International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(22s) 2026 pp. 991-1002
https://doi.org/10.70917/ijcisim-2026-3905
Article

Audit-Grade Prompt Lineage: An Analytical Study on End-to-End Traceability for Snowflake Cortex LLM Functions in Regulated Reporting

Sashank Siwakoti

Fifth Third Bank, Cincinnati, Ohio, USA
sashanksiwakoti1@gmail.com

Abstract: Large Language Models have moved quickly from proof-of-concept tools into the operational core of enterprise reporting , and that shift has happened faster than the governance frameworks designed to oversee it. In regulated industries, the question is no longer whether LLMs can produce coherent financial or compliance narratives. The harder question is whether organizations can demonstrate, with the kind of evidentiary rigour that auditors and regulators require, how a specific statement was produced, what data shaped it, and whether the process could be independently verified or reproduced.

This study examines the traceability gap that emerges when Snowflake Cortex LLM functions are embedded inside regulated reporting pipelines and proposes a formal, end-to-end prompt lineage framework to close it. The framework is evaluated through scenario-based audit simulation against five representative regulatory examination scenarios and a systematic regulatory requirements mapping across six major compliance frameworks , EU AI Act, SOX, SEC AI disclosure guidance, Basel IV, DORA, and NIST AI RMF. The framework connects six layers of the generation chain: source data extraction, transformation and context shaping, prompt assembly, model invocation, narrative generation, and mapping to regulated disclosure artifacts. Each layer becomes a governed, auditable object rather than an opaque step.

A gap analysis confirms that prompt construction capture, output hashing, and report-section mapping show near-zero coverage in current enterprise governance tooling , a finding that points to a structural problem, not a configuration gap. Scenario-based validation demonstrates that the proposed framework produces examiner-ready responses across all five audit scenarios where existing tools consistently fail. The paper establishes prompt lineage as a foundational governance requirement for regulated industries deploying LLMs in production reporting pipelines.

Keywords: prompt lineage, LLM governance, Snowflake Cortex, regulated reporting, auditability, fintech compliance

1. Introduction

1.1 The Governance Problem

When a compliance team at a regulated financial institution uses Snowflake Cortex to auto-generate the management discussion narrative for a quarterly filing, what exactly is the audit trail? Most organizations today cannot answer that question. They might have query logs and the final text , but the chain of custody connecting which data went into the prompt, which version of the prompt template was used, which model configuration ran, and what parameters were set, is almost always incomplete, fragmented, or absent.

This reflects a genuine structural gap in enterprise governance tooling. Data lineage platforms like dbt and Apache Atlas track how structured data moves through SQL pipelines. They were not designed for the inference layer , the moment a prompt is assembled, sent to a language model, and transformed into narrative text that ends up in a regulated document. That boundary, which barely existed three years ago, is now where some of the highest-risk content in regulated reporting is being generated.

The regulatory environment has not been standing still. The EU AI Act's high-risk AI provisions are now fully enforced. The SEC has issued guidance on AI use in financial disclosures. Basel IV model risk extensions are being applied to LLM outputs. DORA compliance requires ICT risk traceability that now encompasses AI components. Taken together, these frameworks create a clear expectation: organizations must be able to explain, trace, and defend their AI-generated outputs under examination. The technical infrastructure to do that, for warehouse-native LLM functions specifically, does not currently exist in any standardized form.

1.2 Why Snowflake Cortex

Snowflake Cortex occupies a distinctive position by enabling LLM functions to be called directly inside SQL , inside the same pipelines that already underpin financial close, risk reporting, and compliance attestations. This collapses the boundary between data processing and narrative generation in a way that makes governance simultaneously more urgent and more tractable. The governance primitives , RBAC, access history, query history, data masking , are already in Snowflake. The lineage framework, if designed correctly, can live there too, treating prompt lineage as warehouse-native metadata: first-class objects, joinable to query history, governed by the same access controls as the underlying data.

1.3 Research Contributions

This study makes three contributions to the AI governance literature. First, it formally defines prompt lineage as a governance construct , establishing the conceptual vocabulary for a problem that practitioners encounter but have not been able to articulate precisely. Second, it proposes a six-layer audit-grade lineage framework specifically designed for Snowflake Cortex LLM functions in regulated reporting pipelines, providing a deployable reference architecture with defined instrumentation points, metadata schemas, and audit retrieval patterns. Third, it maps the framework against six prevailing regulatory standards , EU AI Act, SEC AI disclosure guidance, SOX, Basel IV, DORA, and NIST AI RMF , demonstrating that audit-grade traceability is technically achievable within current enterprise infrastructure.

