Engineering Trust in AI Systems: A Data-Layer Framework for Explainability and Auditability

[1] Mohammed Arbaaz Shareef
[1] Lead Data Engineer at Anblicks

Received: 02 Jan 2026 | Received Revised Version: 14 Jan 2026 | Accepted: 13 Feb 2026 | Published: 24 Feb 2026

Volume 08 Issue 02 2026 | Crossref DOI: 10.37547/tajiir/Volume08Issue02-11

Abstract

The article examines engineering approaches for strengthening trust in AI systems through data-layer controls that make decisions explainable and verifiable in audits. The widening regulatory and organizational demands for the traceability of training data, the reproducibility of pipelines, and the defensible documentation of model behavior in production drive practical relevance. Scientific novelty lies in integrating provenance capture, lineage graphs, feature-store governance, and standardized documentation artifacts into one coherent data-layer framework that produces machine-checkable evidence. The work describes lifecycle evidence generation from ingestion to inference, studies how provenance models and lineage datasets support inspection, and analyzes how documentation instruments complement technical traces. Special attention is given to preventing evidence gaps caused by opaque preprocessing, weak versioning, and incomplete logging. The study aims to systematize a data-layer architecture that supports explainability and auditability without relying on new model classes. A comparative analysis of recent research, a synthesis of published frameworks, and a structured review of sources are employed in this study. The conclusion summarizes actionable controls and their expected audit outputs. The article targets engineers, MLOps teams, risk functions, and internal/external auditors.

Keywords: *trustworthy AI, explainability, auditability, data provenance, data lineage, feature store, model documentation, traceability, metadata governance, reproducible pipelines.*

Cite This Article: Shareef, M. A. (2026). Engineering Trust in AI Systems: A Data-Layer Framework for Explainability and Auditability. The American Journal of Interdisciplinary Innovations and Research, 8(2), 83–89. https://doi.org/10.37547/tajiir/Volume08Issue02-11

1. Introduction

Trust in AI deployments increasingly depends on whether a decision can be reconstructed from evidence: which data were used, how they were transformed, which features were served, which model version produced the output, and which safeguards prevented tampering or silent drift. In many organizations, these requirements often conflict with fragmented data stacks, ad-hoc preprocessing, and inadequate documentation practices, resulting in audit trails that cannot support technical scrutiny or formal assurance.

This study aims to develop a data-layer framework that enables engineers to provide verifiable decision evidence while improving the operational quality of explainable artifacts.

The tasks of the study are:

1) to formalize the evidence objects that must be produced at the data layer for decision reconstruction across training and inference;

2) to describe architectural mechanisms that generate, store, and query such evidence, focusing on provenance, lineage, feature management, and versioning;

3) to systematize documentation instruments that connect technical traces with review procedures and stakeholder-readable reporting.

Scientific novelty is associated with a unified engineering view that couple's provenance/lineage instrumentation with standardized documentation (model passports, model cards, audit cards) and positions feature governance as an evidence-producing subsystem rather than only a productivity tool.

2. Materials and Methods

The study draws on recent peer-reviewed and archival publications that cover provenance technologies and compliance-driven provenance requirements (M. Ahmed [1]); a real-world dataset of lineage graphs for governance research (Y. Chen [2]); an enterprise feature-store architecture and pipeline separation across feature, training, and inference flows (J. de la Rúa Martínez [3]); layered model documentation for safety and procurement decisions (S. Gilbert [4]); a standardized "model passport" concept for traceability and lifecycle identity (V. Kalokyri [5]); feature-store pipeline mechanisms and point-in-time correctness considerations (R. Liu [6]); large-scale evidence on dataset licensing and attribution provenance risks (S. Longpre [7]); a multi-level audit blueprint for advanced AI systems (J. Mökander [8]); an end-to-end PROV-compliant provenance model for ML pipelines with automated extraction (M. Schlegel [9]); and structured reporting templates for audits that preserve interpretability of evaluation results (L. Staufer [10]).

For writing the article, comparative analysis of research positions, analytical synthesis of engineering mechanisms, and structured review of documented designs were applied to derive an integrated data-layer framework and to map evidence artifacts to audit and explanation needs.

