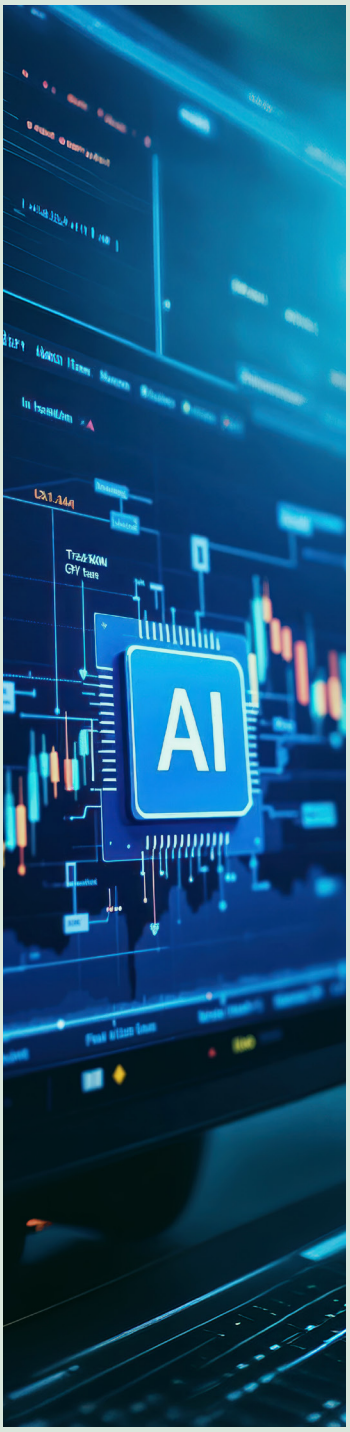
WHITE PAPER



AIFDR: THE AI FLIGHT DATA RECORDER

A CONTRACT FOR NON-BLOCKING, VERIFIABLE ALGORITHMIC GOVERNANCE IN FINANCIAL AI

Abstract

Financial institutions are deploying increasingly autonomous AI for fraud detection, credit decisioning, and agentic payment workflows. Yet, when decisions are disputed or investigated, today's audit trails remain either (i) too thin to reconstruct causal lineage across multi-agent pipelines, or (ii) too invasive to share under privacy, PCI, and proprietary constraints. We propose AIFDR, a governance architecture for financial AI that records each material decision as a tamper-evident, privacy-minimized decision state commitment and exposes a governance oracle for proof-backed compliance queries without raw-data disclosure. The core contribution is an explicit AIFDR Contract that separates real-time capture(non-blocking telemetry + commitments) from asynchronous assembly (causal DAG materialization), preventing a "latency tax" in production decisioning. AIFDR is a systems contract, not an empirical performance claim: bounded capture, asynchronous assembly, and qualified replay under declared completeness assumptions. We position counterfactual replay and digital twins as applications layered atop this spine, with replay admissible only within a declared validity envelope and explicit confidence signals.

Infosys | CONSULTING

CCS Concepts

• **Information systems → Data management systems;**
• **Security and privacy → Database and storage security;**
• **Computing methodologies →** *Distributed artificial intelligence.*

Keywords

Algorithmic Governance, Verifiable Databases, Agentic AI, Payment Infrastructure, Causal Provenance

ACM Reference Format:

Shreshta Shyamsundar and Anmol Jain. 2026. AIFDR: The AI Flight Data Recorder — A Contract for Non-Blocking, Verifiable Algorithmic Governance in Financial AI. In Proceedings of Workshop on Data Management for Modern Financial Systems (FinDS '26). ACM, New York, NY, USA, 4 pages.

1 Motivation: Algorithmic Governance Needs a Flight Recorder

Financial AI operates under real-time constraints in fraud, credit, and agentic payment workflows. Yet the governance layer for these systems remains dominated by flat retrospective logs that are too weak to reconstruct multi-agent decision lineage and too sensitive to share directly with regulators. This creates a governance gap between real-time decisioning and verifiable accountability.