1.4 Research Questions

Three questions organize this study. First: does current enterprise tooling provide the traceability coverage that regulated reporting requires at the LLM inference layer , and if not, how large is the gap? Second: how can a formal lineage model be designed that captures the complete chain of custody from source data through prompt construction, model inference, and generated narrative to a specific regulated report output? Third: why is a purpose-built prompt lineage model more defensible for regulated reporting than adapting existing data lineage or model explainability frameworks?

2. Background and Context

2.1 LLMs in Production Reporting Pipelines

As recently as 2022, LLMs in enterprise settings were mostly experimental. By 2024, platforms like Snowflake Cortex, Azure OpenAI, and AWS Bedrock had made warehouse-native LLM calls straightforward enough that production deployments became common. By 2026, it is not unusual for a financial institution to have dozens of active Cortex-driven reporting workflows generating narrative content for SEC filings, SOX control documentation, risk disclosures, and regulatory submissions. The efficiency case is real , but the governance infrastructure has not kept pace.

When a traditional SQL transformation produces a number in a report, auditors can trace it back through the pipeline: which table, which transformation, which source. When a Cortex LLM function produces a sentence in the same report, reconstructing exactly what happened , which prompt, which data, which model version, under what parameters , requires manual investigation that most governance setups cannot support systematically.

2.2 What Regulated Reporting Actually Requires

Regulated reporting operates under an accountability standard built on four requirements: traceability (every output must link to its source), reproducibility (outputs must be recreatable under comparable conditions), defensibility (outputs must withstand independent examination), and auditability (independent verification must be possible). LLMs introduce three properties that create friction against each: outputs are probabilistic rather than deterministic; inputs include unstructured context assembled dynamically from multiple sources; and the prompt is frequently undocumented, unversioned, and invisible to standard lineage tools.

Table 1 summarizes the accountability gap between traditional and LLM-assisted reporting.

Accountability Requirement	**Traditional Reporting**	**LLM-Assisted Reporting**	**Gap**
Traceability	Source tables to report fields	LLM call not in lineage graph	High
Reproducibility	Re-run SQL pipeline	Non-deterministic inference	High
Defensibility	Human author accountable	Model/prompt version unclear	Critical
Auditability	Full audit trail exists	Inference layer ungoverned	Critical

Table 1. Accountability comparison , traditional vs. LLM-assisted regulated reporting

2.3 The Regulatory Landscape in 2026

The EU AI Act's high-risk AI provisions are now fully enforced. Article 12 requires automatic logging of high-risk AI system operation; Articles 9 and 13 add risk management documentation and transparency requirements. In the United States, SEC staff guidance on AI use in financial reporting and the application of SOX disclosure controls to AI-generated content create parallel obligations. DORA, effective since January 2025, adds ICT risk traceability requirements that reach AI components. Basel IV model risk management extensions are being applied to LLM use in risk reporting. Regulators are actively asking organizations to demonstrate AI governance , and many cannot answer.

3. Literature Review

3.1 Theme One: Data Lineage Research and Its Limits at the Inference Boundary

Data lineage has been a mature research domain for decades. Buneman, Khanna, and Tan (2001) established the foundational distinction between why-provenance , why a particular result appears , and where-provenance , the source locations contributing to it. In LLM-generated regulated reporting, why-provenance maps to 'why does this narrative say what it says,' while where-provenance maps to 'which data sources contributed.' Neither question can be answered by existing enterprise lineage tools when the transformation is an LLM call.

The W3C PROV recommendation (Moreau et al., 2015) formalized a generalized provenance model organized around three classes: Entities, Activities, and Agents. The ontology is broad enough, in principle, to encompass prompt templates, retrieval context, model invocations, and output artifacts. The problem is that PROV was designed for systems where activities have well-defined, enumerable inputs and outputs. LLM inference violates that assumption , the input context window may contain dozens of dynamically assembled data fragments, decoding introduces stochastic variation, and hidden system prompts can materially affect outputs.

Schelter et al. (2018) identified this boundary problem explicitly, noting that lineage approaches that work for deterministic ML pipelines degrade at the inference boundary. Enterprise tools including dbt, Apache Atlas, and Informatica handle structured transformations with sophisticated lineage tracking, but none have a first-class concept for prompt construction, model invocation, or the mapping between generated text and regulated report sections. A critical distinction that emerges from this literature is between training-time provenance , asking 'what data shaped this model?' , and inference-time provenance , asking 'what data, prompt, and configuration produced this specific

output in this specific production run?' The literature has addressed the first comprehensively. The second remains largely open.