3. Results

Trust engineering at the data layer can be specified as a discipline of producing decision evidence with predictable structure, queryability, and integrity guarantees. Under this view, explainability no longer depends solely on post-hoc interpretation techniques; it depends on whether the system preserves a reproducible trail that links a prediction to (i) raw inputs, (ii) transformation code and parameters, (iii) feature materializations, (iv) model identity and training lineage, and (v) operational logs that confirm when and how the decision occurred. The reviewed literature supports treating provenance and lineage as the core representational substrate of that trail [1; 9], while documentation instruments act as boundary objects that translate technical evidence into stakeholder-readable disclosures [4; 5; 10].

A data-layer trust framework can be grounded in four evidence strata. The first stratum is source evidence, which captures origins, licensing constraints, collection pathways, and ownership expectations for training and fine-tuning datasets. Large-scale audits of dataset licensing demonstrate that missing or inconsistent license metadata result in a compliance exposure that cannot be easily corrected after models are trained, as training is resource-intensive and decisions about reuse become irreversible in practice [7]. From an engineering perspective, this motivates a mandatory capture of dataset identity, license pointers, and provenance notes as first-class metadata objects attached to ingestion events and propagated forward to training artifacts. The second stratum is transformation evidence, which captures how raw data become training- and inference-ready tables, including schema evolution, filtering, joins, imputation, aggregation windows, and feature extraction logic. Provenance technologies surveyed in applied domains emphasize logging, cryptographic integrity measures, blockchain-style immutability, and ontology/metadata approaches as families of mechanisms that can be combined to preserve the history of processing operations [1]. In ML pipelines, the operational gap repeatedly appears at this stage: teams can often reproduce code, yet cannot reliably reproduce the precise data state and transformation sequence that led to a specific model snapshot. A PROV-compliant provenance model that ties pipeline artifacts to execution events and version control signals addresses that gap by representing the pipeline as a graph of entities and activities that can be queried and inspected [9]. The third stratum is feature evidence, which captures point-in-time correctness, feature versioning, serving parity between training and production, and feature reuse boundaries. Feature stores are increasingly treated as a DBMS-like layer for ML features. Still, their trustworthiness lies in their ability to enforce consistent feature definitions, mitigate leakage through time-aware joins, and provide a governance surface for feature discovery and reuse [6]. The fourth stratum is model and decision evidence, which binds a prediction to a model identity, deployment configuration, and runtime signals (inputs, outputs, timing, and operational environment). A "model passport" formalizes this binding through structured metadata that uniquely identifies, traces, and monitors models from data acquisition through deployment, shifting documentation away from ad-hoc narratives to standardized, automation-friendly records [5].

These strata become operational only if evidence is produced continuously and can be validated. Here, audit literature offers two complementary mechanisms. First, audits can be structured across multiple levels—organizational procedures, model-level properties, and application-level behavior—so that evidence produced in production informs redesign and risk management rather than being an afterthought [8]. Second, even rigorous evaluations lose interpretability when reporting omits situational information about who performed the assessment, what access existed, how procedures were executed, and which review mechanisms were applied. Audit cards operationalize this by prescribing a structured report template that captures evaluator identity, evaluation scope, methodology, resource access, process integrity, and review procedures, ensuring that audit results retain interpretability when transferred across stakeholders [10]. Within the proposed data-layer framework, audit cards serve as a reporting wrapper around evidence strata, ensuring that evidence consumption is disciplined and comparable.

A practical architecture emerges when the above mechanisms are composed into a single evidence graph that spans ingestion, transformation, feature computation, training, deployment, and inference. Lineage graphs provide the natural organizing structure for this composition. A real-world dataset of lineage graphs introduced for governance research demonstrates that lineage can be represented as graphs of assets and relationships, enabling the study of governance tooling and analytics over lineage structures rather than relying on toy examples [2]. In an engineering implementation, lineage graphs serve two functions: (i) operational dependency tracing (what downstream models depend on a given upstream table or transformation), and (ii) evidentiary reconstruction (what upstream data and code produced the particular feature vector used in a decision).

Figure 1 integrates the reviewed evidence mechanisms into a single data-layer trust design that supports both explainability and auditability by construction, rather than by retrospective investigation.