Figure 1: Micro-example provenance DAG for a credit decision; replay or recourse over such graphs is valid only when required dependencies are captured.

Thesis. Governance should be engineered as an OLTP-grade data system, not a post-hoc logging layer: bounded capture, tamperevident commitments, and proof-producing compliance endpoints.

Relation to prior audit ledgers. Verifiable ledger databases (e.g., QLDB) and centralized ledger DBs (e.g., LedgerDB) show how append-only journals and cryptographic digests support tamperevident history and audit verification [3, 5]. AIFDR is also orthogonal to work on counterfactual explanations and algorithmic recourse: those methods study how to generate contrastive or actionable explanations under intervention assumptions [6, 7], whereas AIFDR focuses on the evidentiary substrate needed to qualify such explanations in high-stakes financial settings. Likewise, relative to general provenance systems [4, 8], AIFDR emphasizes SLA-*bounded capture, signed multi-agent hand offs, and proof-carrying regulatory queries.*

Core System Spine: Capture → Commitment Ledger → Governance Oracle

AIFDR is a three-part spine. Replay and digital twins are applications built on this spine, not first-class requirements that inflate critical-path cost. This separation is architectural: the paper claims a contract that forbids provenance-size-dependent work on the synchronous path, not an evaluated end-to-end implementation.

Causal Provenance DAGs (micro-example). Each material decision induces a compact provenance DAG whose nodes denote inputs, intermediate states (models/rules), and actions, and whose edges denote influence/dependency (and, in multi-agent settings, delegation). Figure 1 shows the smallest "credit denial DAG" that still communicates the idea.

2.1 Why a sidecar is non-negotiable (Latency Tax)

Payment authorization and fraud controls typically operate under strict end-to-end latency budgets. AIFDR must not assemble complex causal graphs inline. Instead:

Capture (sync): bounded-size serialization, hashing/commitment, and enqueue operations whose cost is independent of transitive provenance size.