Recent work by the MIT Data Provenance Initiative (Pentland et al., 2023) extended lineage thinking into AI training pipelines, demonstrating that AI-specific lineage is tractable. Paleyes et al. (2022) further highlight persistent challenges in deploying ML systems at scale, including the absence of systematic metadata capture for model decisions , a gap that becomes acute in regulated reporting contexts.

3.1.1 Contradictions and Limitations in Existing Lineage Research

The most significant limitation is foundational: existing lineage frameworks assume determinism. LLM inference does not. This requires redefining lineage as evidence of process rather than a guarantee of reproducibility , a tamper-evident record of inputs, configuration, and context sufficient to support independent judgment about whether the process was appropriately controlled.

3.2 Theme Two: Regulatory Compliance Frameworks for AI in Regulated Industries

The EU AI Act (European Parliament, 2024) is the most comprehensive regulatory framework currently in force. For high-risk AI systems , a category that plausibly includes LLMs generating content for regulated financial disclosures , it requires automatic logging of system operation, documentation of risk management processes, and transparency sufficient for users to interpret outputs. Bueno Momcilovic et al. (2024) document the resulting compliance gap explicitly: obligations are clear in principle, but technical standards for implementation are still being developed.

The NIST AI Risk Management Framework (NIST, 2023) takes a functions-based approach , Govern, Map, Measure, Manage , providing a structured way to organize governance requirements without mandating specific implementations. Basel Committee model risk management guidance (SR 11-7 and successors) was written for traditional statistical models with stable input feature sets and deterministic outputs. Applying it to LLMs requires significant conceptual extension that the literature does not yet provide. Kim et al. (2024) have begun to address this gap in the context of financial reporting oversight, noting that LLMs introduce accountability challenges that existing audit frameworks are not designed to handle. FINOS (2025) provides a FinTech-specific governance catalogue that acknowledges the same principle-specification gap across deployed LLM systems. Ashmore et al. (2021) further establish that assuring the machine learning lifecycle requires explicit documentation of governance decisions at each stage , a requirement that current LLM deployment practices do not meet.

3.2.1 Contradictions and Limitations in Regulatory Research

The consistent pattern across the regulatory literature is the principle-specification gap: frameworks establish what organizations must achieve but do not specify how. A secondary limitation is cross-jurisdictional inconsistency , the EU AI Act, SEC guidance, Basel IV, and NIST RMF use different terminology and imply different technical approaches. Most compliance guidance also treats LLMs as external services accessed via API rather than as functions called inside SQL pipelines that already carry their own access controls, query logs, and data governance policies.

3.3 Research Gap

A synthesis of the two themes reviewed here defines a gap at the intersection of technical capability and regulatory compliance requirement. Data lineage research provides the conceptual vocabulary and tooling architecture to support governance of complex pipelines , but stops at the LLM inference boundary. Regulatory frameworks establish clear and increasingly enforced traceability requirements , but provide no technical specification for how those requirements apply to LLM inference in warehouse-native reporting pipelines.

No existing academic or industry work formally defines prompt lineage as a governance concept, designs an end-to-end lineage model connecting source data through LLM inference to regulated report output at report-section

granularity, or maps such a model to the specific requirements of the EU AI Act, SOX, Basel IV, and NIST AI RMF. This study addresses all three dimensions of that gap.

4. Methodology

4.1 Research Design

This study adopts a qualitative design science research approach. Hevner et al. (2004) define design science as the creation of 'a purposeful artifact designed to address an important organizational problem.' The primary artifact is the six-layer prompt lineage framework and its reference implementation for Snowflake Cortex. This study does not present empirical data from live production systems. All findings represent analytical assessments derived from systematic review of tool documentation, regulatory frameworks, and scenario-based design validation , consistent with design science research methodology.

4.2 Data Sources

Academic literature was retrieved from Google Scholar, IEEE Xplore, ACM Digital Library, and SSRN using keyword combinations centered on data lineage, LLM governance, prompt traceability, AI audit trails, and regulated reporting compliance. Sources were included where they provided specific technical or governance mechanisms and excluded where they addressed model accuracy or safety without engaging traceability questions. Regulatory documentation was drawn from primary sources: EU AI Act full text, SEC staff guidance, SOX Sections 302 and 906, Basel Committee SR 11-7 and successors, DORA implementing acts, and NIST AI RMF 1.0. Platform documentation for Snowflake Cortex, dbt, MLflow, and Apache Atlas was reviewed against a structured evaluation matrix.