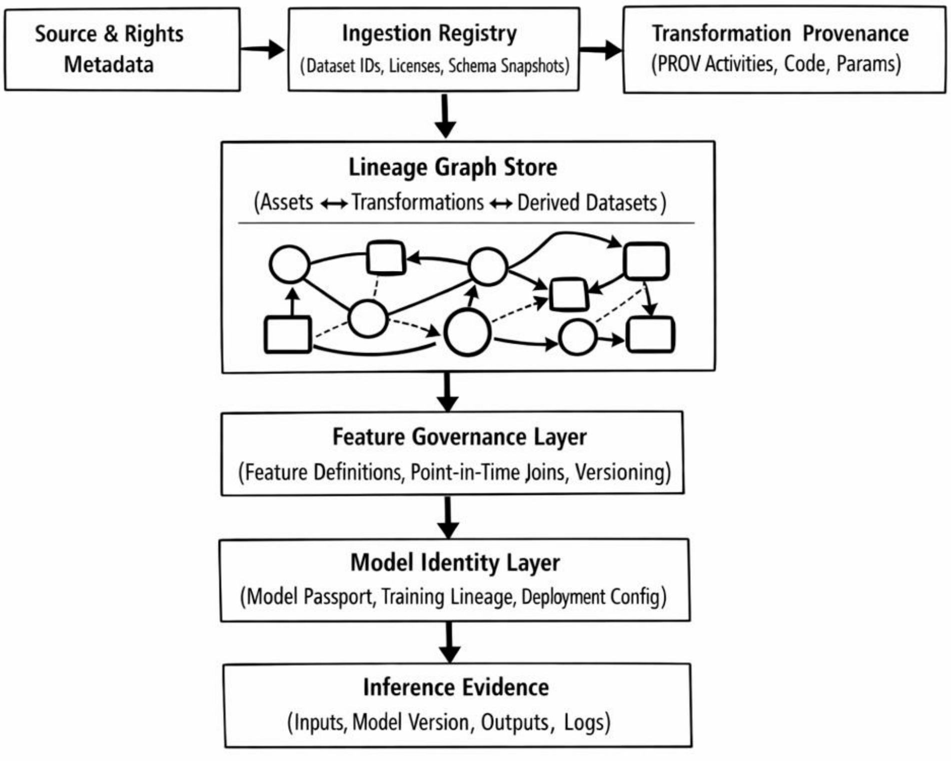
Source & Rights Metadata
Ingestion Registry
(Dataset IDs, Licenses, Schema Snapshots)
Transformation Provenance
(PROV Activities, Code, Params)
Lineage Graph Store
(Assets ↔ Transformations ↔ Derived Datasets)
Feature Governance Layer
(Feature Definitions, Point-in-Time Joins, Versioning)
Model Identity Layer
(Model Passport, Training Lineage, Deployment Config)
Inference Evidence
(Inputs, Model Version, Outputs, Logs)

Figure 1. Data-layer trust framework linking provenance, lineage, feature governance, and audit reporting [5; 9; 10]

The framework clarifies how explainability narratives can be supported with verifiable backing. Documentation artifacts such as model cards, when layered and linked to deeper evidence, support procurement and safety reasoning by directing stakeholders to the underlying records rather than substituting for them [4]. In this architecture, stakeholder-facing documentation serves as a pointer system, referencing traceable identifiers, reproducible pipeline runs, and queryable provenance graphs. That coupling reduces the standard failure mode where documentation exists but cannot be corroborated.

4. Discussion

The synthesized framework suggests that data-layer trust engineering is best evaluated by audit questions rather than by platform slogans. Three audit questions dominate: (i) decision reconstruction—can a decision be traced to specific data and transformations; (ii) evidence integrity—can tampering or silent substitution be detected; (iii) comparability—can different evaluations be interpreted consistently across reviewers. Provenance models and extraction tools support reconstruction by representing pipelines as queryable graphs [9]. In contrast, provenance technologies surveyed in applied sectors motivate the combination of logging with integrity controls to make historical records dependable [1]. Audit cards enhance comparability by standardizing the information required to be stated about the evaluation process, thereby preventing interpretive failures when results are disseminated without adequate procedural disclosure [10]. Multi-level auditing blueprints emphasize that evidence must circulate across organizational and technical layers, or audits will degrade into isolated checks that do not impact engineering practice [8]. Model passports and layered model documentation strengthen the binding between models and their lifecycle evidence, allowing reviewers to locate precise identifiers rather than relying on informal naming conventions [5; 4]. Dataset provenance audits show that rights metadata omissions generate a persistent governance debt that propagates downstream and complicates assurance claims [7].