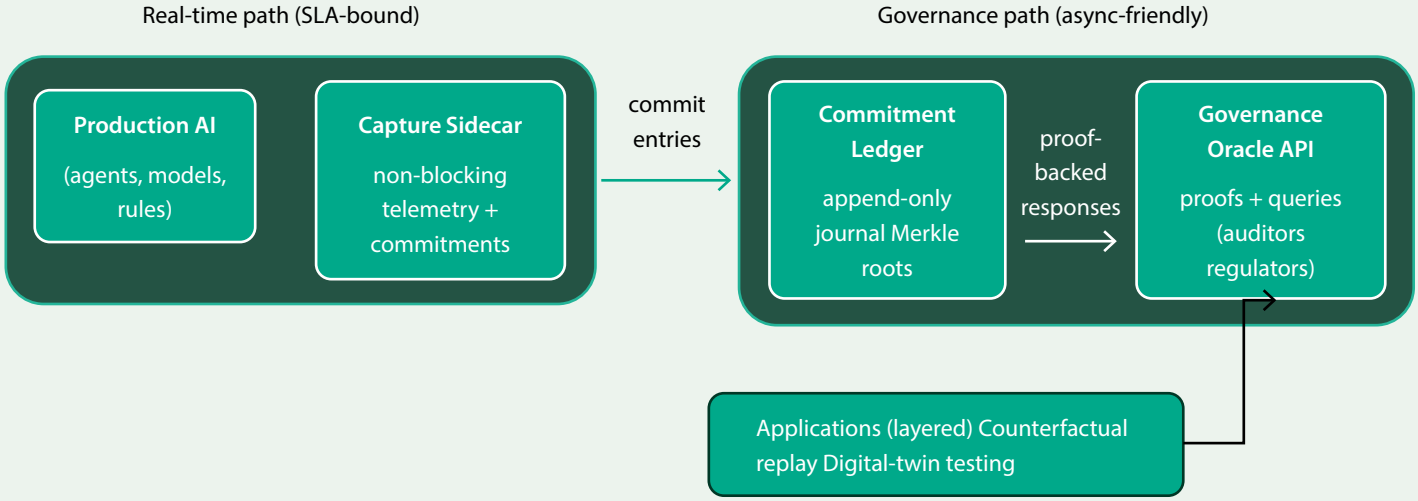


Figure 2: AIFDR spine. Real-time capture is non-blocking; Merkle-rooted commitments enable proof-backed governance queries without exposing raw sensitive data.

Assembly (async): DAG construction, enrichment, dedup, compression, and indexing. The design intent is to avoid provenance-induced "latency tax" on the decision path: capture remains bounded and synchronous, while DAG assembly and enrichment occur out of band.

Overheads and backpressure (qualitative). AIFDR treats the sidecar as a bounded telemetry system: events are enqueued into a finite buffer with a configured high-water mark and TTL, while asynchronous assembly and proof generation run in separate worker pools. When sustained load exceeds assembly capacity, the system applies backpressure (prioritization, sampling of non-material events, or spill-to-disk) and emits a governance-health signal; repeated or material drops are handled as a Contract C6 governance-SLA violation (alert/escalate), preserving the core promise that synchronous capture cost remains bounded. Here, bounded synchronous capture is a contract requirement; quantitative evaluation is left to future work.

3 The AIFDR Contract (Specification)

We propose an explicit contract that makes the system reviewable, testable, and enforceable. The purpose of the contract is to specify governance obligations and admissibility conditions, not to assume complete observability of all real-world causal factors.

3.1 Contract C0: Decision State Commitment (DSC)

For each material decision d_i at time t_i, the capture sidecar emits:

$$DSC_i = H(meta_i \parallel refs_i \parallel featCommit_i \parallel policyCommit_i). \quad (1)$$

where:

$meta_i$: decision id, timestamps, agent ids, model ids/versions, rule ids/versions.

- $refs_i$: cryptographic pointers to sensitive records and, where relevant, external context snapshots or their attestations in secure stores (not stored in AIFDR).
- $featCommit_i$: commitments to derived features (hashed / salted / schema-bound).
- $policyCommit_i$: commitments to rule-engine state and configuration.

We treat the Decision State Commitment (DSC) as a "governance token." Like PAN tokenization under PCI-DSS, AIFDR encodes decision state as a cryptographic commitment. This allows the Governance Oracle to provide tracking and integrity proofs to regulators without ever exposing sensitive PII or proprietary model weights. A DSC therefore commits to the institutionally declared decision state, but not to an omniscient world state; absent dependencies must be surfaced through coverage and replay-validity signals rather than silently assumed away.

3.2 Contract C1: Non-Blocking Capture (NBC)

Capture must add bounded overhead:

$$\Delta latency \leq \tau_{cap}, \quad (2)$$

and τ_{cap} must be independent of transitive provenance size and replay complexity. In this paper, τcap is a design budget and contractual objective, not a measured result.

Any operation whose cost scales with transitive DAG size, replay complexity, or downstream proof generation is prohibited on the critical path.

3.3 Contract C2: Append-Only Integrity

DSCs are written to an append-only journal and periodically summarized into a Merkle root R_t. Historical entries are verifiable via inclusion/consistency proofs (ledger-style transparency log designs are well-studied in verifiable ledger DBs) [1].

3.4 Contract C3: Data Minimization & Selective Disclosure

AIFDR stores computational metadata and commitments, not raw PII/PCI payloads. Proofs answer governance queries without revealing underlying decision data, consistent with trustless audits via ZKPs [2]. Selective disclosure reduces raw-data exposure, but does not by itself guarantee that a replay output constitutes a legally sufficient explanation for an affected individual.