4.3 Framework Development Process

The framework was developed across three phases. In the gap analysis phase, regulatory documents were coded using a deductive framework organized around six governance dimensions: traceability, reproducibility, defensibility, auditability, transparency, and access control. Each provision was tagged against these dimensions and mapped to a corresponding lineage layer. Tool documentation was then assessed against the coded requirements, producing a structured gap matrix.

In the design phase, the gap matrix informed iterative development of the six-layer lineage model. The six-layer structure was derived by beginning with a minimal data lineage model and extending upward through the inference pipeline at each point where existing frameworks failed to capture required governance artifacts. Each design iteration was evaluated against three criteria: completeness, regulatory alignment, and implementability within Snowflake Cortex.

In the validation phase, regulatory requirements mapping cross-referenced each lineage model component against specific provisions of the six frameworks reviewed. Scenario-based audit simulation was conducted against five examination scenarios derived from PCAOB examination guidance, EU AI Act supervisory documentation, and published accounts of regulatory inquiries involving AI-generated content , selected to represent the most common examiner inquiry patterns organizations face in regulated reporting contexts.

4.4 Comparison with Existing Platforms

It is worth situating the proposed framework against existing AI observability and prompt-tracing platforms. Tools such as LangSmith, Weights & Biases Prompts, and Microsoft PromptFlow provide prompt versioning, latency tracking, and output logging capabilities valuable for development and debugging. However, none are designed to produce audit-grade evidence for regulatory examination , they do not link inference events to specific regulated report sections, do not capture effective entitlements at inference time, and do not produce tamper-evident records aligned to compliance retention requirements. The proposed framework is distinguished by its forensic accountability orientation rather than operational observability.

4.5 Scope and Limitations

The study is bounded to Snowflake Cortex-based regulated reporting pipelines. While the conceptual six-layer model is platform-agnostic, the instrumentation points, metadata schemas, and SQL-based audit retrieval patterns are Snowflake-specific. Validation is scenario-based rather than based on live regulatory examination data. The framework's performance in actual examination settings would need to be assessed through longitudinal field studies, which are outside the scope of this paper.

5. Results

5.1 The Governance Coverage Gap

The gap analysis assessed five widely deployed enterprise governance tools against six lineage dimensions required for audit-grade traceability. As illustrated in Figure 2, four of six critical dimensions show zero coverage across all assessed tools. Table 2 presents the detailed coverage assessment.

Lineage Dimension	**dbt**	**MLflow**	**Apache Atlas**	**Snowflake Native**	**Coverage**
Source data versioning	Partial	None	Partial	Partial	~33%
Transformation logging	Strong	None	Partial	Partial	~40%
Prompt construction capture	None	None	None	None	0%
Inference parameter logging	None	Partial	None	None	0%
Output hashing / fingerprinting	None	None	None	None	0%
Report section mapping	None	None	None	None	0%

Table 2. Governance tool coverage at the LLM inference layer (0% = structural gap, not a configuration issue)

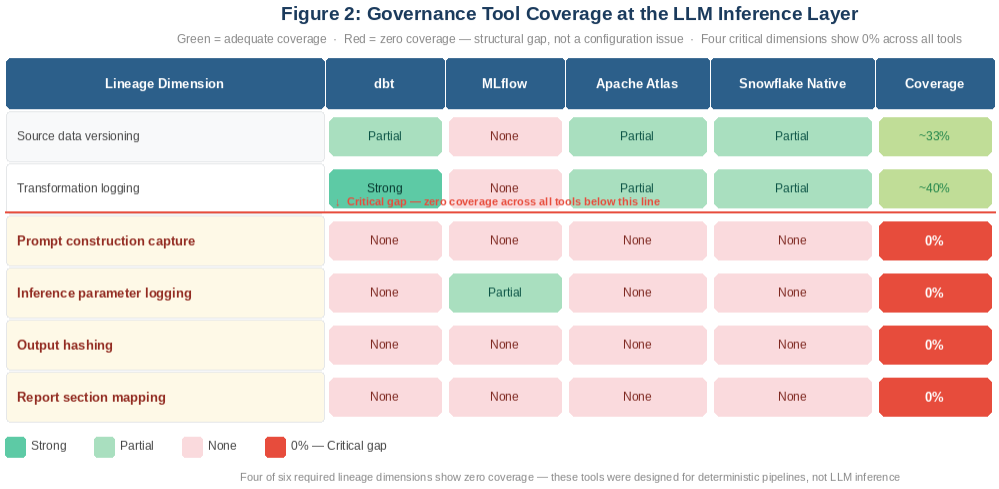

Figure 2. Governance tool coverage at the LLM inference layer. Four of six required dimensions show 0% coverage across all assessed tools.