Table 1 systematizes the data-layer artifacts that serve which assurance function and the evidence output they produce.

Table 1. Data-layer artifacts and the audit evidence they generate [1–10]

Data-layer artifact	Assurance purpose	Evidence output for reviewers	Typical source support
Dataset provenance record	Provenance of origins and usage constraints	Dataset identifiers, license pointers, sourcing notes	Dataset provenance auditing
Transformation provenance graph (PROV-style)	Reproducibility of preprocessing and training flows	Graph of entities/activities, links to code, and runs	PROV-compliant ML pipeline provenance
Lineage graph store	Dependency tracing across assets	Upstream/downstream impact paths, asset relation graph	Lineage dataset and governance graph framing
Feature store governance	Point-in-time correctness and serving parity	Feature definitions, version history, time-aware joins	Feature store pipeline architecture and correctness concerns; feature-store system design

Model identity record (model passport)	Verifiable model identification across the lifecycle	Structured model metadata, versioning, trace links	Model passport/traceability framework
Layered model documentation	Stakeholder-readable disclosure linked to evidence	Standardized summaries with pointers to deeper records	Layered model card reasoning in deployment settings
Audit reporting template (audit cards)	Interpretability of evaluations across reviewers	Structured disclosure of evaluation procedures and access	Audit card framework and features to report
Multi-level audit mapping	Alignment of evidence production with governance needs	Audit scope separation across org/model/app levels	Three-layer audit blueprint for advanced AI systems

None of the artifacts alone provides assurance. Assurance emerges from cross-linking identifiers, so that each layer can be corroborated by at least one independent trace. Lineage graphs corroborate transformation graphs, feature store versions corroborate inference logs, model passports corroborate deployment identities, and audit cards corroborate how evaluations were performed. This chaining logic matches the shift in audit thinking toward structured, multi-level procedures that rely on evidence flows rather than single-point attestations [8].

The second table focuses on practical failure modes that repeatedly break explainability and auditability, together with data-layer controls suggested by the reviewed sources.

Table 2. Frequent trust failures and data-layer mitigations with traceable evidence outputs [1–10]

Failure mode	**Data-layer mitigation**	**Evidence produced**
Untraceable preprocessing decisions	PROV-style capture of transformations and run metadata	Queryable provenance graph tied to executions
Training–serving mismatch and leakage	Feature store enforcing time-aware joins and versioned definitions	Feature definitions, point-in-time join logs, version history
Model identity ambiguity across environments	Model passport metadata bound to deployment configs	Model identifiers, lifecycle trace links
Rights and attribution uncertainty for training data	Mandatory dataset provenance capture and license linking at ingestion	Dataset provenance records and license pointers

Lineage blind spots across heterogeneous assets	Central lineage graph repository with asset relations	Upstream/downstream paths and impact traces
Evaluation results misinterpreted by reviewers	Audit card reporting of evaluator identity, access, and procedures	Structured audit report template fields
Audit procedures disconnected from engineering action	Multi-level audit design with feedback loops for development	Cross-level audit outputs feeding redesign
Documentation that cannot be corroborated	Layered documentation tied to traceable identifiers	Documentation pointers to concrete evidence objects

In organizations without experimental validation, analytical assurance still benefits from "evidence sufficiency" criteria, meaning that each mitigation should emit artifacts that enable an independent reviewer to reproduce at least the data state and transformation history associated with a decision. The provenance extraction approach, grounded in MLflow/Git activity traces, illustrates one path to automating sufficiency by continuously extracting provenance graphs rather than relying on manual reconstruction [9]. At the same time, dataset provenance audits reveal that governance failures can be upstream and non-technical (such as licensing ambiguity), so evidence production must begin at ingestion rather than at model training checkpoints [7]. Audit cards and layered documentation show that reviewers require procedure disclosure to interpret evidence correctly, so evidence production without disciplined reporting still leaves room for misinterpretation and weak assurance claims.