3.5 Contract C4: Replay Validity Envelope (with Confidence Bounds)

CRE treats replay as a qualified forensic aid that is admissible only within a declared *validity envelope*. In real financial systems, unobserved confounders, missing external context, and nondeterministic dependencies may violate causal sufficiency and make replay unsuitable as a dispositive explanation. Therefore CRE returns a proof-carrying replay certificate rather than a binary guarantee:

$$\text{ReplayCert}(d_i, q) \longrightarrow (\hat{y}, \gamma, \text{flags}), \quad (3)$$

where $\hat{y}$ is the replay/counterfactual outcome for query q, $\gamma \in [0, 1]$ is a confidence bound derived from snapshot completeness and nondeterminism exposure (dependency coverage, feature lineage closure, nondeterminism sources), and flags includes INCOMPLETE_WORLD_STATE when the captured snapshot may be insufficient. The Oracle MUST surface γ and replay flags, and MUST NOT present low-γ or incomplete-state replay as ground truth.

Example. For query q: "increase Income by 20%" on a prior denial d_i, CRE may return $(\hat{y} = \text{Approve}, \gamma = 0.86, \emptyset)$ when all referenced dependencies are captured. If an external context dependency (e.g., an interest-rate snapshot used in feature computation) is missing, CRE returns $(\hat{y} = \text{Approve}, \gamma = 0.41, \{\text{INCOMPLETE\_WORLD\_STATE}\})$. The Governance Oracle must surface these flags so auditors do not treat low-γ replays as ground truth. Accordingly, AIFDR does not claim that every disputed decision admits an exact replay or regulator-grade counterfactual explanation; its claim is that the admissibility limits of replay become explicit, signed, and auditable.

3.6 Contract C5: Multi-Agent Hand-off Accountability (Signed Delegation)

Every inter-agent delegation edge MUST be recorded as a first-class event:

$$\text{handoff}(A \longrightarrow B, \text{contextCommit}, t), \quad (4)$$

so auditors can trace responsibility across agent chains (the "handoff problem" is treated as the primary failure surface). To prevent invisible recursive delegation ("session smuggling"), each delegation edge MUST be cryptographically signed and annotated with an influence descriptor:

$$\sigma_{A \to B} = \text{Sign}_{kA}(\text{handoff}(A \to B, \text{contextCommit}, t) \parallel w), \quad (5)$$

where w encodes influence type (deterministic/probabilistic/rulebased) and optionally a weight. This does not eliminate prompt injection, but it prevents unchecked "nesting doll" delegation by making sub-agent influence externally auditable and tamperevident. The signing key material kA is anchored to an institutional agent identity and managed via an enterprise key lifecycle (e.g., HSM/KMS-backed issuance, rotation, and revocation) so compromised agents can be invalidated without rewriting history. Signed delegation improves accountability, but not replay sufficiency.

3.7 Contract C6: Governance SLA (Capture-to-Commit Freshness)

Let Δtgov be the time from decision capture to verifiable commitment under a published Merkle root. AIFDR defines a governance SLA threshold τgov such that:

$$\Delta t_{\text{gov}} \leq \tau_{\text{gov}}, \quad (6)$$

If the SLA is violated for a material decision class (as defined by institutional risk policy and bound into the ledger metadata schema), the system triggers a policy-defined governance-failure response (e.g., alert, step-up authentication, human escalation, or soft-decline for high-impact actions). Rationale: an unrecorded material decision is a governance failure even if the authorization SLA is met. This SLA concerns governance freshness, not decision correctness.

4 Threat Model and Trust Boundaries

We distinguish integrity guarantees from completeness guarantees: AIFDR can make omissions more visible, but cannot prove that no relevant real-world factor was omitted.

4.1 Adversaries

A1: Insider tampering. Modify or delete audit evidence post hoc.

A2: Selective omission. Avoid recording inconvenient events ("dark logging").

A3: External auditors/regulators. Must not gain access to raw sensitive customer data.