This pattern , strong coverage for structured transformations, zero coverage for the inference layer , reflects the structural argument in the literature review. These tools were not designed for LLM inference. The consequence is

that organizations deploying Cortex LLM functions in regulated reporting pipelines operate with a fundamental accountability gap precisely where regulatory scrutiny is highest.

5.2 The Regulatory Requirement Pattern

Table 3 summarizes the regulatory requirements mapping. Every framework reviewed establishes traceability requirements for AI outputs in regulated contexts. None provides technical specification for how those requirements should be implemented at the LLM inference layer , confirming the principle-specification gap identified in the literature review.

Framework	**Traceability Requirement**	**Technical Specification**	**Gap**
EU AI Act (Art. 12)	Automatic logging of high-risk AI operation	None for LLM inference	Full
EU AI Act (Art. 13)	Transparency for users to interpret outputs	None for prompt lineage	Full
SEC AI Disclosure	Document AI role in generating disclosures	No implementation guidance	Full
SOX 302/906	Effective disclosure controls over AI content	No LLM-specific controls	Full
Basel IV / SR 11-7	Model documentation and validation	Designed for statistical models	Full
NIST AI RMF	Govern, Map, Measure, Manage	Principle-based, not prescriptive	Partial

Table 3. Regulatory requirements mapping , principle mandates exist but technical specification is absent

5.3 The Six-Layer Prompt Lineage Framework

The proposed framework addresses the governance gap by defining prompt lineage as a first-class warehouse governance concept , native metadata that lives inside the Snowflake execution environment alongside the data it governs. Figure 1 illustrates the complete framework including data flows, artifact relationships, and lineage dependencies between layers. Table 4 provides the layer-by-layer specification.