5. Conclusion

The study formulated a data-layer trust framework that treats explainability and auditability as properties of evidence production rather than of model class selection. The first task was addressed by specifying evidence objects spanning dataset provenance, transformation provenance graphs, lineage graph stores, feature governance records, model identity metadata, and runtime decision logs, complemented by standardized documentation and audit reporting. The second task was addressed by describing an architectural composition in which lineage and provenance graphs act as the integrating substrate. At the same time, feature stores and model passports bind training and inference to verifiable identifiers, ensuring consistency and transparency. The third task was addressed by systematizing documentation instruments—layered model documentation and audit cards—as mechanisms that preserve interpretability of technical evidence under review conditions, enabling consistent evaluation across stakeholders and audit cycles.

References

1. Ahmed, M., Dar, A. R., Helfert, M., Khan, A., & Kim, J. (2023). Data provenance in healthcare: Approaches, challenges, and future directions. Sensors, 23(14), 6495. https://doi.org/10.3390/s23146495

2. Chen, Y., Zhao, Y., Li, X., Zhang, J., Long, J., & Zhou, F. (2024). An open dataset of data lineage graphs for data governance research. Visual Informatics, 8(1), 1–5. https://doi.org/10.1016/j.visinf.2024.01.001

3. de la Rúa Martínez, J., Buso, F., Kouzoupis, A., Ormenisan, A. A., Niazi, S., Bzhalava, D., Mak, K., Jouffrey, V., Ronström, M., Cunningham, R., Zangis, R., Mukhedkar, D., Khazanchi, A., Vlassov, V., & Dowling, J. (2024). The Hopsworks feature store for machine learning. In Companion of the 2024 International Conference on Management of Data (SIGMOD '24) (pp. 135–147). Association for Computing Machinery. https://doi.org/10.1145/3626246.3653389

4. Gilbert, S., Adler, R., Holoyad, T., & Weicken, E. (2025). Could transparent model cards with layered accessible information drive trust and

safety in health AI? npj Digital Medicine, 8(1), 124. https://doi.org/10.1038/s41746-025-01482-9

5. Kalokyri, V., Tachos, N. S., Kalantzopoulos, C. N., Sfakianakis, S., Kondylakis, H., Zaridis, D. I., Colantonio, S., Regge, D., Papanikolaou, N., Marias, K., Fotiadis, D. I., Tsiknakis, M., & (2025). AI model passport: Data and system traceability framework for transparent AI in health. Computational and Structural Biotechnology Journal, 28, 386–404. https://doi.org/10.1016/j.csbj.2025.09.041

6. Liu, R., Park, K., Psallidas, F., Zhu, X., Mo, J., Sen, R., Interlandi, M., Karanasos, K., Tian, Y., & Camacho-Rodríguez, J. (2023). Optimizing data pipelines for machine learning in feature stores. Proceedings of the VLDB Endowment, 16(13), 4230–4239. https://doi.org/10.14778/3625054.3625060

7. Longpre, S., Mahari, R., Chen, A., et al. (2024). A large-scale audit of dataset licensing and attribution in AI. Nature Machine Intelligence, 6, 975–987. https://doi.org/10.1038/s42256-024-00878-8

8. Mökander, J., Schuett, J., Kirk, H. R., et al. (2024). Auditing large language models: A three-layered approach. AI Ethics, 4, 1085–1115. https://doi.org/10.1007/s43681-023-00289-2

9. Schlegel, M., & Sattler, K.-U. (2025). Capturing end-to-end provenance for machine learning pipelines. Information Systems, 132, 102495. https://doi.org/10.1016/j.is.2024.102495

10. Staufer, L., Yang, M., Reuel, A., & Casper, S. (2025). Audit cards: Contextualizing AI evaluations (arXiv:2504.13839). arXiv. https://arxiv.org/abs/2504.13839