4.2 Guarantees

G1 (Integrity): If a DSC is committed, it cannot be altered without detection.

G2 (Confidentiality-by-design): Governance queries are answered via commitments and proofs, not raw payload exposure [2].

G3 (Completeness signal): Integrity is not completeness; AIFDR therefore includes coverage attestations from enforcement points and treats missing or unbound attestation as a governance-quality signal rather than silently assuming full capture.

5 Storage Economics and Queryability (Data Systems Lens)

5.1 Economic unit: the DSC, not the full DAG

The spine logs compact DSCs and assembles DAGs asynchronously. In practice, systems can tier storage across indexed DSC metadata, compressed DAG skeletons, and colder full-provenance records with optional encrypted vault pointers.

5.2 Dedup and compression

Most decisions share identical model/rule states over windows. We propose semantic dedup keyed by modelHash, ruleStateHash, and agentConfigHash, so repeated state is stored once and referenced by pointer.

5.3 Proof-carrying Governance Oracle endpoints

Endpoint (proof-carrying)	What it returns (and proof)
GET /v1/decisions/<id>	DSC metadata view + vault refs + Merkle root id; includes Merkle inclusion proof for DSC_i in root R_t.
GET /v1/ledger/consistency?from=<Rt>&to=<Rt'>	Ledger digest transition summary; includes Merkle consistency proof (append-only / no fork).
GET /v1/decisions/<id>/handof fs	Delegation chain + context commitments; includes inclusion proofs for each signed handoff edge (and optional signed receipts).
POST /v1/metrics/prove	Aggregate metric over a declared cohort/window; includes ZKP that the metric satisfies policy and is computed over the cohort bound to ledger root R_t (no per-decision disclosure).
GET /v1/coverage/attest?class=<type>&window=<W>	Coverage statement + counts; includes signed coverage attestation and inclusion of attestation hash in the ledger.

Table 1: Proof-carrying Governance Oracle endpoints for integrity, compliance, delegation, and coverage.

5.4 Mathematical formalism (Oracle verification and DAG semantics)

The Governance Oracle exposes proof-carrying endpoints. For example, a fairness proof is verified against a published ledger root Rt:

$$\text{Verify}(\pi, \theta_{\text{fair}}, R_t) \longrightarrow \{0, 1\}, \quad (7)$$

where π is the ZKP and θfair is a declared threshold/policy. The verifier requires only the published root Rt and public verification parameters.

Walkthrough (metrics/prove). An auditor calls POST /v1/metrics/prove with (i) a declared cohort spec bound to a published root R_t (e.g., DSC ids or a predicate over committed metadata) and (ii) a policy threshold θ_{fair}. The Oracle returns the aggregate metric value and a ZKP π asserting the metric was computed over exactly that committed cohort and satisfies the declared policy; the verifier then checks $\text{Verify}(\pi, \theta_{\text{fair}}, R_t) = 1$ without receiving per-decision sensitive features [2].

AIFDR's DAG is primarily a provenance/lineage structure; when a probabilistic causal semantics layer is available (e.g., for counterfactual reasoning), it can be expressed via a standard factorization:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(xi \mid \text{pa}_G(x_i)), \quad (8)$$

where $\text{pa}_G(xi)$ denotes the parents of xi in graph G. This factorization is optional replay scaffolding, not a claim of a closed causal model.

6 Applications Layered on the Spine

6.1 Counterfactual Replay Engine (CRE)

CRE consumes commitments and, when available, assembled DAGs to support forensic replay, bounded counterfactual queries under Contract C4, and divergence reports. Replay is either rejected or explicitly flagged as non-dispositive unless declared dependency coverage and validity-envelope conditions hold. In high-velocity dispute settings, CRE can help institutions distinguish exact reconstruction from bounded counterfactual analysis. Its value is not that it guarantees a unique explanation for every denial or block, but that it makes evidentiary limits explicit when replay cannot be trusted as ground truth.