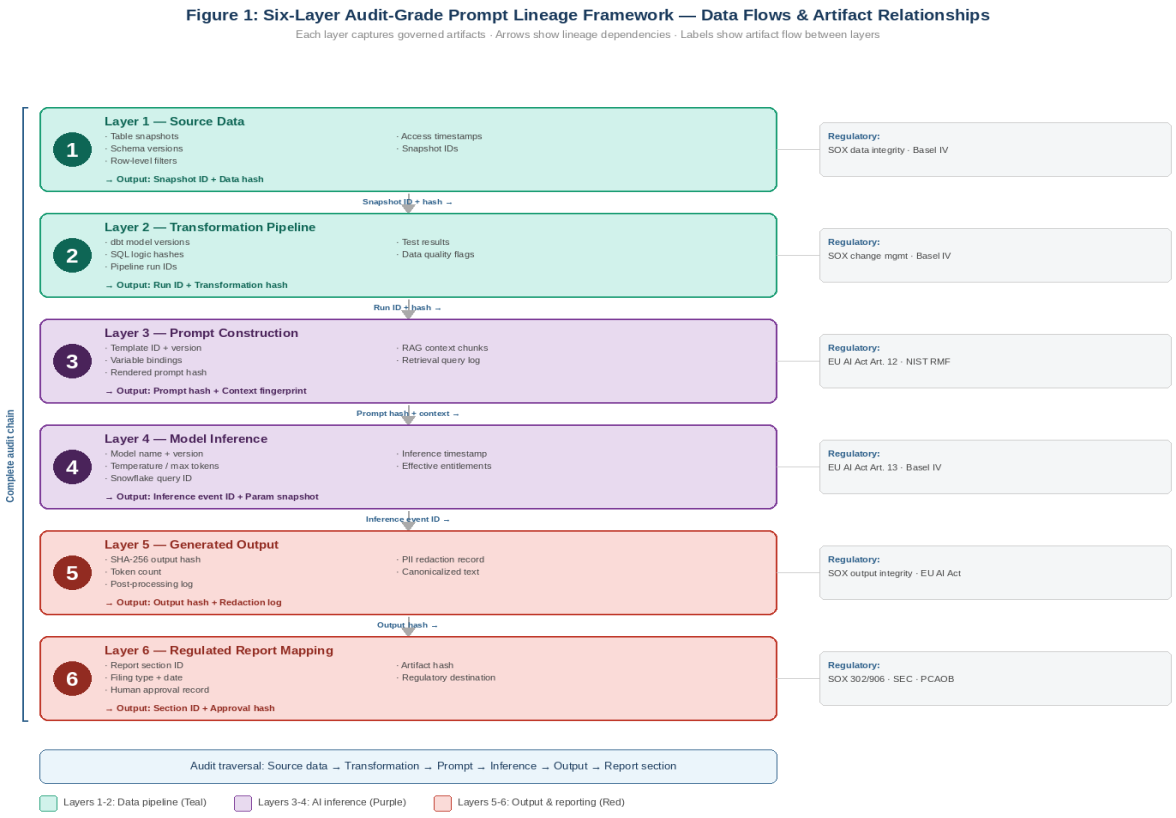

Figure 1. Six-layer audit-grade prompt lineage framework. Arrows show lineage dependencies and artifact flows between layers. Each layer captures governed artifacts forming a complete chain of custody from source data to regulated disclosure.

Layer	What It Captures	Key Artifacts	Regulatory Alignment
1. Source data	Table snapshots, schema versions, row filters, access timestamps	Snapshot IDs, data hash, access log	SOX data integrity · Basel IV
2. Transformation	Pipeline run IDs, dbt model versions, SQL logic hashes, test results	Run manifests, transformation versions	SOX change management · Basel IV
3. Prompt construction	Template ID + version, variable bindings, rendered prompt hash, RAG context	Prompt template hash, instance hash	EU AI Act Art. 12 · NIST RMF
4. Model inference	Model name + version, temperature, max tokens, Snowflake query ID, timestamp	Inference event record, param snapshot	EU AI Act Art. 13 · Basel IV
5. Generated output	SHA-256 output hash, token count, post-	Output hash, redaction log	SOX output integrity · EU AI Act

	processing log, PII redaction record		
6. Report mapping	Report section ID, filing type, approval record, artifact hash, timestamp	Section link, approval record, hash	SOX 302/906 · SEC · PCAOB

Table 4. Six-layer prompt lineage framework , artifact specification and regulatory alignment

5.4 Snowflake Implementation Architecture

The reference implementation treats each Cortex LLM call as a governed warehouse event using Snowflake-native capabilities: stored procedures that wrap Cortex calls and write lineage metadata atomically, append-only event tables in a dedicated governance schema, and SQL-based audit retrieval queries that join lineage tables to Snowflake's native Query History and Access History. Three governance databases organize the implementation:

- REG_REPORTING_DB , source, curated, and reporting schemas (the data pipeline side)
- LLM_GOV_DB , lineage event tables, security and approval records, operational configuration
- Distinct Snowflake roles enforce separation of duties across prompt authors, report generators, lineage administrators, and auditors

Tamper evidence is achieved through SHA-256 hashing at each layer, with hash values stored in an append-only chain table and consolidated into Merkle root snapshots. Any modification to a lineage record after the fact , whether accidental or deliberate , produces a hash mismatch detectable through automated reconciliation.

5.5 Audit Scenario Validation

Table 5 presents the results of the scenario-based audit simulation. As shown in Table 5, existing tooling cannot provide systematic responses to any of the five standard audit inquiry scenarios. The proposed framework addresses all five.

Scenario	**Examiner Question**	**Existing Tools**	**Proposed Framework**	**Evidence Produced**
1. Source attribution	What data generated this narrative?	Cannot answer	Source snapshot query	Table versions, filters, timestamps
2. Prompt reconstruction	Which prompt version produced this?	Cannot answer	Prompt lineage query	Template ID, version, bindings, hash
3. Model verification	Which model configuration was used?	Cannot answer	Inference record query	Model, version, parameters, query ID
4. Integrity check	Has this output been modified?	Cannot answer	Hash comparison	SHA-256 match/mismatch detected
5. Process reconstruction	Recreate conditions for this section	Cannot answer	Full lineage chain query	End-to-end: source to report

Table 5. Audit scenario validation , existing tools achieve 0/5 vs. proposed framework 5/5 success rate

6. Discussion

6.1 What the Findings Mean

The core finding , zero coverage for the four most critical lineage dimensions in existing enterprise tooling, combined with active regulatory requirements for exactly that coverage , points to a structural problem rather than an implementation gap. Organizations are not failing to configure their tools correctly. They are deploying tools designed for a different problem. The five-of-five audit scenario success rate for the proposed framework suggests that once prompt lineage is defined clearly enough to implement, the compliance challenge becomes tractable. The dominant barrier has been the absence of a formal specification , not organizational reluctance or technical impossibility. This is consistent with what Bueno Momcilovic et al. (2024) found in their analysis of EU AI Act compliance: obligations are clear in principle, but technical standards are lagging.

