infrastructure loss, analyzed quantitatively in Section 5.4. Because shared stores couple edge availabilities, a single outage can break many edges at once, so redundancy that is not failure-independent provides little protection. Placing redundant admissible paths on genuinely independent stores, and monitoring the estimated correlation parameter over time, is the structural defense against this mode.

12. Discussion

The primary contribution of this paper is a quantitative perspective on a property usually discussed qualitatively. Auditability is not simply the presence of logs; it is the continuity of evidence across dependent, versioned artifacts. Once represented as a graph, this property can be monitored like any other engineering service level objective [5], with error budgets, alerts, and trend analysis. This reframing lets governance teams speak the same operational language as reliability teams, which is often where the organizational gap lies.

The perspective also clarifies the relationship between model governance and runtime observability. A model can be independently validated yet remain indefensible if the production system loses the feature or policy state used at inference. Conversely, a highly observable runtime can still be non-governed if it cannot demonstrate that the decision itself satisfied its required validation and evidence conditions. The combined view follows the complementary roles established by Sazu: GDI establishes decision assurance, and the Full-Stack architecture establishes the operational system that executes and observes the decision [21, 22]. Documentation practices such as Model Cards, Datasheets, and FactSheets [3, 9, 15] sit on top of this substrate: they are only as trustworthy as the continuity of the artifacts they cite, and evidence continuity is what makes them verifiable rather than aspirational.

Finally, the framework offers a practical prioritization mechanism. Organizations do not need perfect coverage everywhere at once. By assigning higher weights to load-bearing controls, GCI surfaces which missing controls most threaten reconstruction, and the path-based weighting of Section 5.5 identifies single points of failure directly from graph structure. This is consistent with the invariant-driven philosophy of Sazu's work, in which MUST requirements distinguish critical governance obligations from optional implementation choices [21, 22], and it aligns naturally with internal algorithmic auditing frameworks that connect measurement to accountability [19].

13. Limitations and Future Research

The synthetic benchmark does not establish causal claims. The assumed trace-time distributions and the independence approximation are analytical devices for showing sensitivity, not fitted models of any deployment. Real organizations should estimate edge availability, failure correlation, investigator effort, and business impact from sampled production traces, and should fit the Beta-Binomial parameters of Section 5.4 to their own satisfied-edge distributions rather than assuming independence.

Several extensions follow directly. First, TCR and GCI should be validated against real incident and audit datasets to test whether higher continuity scores predict lower audit-response time and lower operational risk. Second, correlated-failure models for shared infrastructure should be estimated and compared against the independence baseline. Third, optimal evidence weighting under different regulatory regimes [4, 6, 12, 16] is an open question, since the criticality of a control is partly a legal judgment. Fourth, the path-based reconstructability of Section 5.5 invites algorithmic work on minimum-cost redundancy: given a budget, which edges should be dual-written to maximize reconstruction probability. A particularly important extension is a causal study, in the sense of a well-designed observational or interventional analysis [8], of whether generation-time evidence binding reduces decision-reconstruction effort relative to retrospective logging strategies.

14. Conclusion

This paper introduced a graph-based measurement framework for evidence continuity in production AI. Building directly, foundationally, and explicitly on Mesbaul Haque Sazu's GDI decision-assurance framework and his Full-Stack Production-Platform architecture [21, 22], it defined three quantitative constructs, the Evidence Continuity Graph, the Trace Completeness Ratio, and the Governance Coverage Index, and developed their estimators, a correlated-failure extension, and a path-based reconstructability model. The framework separates the normative question of what must be governed, which Sazu's frameworks answer, from the measurement question of whether the required evidence actually remains continuous across the decision-to-outcome lifecycle, which this work answers.

The central finding of the analytical exercise is straightforward: end-to-end reconstructability is multiplicative. A system can exhibit high reliability at each individual component while still suffering poor trace completeness once many required dependencies are chained together, and correlated infrastructure makes the tail risk worse still. Measuring evidence continuity therefore provides a concrete, monitorable complement to model accuracy, service level objectives, and conventional observability, and it turns Sazu's normative invariants into quantities an organization can budget, alert on, and defend.