6.2 Digital Twin Governance Testing

A digital twin is treated as a downstream consumer of governance outputs rather than part of the critical path: it mirrors committed decision streams, injects scenarios, compares proof-backed compliance metrics, and supports regulator-injected tests.

7 Conclusion

AIFDR reframes algorithmic governance as a systems contract: bounded capture, commitment-ledger integrity, and proof-backed governance APIs. Treating replay and digital twins as layered applications avoids critical-path latency penalties while enabling forensic and compliance capabilities. The contribution of this short paper is therefore not a claim of complete causal observability or measured deployment overhead, but a reviewable contract for making governance guarantees, omissions, and replay admissibility explicit in financial AI. Next steps include prototype evaluation of sidecar overhead, coverage attestation, and proof cost, including hardware acceleration. Open challenges include completeness vs. privacy, prototype overhead evaluation, proof-cost reduction, and accountability semantics across delegation graphs.

References

[1] C. Yue et al. GlassDB: An Efficient Verifiable Ledger Database System Through Transparency. Proc. VLDB Endow., 16(6):1359–1372, 2023. https://doi.org/10 .14778/3583172.3583175

[2] S. Waiwitlikhit et al. Trustless audits without revealing data or models. arXiv preprint arXiv:2404.04500, 2024. https://arxiv.org/abs/2404.04500

[3] X. Yang et al. LedgerDB: A Centralized Ledger Database for Universal Audit and Verification. Proc. VLDB Endow., 13(12):3138–3151, 2020. https://doi.org/10.1 4778/3415478.3415540

[4] J. Cheney, L. Chiticariu, and W.-C. Tan. Provenance in Databases: Why, How, and Where. In Foundations of Software Technology and Theoretical Computer Science (FSTTCS), pages 379–394. Springer, 2009. https://doi.org/10.1007/978-3- 642-10631-6_31

[5] J. Barr. Now Available – Amazon Quantum Ledger Database (QLDB). AWS News Blog, 2019. https://aws.amazon.com/blogs/aws/now-available-amazonquantum- ledger-database-qldb/

[6] S. Verma, V. Boonsanong, M. Hoang, K. Hines, J. Dickerson, and C. Shah. Counterfactual Explanations and Algorithmic Recourses for Machine Learning: A Review. ACM Computing Surveys, 57(1), 2024. https://doi.org/10.1145/3677119

[7] A.-H. Karimi, B. Schölkopf, and I. Valera. Algorithmic Recourse: From Counterfactual Explanations to Interventions. In Proceedings of the 2021 ACM FAccT, pages 353–362, 2021. https://doi.org/10.1145/3442188.3445899

[8] B. Pérez, J. Rubio, and C. Sáenz-Adán. A Systematic Review of Provenance Systems. Knowledge and Information Systems, 57:495–543, 2018. https://doi.org/10.1 007/s10115-018-1164-3

About the Authors:

Shreshta Shyamsundar
Infosys Limited, Australia
shreshta_shyamsundar@infosys.com

Shreshta Shyamsundar is a globally trusted thought leader and specialist in the field of establishing enterprise-wide high-performing software engineering practices, extending to quality engineering, DevSecOps, site reliability engineering, cloud expansion, and optimization to deliver business capabilities at scale and improve the quality of work for software developers and the quality engineering community across industries.

Anmol Jain
Infosys Consulting Limited, Australia
Anmol_jain10@infosysconsulting.com

Anmol Jain heads Infosys Consulting for APAC, running a diverse group of management consultants who cover markets from Australia and New ZeaWland to Singapore, China, and Malaysia. He's a passionate builder of teams and works relentlessly to help clients to steer an innovation agenda to win market share and achieve competitive advantage. Anmol has experience in leading large-scale core and digital transformation for financial services. He's also a business mentor, serving as a pro bono advisor to a number of startups.

For more information, contact consulting@infosys.com

Infosys® | CONSULTING