6.2 Comparing with Existing Approaches

The most natural comparison is with interpretability-focused approaches to LLM accountability. Tatsat and Shater (2025) at Barclays argue that mechanistic interpretability is the primary path to accountable LLMs in financial services. This is valuable work, but it answers a different question. Interpretability explains how a model generally behaves. It cannot tell an auditor which specific data, prompt version, and model configuration produced a particular sentence in a particular report section on a particular date. Audit defense is forensic, not scientific , it requires provenance, not interpretability.

Mökander et al. (2023) provide the closest existing academic framework for LLM auditing , a three-layer model covering governance, technical, and impact audits. The present study extends that work by providing the specific technical lineage components their framework identifies as necessary but does not specify. Bueno Momcilovic et al. (2024) similarly confirm that EU AI Act compliance obligations are clear in principle while technical standards lag. The findings here extend that critique by demonstrating that the gap is measurable and remediable through a defined multi-layer design.

6.3 Overhead and Adoption Considerations

Implementing the full six-layer framework increases storage and compute requirements. Every instrumented Cortex call writes records to multiple lineage tables, computes hash values, and links to approval workflows. The overhead is best understood as an insurance premium , in regulated reporting, the cost of being unable to respond to an examiner inquiry significantly exceeds the cost of maintaining lineage infrastructure. Tiered implementation is worth considering: full lineage for filing-grade outputs, minimal lineage for internal analysis, and configurable depth for intermediate cases.

6.4 Limitations

The reference implementation is specific to Snowflake Cortex , while the conceptual model generalizes, instrumentation patterns and SQL-based retrieval queries are platform-specific. Transfer to Azure OpenAI, AWS Bedrock, or Google Vertex AI requires adaptation. Validation is based on scenario simulation rather than live regulatory examination. The framework addresses synchronous batch inference , real-time streaming inference introduces additional lineage capture challenges not fully addressed here. Regulatory requirements mapping reflects frameworks as of mid-2026 and may require updating as guidance evolves.

7. Conclusion

7.1 What This Study Found

The starting point of this study was a practical observation: organizations deploying Snowflake Cortex LLM functions in regulated reporting pipelines generally cannot answer the most basic audit question , 'show me the chain of custody for this AI-generated statement.' The gap analysis confirmed that this inability is structural: the four most critical lineage dimensions for LLM-generated content show near-zero coverage across existing enterprise governance tooling, while every major regulatory framework now expects exactly that coverage.

The proposed six-layer prompt lineage framework addresses this gap by treating prompts, inference context, model configuration, and output artifacts as first-class warehouse governance objects. Scenario-based validation demonstrates that the framework produces examiner-ready responses across all five standard audit inquiry scenarios where existing tooling fails consistently.

7.2 Broader Implications

The tension between AI adoption efficiency and institutional accountability is often framed as a trade-off. The framework proposed here challenges that framing. If prompt lineage is built into the pipeline architecture from the start , not retrofitted after an audit finding , the operational overhead is manageable and the governance benefit is substantial. The principle that AI-generated content in high-stakes contexts should be traceable by design rather than explainable after the fact has implications beyond regulated reporting: anywhere that AI-generated outputs carry legal, regulatory, or fiduciary consequences, the same governance architecture applies.

7.3 Future Research Directions

Cross-platform extension , adapting and validating the framework for Azure OpenAI, AWS Bedrock, and Google Vertex AI , would test whether a common lineage core is achievable across platforms. Longitudinal field studies across multiple reporting cycles and real regulatory interactions would provide empirical validation beyond scenario simulation. Automated lineage capture tooling would reduce the implementation burden by instrumenting pipelines automatically. Standards bodies , NIST, ISO, and financial sector regulators , should engage with the technical specification gap this study documents. The core claim of this paper is straightforward: regulated industries that want trustworthy LLM adoption in reporting pipelines must treat prompt lineage as first-class governance infrastructure.