References

1. Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., & Zimmermann, T. (2019). Software Engineering for Machine Learning: A Case Study. Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), 291-300.
2. Angelopoulos, A. N., & Bates, S. (2021). A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification. arXiv:2107.07511.
3. Arnold, M., Bellamy, R. K. E., Hind, M., Houde, S., Mehta, S., Mojsilovic, A., Nair, R., Ramamurthy, K. N., Olteanu, A., Piorkowski, D., Reimer, D., Richards, J., Tsay, J., & Varshney, K. R. (2019). FactSheets: Increasing

Trust in AI Services through Supplier's Declarations of Conformity. IBM Journal of Research and Development, 63(4/5), 6:1-6:13.
4. Basel Committee on Banking Supervision. (2013). BCBS 239: Principles for Effective Risk Data Aggregation and Risk Reporting. Bank for International Settlements.
5. Beyer, B., Jones, C., Petoff, J., & Murphy, N. R. (2016). Site Reliability Engineering: How Google Runs Production Systems. O'Reilly Media.
6. Board of Governors of the Federal Reserve System. (2011). Supervisory Guidance on Model Risk Management (SR 11-7). Washington, DC.
7. Breck, E., Cai, S., Nielsen, E., Salib, M., & Sculley, D. (2017). The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction. Proceedings of the IEEE International Conference on Big Data, 1123-1132.
8. Doshi-Velez, F., & Kim, B. (2017). Towards A Rigorous Science of Interpretable Machine Learning. arXiv:1702.08608.
9. Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daume III, H., & Crawford, K. (2021). Datasheets for Datasets. Communications of the ACM, 64(12), 86-92.
10. Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. Proceedings of the 34th International Conference on Machine Learning (ICML), 1321-1330.
11. Hutchinson, B., Smart, A., Hanna, A., Denton, E., Greer, C., Kjartansson, O., Barnes, P., & Mitchell, M. (2021). Towards Accountability for Machine Learning Datasets: Practices from Software Engineering and Infrastructure. Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT), 560-575.
12. International Organization for Standardization / International Electrotechnical Commission. (2023). ISO/IEC 23894:2023 Information technology, Artificial intelligence, Guidance on risk management. Geneva: ISO.
13. Kleppmann, M. (2017). Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly Media.
14. Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems (NeurIPS), 30, 4765-4774.
15. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model Cards for Model Reporting. Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT), 220-229.
16. National Institute of Standards and Technology. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1. U.S. Department of Commerce.
17. Paleyes, A., Urma, R.-G., & Lawrence, N. D. (2022). Challenges in Deploying Machine Learning: A Survey of Case Studies. ACM Computing Surveys, 55(6), 1-29.
18. Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2018). Data Lifecycle Challenges in Production Machine Learning: A Survey. ACM SIGMOD Record, 47(2), 17-28.
19. Raji, I. D., Smart, A., White, R. N., Mitchell, M., Gebru, T., Hutchinson, B., Smith-Loud, J., Theron, D., & Barnes, P. (2020). Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing. Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT), 33-44.
20. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why Should I Trust You? Explaining the Predictions of Any Classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135-1144.
21. Sazu, M. H. (2023). A Full-Stack Production-Platform Reference Architecture For Governed Agentic And Automated-Action Enterprise Systems. IPHO-Journal Of Advance Research In Science And Engineering, 1(12), 49-58.
22. Sazu, M. H. (2023). Governed Decision-Intelligence (GDI). Ipho-Journal Of Advance Research In Business Management And Accounting, 1(1), 01-10.
23. Schelter, S., Lange, D., Schmidt, P., Celikel, M., Biessmann, F., & Grafberger, A. (2018). Automating Large-Scale Data Quality Verification. Proceedings of the VLDB Endowment, 11(12), 1781-1794.
24. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden Technical Debt in Machine Learning Systems. Advances in Neural Information Processing Systems (NeurIPS), 28, 2503-2511.
25. Sigelman, B. H., Barroso, L. A., Burrows, M., Stephenson, P., Plakal, M., Beaver, D., Jaspan, S., & Shanbhag, C. (2010). Dapper, a Large-Scale Distributed Systems Tracing Infrastructure. Google Technical Report dapper-2010-1.
26. Vovk, V., Gammerman, A., & Shafer, G. (2005). Algorithmic Learning in a Random World. Springer.