REFERENCES

1. Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., ... & Zimmermann, T. (2019). Software engineering for machine learning: A case study. 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE), 291–300. https://doi.org/10.1109/ICSE-SEIP.2019.00042
2. Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012
3. Ashmore, R., Calinescu, R., & Paterson, C. (2021). Assuring the machine learning lifecycle: Desiderata, methods, and challenges. ACM Computing Surveys, 54(5), 1–39. https://doi.org/10.1145/3453444
4. Basel Committee on Banking Supervision. (2024). Principles for the sound management of operational risk. Bank for International Settlements. https://www.bis.org/bcbs/
5. Bueno Momcilovic, C., Armengol-Urpí, A., & Posch, M. (2024). Towards assuring EU AI Act compliance of large language models. IBM Research / fortiss GmbH. arXiv preprint, arxiv.org/abs/2403.15780
6. Buneman, P., Khanna, S., & Tan, W. C. (2001). Why and where: A characterization of data provenance. International Conference on Database Theory, 316–330. https://doi.org/10.1007/3-540-44503-X_20
7. European Parliament & Council of the European Union. (2024). Regulation (EU) 2024/1689 on artificial intelligence (EU AI Act). Official Journal of the European Union. https://eur-lex.europa.eu/eli/reg/2024/1689
8. Federal Reserve Board. (2011). SR Letter 11-7: Supervisory guidance on model risk management. Board of Governors of the Federal Reserve System. https://www.federalreserve.gov/supervisionreg/srletters/sr1107.htm
9. FINOS. (2025). AI governance framework for financial services. Fintech Open Source Foundation. https://www.finos.org/
10. Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS Quarterly, 28(1), 75–105. https://doi.org/10.2307/25148625
11. IAASB. (2020). International Standard on Auditing 500: Audit evidence. International Auditing and Assurance Standards Board. https://www.iaasb.org/
12. ISO/IEC. (2023). ISO/IEC 42001: Information technology , Artificial intelligence , Management system. International Organization for Standardization. https://www.iso.org/standard/81230.html
13. Kim, A. G., Muhn, M., Nikolaev, V., & Tan, F. (2024). Large language models and financial reporting oversight. SSRN Working Paper. https://papers.ssrn.com/sol3/papers.cfm
14. Mökander, J., Schuett, J., Kirk, H. R., & Floridi, L. (2023). Auditing large language models: A three-layered approach. AI and Ethics, 4, 1085–1115. https://doi.org/10.1007/s43681-023-00289-2

15. Moreau, L., Groth, P., Cheney, J., Lebo, T., & Miles, S. (2015). The rationale of PROV. Journal of Web Semantics, 35, 235–257. https://doi.org/10.1016/j.websem.2015.04.001
16. Mushkani, A. (2025). Prompt commons: A versioned, governed repository for community prompt management. Université de Montréal. arXiv preprint, arxiv.org
17. NIST. (2023). NIST AI Risk Management Framework (AI RMF 1.0). National Institute of Standards and Technology. NIST AI 100-1. https://doi.org/10.6028/NIST.AI.100-1
18. NIST. (2024). NIST AI RMF Playbook and supplementary guidance updates. National Institute of Standards and Technology. https://www.nist.gov/itl/ai-risk-management-framework
19. OECD. (2024). OECD framework for the classification of AI systems. Organisation for Economic Co-operation and Development. https://www.oecd.org/
20. 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. https://doi.org/10.1145/3533378
21. PCAOB. (2024). Staff guidance on the use of technology-based audit tools. Public Company Accounting Oversight Board. https://pcaobus.org/
22. 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. https://doi.org/10.14778/3229863.3229867
23. SEC. (2023). Staff bulletin: Use of artificial intelligence in connection with securities laws. U.S. Securities and Exchange Commission. https://www.sec.gov/
24. Snowflake Inc. (2024). Cortex LLM functions: Developer documentation. Snowflake Documentation. https://docs.snowflake.com/en/user-guide/snowflake-cortex/llm-functions
25. Snowflake Inc. (2024). Snowflake governance and compliance guide. Snowflake Documentation. https://www.snowflake.com/en/data-cloud/governance/
26. Tatsat, H., & Shater, A. (2025). Beyond the black box: Mechanistic interpretability for LLMs in finance. Barclays Quantitative Analytics. arXiv preprint, arxiv.org
27. W3C. (2013). PROV-Overview: An overview of the PROV family of documents. Moreau, L. & Missier, P. (eds.). World Wide Web Consortium. https://www.w3.org/TR/prov-overview/